Modèles probabilistes et statistiques pour la conception et l'analyse des systèmes de communications

Paola Bermolen

To cite this version:

Paola Bermolen. Modèles probabilistes et statistiques pour la conception et l'analyse des systèmes de communications. domain_other. Télécom ParisTech, 2010. Français. NNT : . pastel-00005853

HAL Id: pastel-00005853
https://pastel.hal.science/pastel-00005853

Submitted on 8 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TELECOM
ParisTech

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de Docteur

de Télécom ParisTech

Spécialité : Informatique et Réseaux

Paola Bermolen

Modèles Probabilistes et Statistiques pour la Conception et l'Analyse des Systèmes de Communications

Soutenue le 26 février 2010 devant le jury composé de

Président	Serge FDIDA	Université Pierre et Marie Curie
Rapporteurs	Marco AJMONE MARSAN	Politecnico di Torino
	Gustavo de VECIANA	University of Texas
Examinateurs	Badih GHATTAS	Université de la Méditerranée
	Daniel KOFMAN	Télécom ParisTech
	Gonzalo PERERA	Universidad de la República
Directeurs de thèse	François BACCELLI	École Normale Supérieure et INRIA
	Dario ROSSI	Télécom ParisTech

Acknowledgements

It has been a long way to arrive to this moment and in it I found many wonderful people that made it possible. I am not going to present here an infinite list of all of them, but I cannot miss the opportunity to thank at least some of them.

I should thank my advisors Dario Rossi and François Baccelli for their always generous guide and for their trust on me. I have learned a lot from both of them and I am sure that their scientific as well as personal advices will be of precious value in my future academic activity. I am particularity fond of these two: always aim at excellence in what you do, and criticism is useful as long as it helps you to advance.

I also want to thank the reviewers of my thesis and the other members of the jury: Marco Ajmone Marsan, Gustavo de Veciana, Serge Fdida, Badih Ghattas, Daniel Kofman and Gonzalo Perera. A very special thank goes to Daniel for giving me the opportunity to start this PhD. I should also mention that he constantly pushes me to improve myself.

I also have to thank the great people I have met at the 5th floor of the Dareau site (specially for the "old" ones that started their PhD at the same time as me): Aruna, Claudio, Dorice, Gege, Masood, Paolo, Salma, Silvio, Stefano, etc. The good mood necessary to finish the thesis was in part due to them.

But all of this is only possible, without any doubts, thanks to the people that makes me believe that academic research is possible even if we are at a little place of the world and we are so few... of course I am referring to my friends and colleagues at the Universidad de la República in Uruguay. But since sensibility is best expressed in VO I will switch the language... Mis queridos artesanos, en especial Laurita, Pablo y Pedro: porque esto no hubiera sido posible sin ustedes, su confianza, aliento y presencia constantes (aunque sea a la distancia), lo único que puedo decir es: muchísimas gracias!

También tengo que agradecer a mi familia, que entendió que esto era lo mejor para mi y por eso se sacrificó. A mis amigos del alma, que por suerte son tantos que no los puedo nombrar. Pero muy especialemente al grupo de "amigas cibérneticas" que fueron un pilar fundamental que me permitió hacer de este doctorado una experiencia

completa y enriquecedora. Dentro de los uruguayos, no puedo dejar de agradecer al pequeño puñado que tengo más cerca, a la barra parisina: Rafa, Chloé, Edú, Chupete, Jairo y Rose. Ellos nos ayudaron a no sentirnos tan exiliados.

Pero si bien todos los que aquí menciono fueron parte de esta aventura, hay alguien que sin su presencia, su confianza total, su continuo apoyo y su amor, esto no hubiera sido ni remotamente posible. Por supuesto me refiero a mi Fede. Porque como dice Benedetti "... pero si falta usted no habrá milagro", estos milagros no hubieran sido ni siquiera imaginables sin vos y aunque ya lo sabes: simplemente gracias.

Abstract

Heterogeneity, at service and technology level, is one of the main characteristics of todays networks. The ever increasing offer of new services has changed not only the amount of traffic in the network but also its structure. As a result, traditional techniques for traffic prediction and classification are becoming obsolete. In the first part of the thesis, we address these two problems by means of advanced statistical tools. We analyze the problem of online prediction of the load on a link based only on past measurements and without assuming any particular model. Concerning traffic classification, and motivated by the widespread use of P2P systems, we focus on the identification of P2P applications, considering more precisely the case of P2P television (P2P-TV). For both cases, our framework makes use of Support Vector Machines (SVM). The algorithms we propose provide very accurate results, they are robust and their computational cost is extremely low. These properties make our solutions specially adapted to an online application.

As traffic is increasingly heterogeneous, so are the access technologies. In particular, wireless is destined to be the access technology of choice. In this context, self-organized systems such as Mobile Ah-doc Networks (MANETs), are of particular importance. The design of an efficient and fair multiple access (MAC) mechanism is then crucial. In the second part of the thesis, we address two different problems related to MAC mechanisms in MANETs (in particular, we concentrate on CSMA since it is the most widely deployed mechanism). Firstly, given the absence of a mathematical model of consensus, an analysis of the existing models, with special emphasis on their correlation with the real protocol, is presented. Some weakness are identified and possible solutions are proposed. The use of stochastic geometry tools allows us to obtain analytical results where other techniques cannot. Secondly, we address the problem of lack of QoS in CSMA and we propose two different mechanisms that guarantee a minimum rate for each accepted transmission. Both take the interference level into account to decide on the set of connections which can access the shared channel at any given time. The main difference between them is the possibility or not of adjusting the transmission power of the nodes. The main aim of our study is to identify which of the proposed mechanisms outperforms CSMA best depending on the scenario.

Résumé

Motivation

Internet est le premier vrai réseau multiservice; des services de voix, vidéo et données sont offerts sur le même réseau. La principale conséquence de cette multiplicité est le changement de la structure du trafic. Non seulement la quantité du trafic a augmenté, mais sa nature a aussi radicalement changé, principalement en raison de l'offre de nouveaux services avancés (à l'impact socio-économique incontestable). La popularité de ces services, tels que par exemple le partage de dossiers (e.g. KaZaa, eDonkey, BitTorrent), la téléphonie par Internet (e.g. Skype, Gtalk, VoIPbuster) ou la télévision par Internet a explosé ces dernières années. Par conséquence, le trafic de données sur le réseau est de plus en plus complexe et dynamique.

Ces deux caractéristiques se sont traduites par une imprévisibilité élevée du trafic, au moins avec des techniques traditionnelles et simples. Or la prédiction du volume de trafic est très utile pour la planification de capacité et pour les systèmes autogestionnaires (même s'ils impliquent des échelles de temps très différentes). Par exemple, une application typique de cette prédiction est la réservation dynamique de ressources dans le cadre du réseau privé virtuel (VPN en anglais) [1]. Une autre application, liée au nouveau domaine de *réseau vert* (ou green networking) [2], est le problème de l'adaptation du taux des liens (ALR en anglais). ALR est proposé comme une manière de réduire la consommation d'énergie en réglant le taux de lien configuré sur le routeur au minimum exigé. Il serait alors intéressant de concevoir un système de prédiction qui prévoie de façon exacte et en ligne, la variabilité de ce trafic et en même temps réduise au minimum les hypothèses sur sa structure (par exemple absence d'hypothèses sur les propriétés statistiques des mesures).

Cependant, la prédiction de quelques caractéristiques du trafic tels que la moyenne ou le maximum sur un intervalle donné de temps fournit peu d'information sur le trafic lui-même. Certaines applications conscientes du trafic (traffic-aware) exigent d'identifier réellement le type de trafic traversant le réseau, ou même l'application qui l'a généré. Par exemple, la classification du trafic (application) est utile à l'application des politiques de qualité de service (QoS), pour bloquer des applications interdites (e.g. P2P ou le chat au bureau), ou encore pour identifier des applications populaires dans des buts commerciaux (e.g. publicité). Un exemple de ces nouveaux types de trafic est P2P-TV (télévision Pair-à-Pair); services de *video-streaming* en temps réel à grande échelle qui exploitent le paradigme de communication pair-à-pair (P2P). Actuellement, plusieurs systèmes déployés de P2P-TV comportent le streaming de basse qualité [3, 4, 5, 6], mais l'utilisation de systèmes de haute qualité sera bientôt répandue [7, 8].

Les systèmes de P2P-TV peuvent être l'origine de problèmes graves pour l'Internet, comme le montre le succès croissant des systèmes commerciaux tels que PPLive, SopCast, TVAnts et beaucoup d'autres. En effet, le trafic de P2P-TV peut se développer sans contrôle, causant une dégradation de la qualité du service perçu par les internautes ou même le collapse du réseau [9]. Tandis que la liaison descendante est limitée par le taux de transfer, la liaison montante peut se développer de façon illimitée comme cela a été déjà observé en [10]. En outre, la conception propriétaire de quelques applications réussies de P2P-TV fait de l'identification de telles applications un thème d'importance croissante. Par exemple, un fournisseur d'accès à Internet (ISP en anglais) sera certainement intéressé à bloquer des applications gloutonnes de ressources, ou au moins à les identifier.

Bien que le problème de classification du trafic ne soit pas nouveau, pour différentes raisons, quelques techniques bien connues sont devenues désuètes. C'est par exemple le cas des techniques basées sur l'identification de ports ou l'identification du contenu des paquets. Tandis que dans le premier cas le mécanisme peut être trompé par le blocage des pare-feu ou bien par l'attribution dynamique des ports, la complexité informatique du deuxième peut être si grande qu'il ne pourrait pas être mis en oeuvre. Comme alternative, de très bons résultats peuvent être obtenus par la classification statistique[11, 12]. En plus, la classification comportementale, basée sur la logique de que les différentes applications produisent de différents modèles, a été employée dans le cadre de la classification à grain grossier des hôtes d'Internet [13, 14]. Ainsi, nous nous intéressons à la conception d'un mécanisme de classification visé aux applications de P2P-TV (c-à-d à grain fin) basées sur le paradigme comportemental.

Une autre caractéristique des réseaux d'aujourd'hui liées aux considérations précédentes est l'utilisation de nouvelles technologies. En effet, comme le trafic injecté dans le réseau est de plus en plus hétérogène, sont ainsi les technologies d'accès. En particulier, l'accès par des réseaux sans fil est de plus en plus utilisé. On s'attend à ce que dans un avenir proche, avec la prolifération des dispositifs intelligents aux nouvelles fonctionnalités, la communication entre les terminaux sans fil ait lieu habituellement sans passer par le coeur d'Internet. Ces terminaux agiront d'une façon autonome et adaptative, tombant dans la catégorie des réseaux ad hoc mobiles (MANETs en anglais).

Pour analyser la conception ou la performance de MANETs, un des aspects les plus difficiles est la nature partagée du canal. En soi, la conception d'un mécanisme du contrôle d'accès au support (MAC pour Medium Access Control) joue un rôle principal. Dans le meilleur des cas, un algorithme bien conçu de MAC permet un nombre maximum de transmissions simultanées tel qu'elles ne s'y interfèrent pas les unes avec les autres. Dans le même temps, l'inégalité dans les occasions d'accès doit être minimisée. Cependant, une telle conception devient vraiment difficile quand un mécanisme décentralisé est considéré (ce qui est le cas idéal).

Le mécanisme de MAC est alors appliqué localement pour chaque noeud, qui a seulement une information locale de l'état de réseau. Le mécanisme le plus largement déployé de MAC est CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), utilisé par exemple dans IEEE 802.11 et IEEE 802.15.4. Même s'il a été relativement bien étudié d'un point de vue empirique, il n'existe aucun modèle mathématique largement reconnu portant sur lui. Une analyse du protocole, et surtout l'identification des caractéristiques principales avec le propos spécifique de la modélisation, est étonnement manquant. Par exemple, les modèles plus répandus ne prennent pas en considération l'aspect aléatoire de l'état du canal dû aux effets de fading/shadowing. En plus, certains aspects de CSMA/CA sont simplement ignorés, comme le Clear Channel Assessment (CCA). Dans ce contexte nous analysons si ces modèles peuvent être généralisés au cas non déterministe, ou s'il est nécessaire et possible de définir un nouveau modèle pour CSMA/CA.

Dans tous les cas, CSMA/CA ne garantit rien en termes de la qualité obtenue pour les transmissions admises. Ce manque de QoS pourrait gêner l'exécution, par exemple, d'applications en temps réel avancées. Il serait alors intéressant de concevoir et d'analyser un mécanisme d'accès au support avec des garanties de performances. Comme mentionné ci-dessus, la conception d'un tel mécanisme doit faire face à la nature décentralisée et aux ressources partagées (d'où le QoS de chacun des connexions dépend des *toutes* les connexions actives) de MANETs.

En un mot, cette thèse présente quatre contributions principales :

- Nous analysons le problème de la prédiction de la charge sur un lien, et proposons un nouvel algorithme basé sur une technique originale d'Apprentissage Statistique, nommé Support Vector Machines (SVM) [15]. Notre proposition comparée à plusieurs techniques paramétriques et non paramétriques, obtient toujours les meilleurs résultats.

- Une variante de SVM est utilisée pour la conception d'un mécanisme de classification à grain fin d'applications de P2P-TV. La performance de notre proposition est validée avec des tests sur plusieurs traces du trafic, qui représentent un large éventail de scénarios. D'excellents résultats ont été obtenus.

- Nous analysons et discutons les modèles possibles pour les différentes types de CSMA/CA, en mettant un accent spécial sur la corrélation entre les hypothèses du modèle et les vraies caractéristiques de protocole. Une section spéciale est consacrée à la modélisation et à la comparaison des différents modes de CCA.

- Nous proposons deux mécanismes d'accès différents qui garantissent un QoS minimum (en termes de débit) pour toutes les transmissions admises. La performance de ces nouveaux mécanismes est comparée à celle du CSMA/CA classique dans plusieurs scénarios, considérant plusieurs métriques de performance. Le but principal de notre étude est d'identifier parmi les mécanismes proposés, celui qui obtient des meilleurs résultats selon le scénario.

Dans la section suivante, nous détaillons les contributions principales de la thèse.

Contributions Principales

Cette thèse est organisée en deux partie. La première est consacrée à la conception des outils basés sur l'apprentissage automatique, en particulier SVM [15, 16], pour faire face à deux des problèmes précédemment mentionnés : la prédiction de la charge sur un lien et la classification des applications P2P-TV. Les SVM sont un ensemble de méthodes pour la classification et la régression fondus dans le cadre de la théorie de l'apprentissage statistique. Si le premier travail dans ce domaine date des années soixante-dix [17], SVM a gagné l'attention de la communauté académique internationale seulement à partir des années quatre-vingt-dix. Les SVM ont été intensivement utilisés principalement dans le cadre de la reconnaissance des formes dans lequel ils ont montré une très bonne performance. Dans le cadre de la gestion des réseaux, ils ont été utilisés, par exemple, pour la détection d'anomalies ou l'estimation du throughput [18, 19]. Nous nous sommes intéressés à ces techniques puisqu'elles fonctionnent bien dans beaucoup de situations, grâce à leur généralisation à des données inconnues. D'ailleurs, elles sont adaptées à l'apprentissage continu et adaptatif en ligne, ce qui constitue une propriété extrêmement souhaitable pour nos buts.

Dans un premier temps, nous réalisons une étude profonde et complète des techniques de SVM. Nous analysons alors en détail leur performance lorsqu'elles sont fois appliquées aux problèmes mentionnés ci-dessus de gestion de réseau. Comme remarque générale, nous pouvons dire que dans les deux cas SVM s'est avéré être un outil très utile et puissant. Considérant que pour la prédiction de charge de lien, la robustesse de SVM et la basse complexité informatique apparaissent en tant que ses principaux atouts, pour la classification des applications de P2P-TV SVM fournit des résultats très précis.

Prédiction de la charge sur un lien

Pour la prédiction de la charge sur un lien, nous suivons la méthodologie connue sous le nom de "embedding procedure". Une série temporelle correspondant à des valeurs moyennes de la charge sur un lien dans une échelle de temps choisie est considérée, et une valeur future de cette série chronologique est prédit sur la base d'un nombre restreint d'observations du passé. En particulier, nous choisissons des échelles de temps petites, c.à.d moins d'une minute.

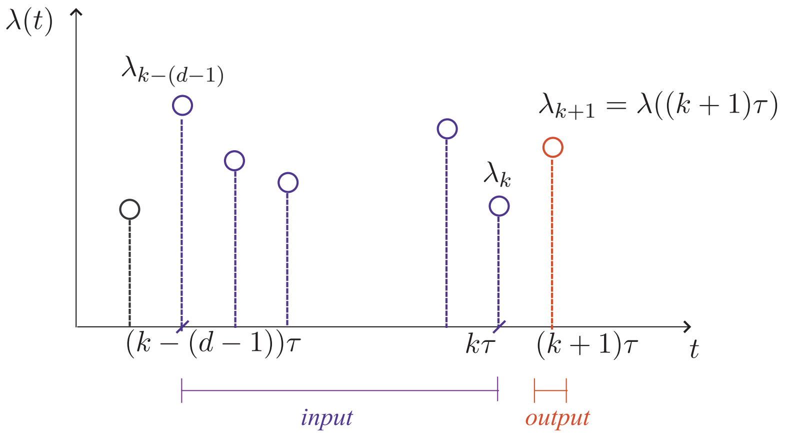


Figure 1: Série temporelle correspondant à la charge sur un lien et les inputs/outputs utilisés pour le embedding process.

Plus précisément, soit $\lambda(t)$ la charge du trafic mesurée sur l'intervalle de temps $[t-\tau, t]$. En quantifiant le temps par les multiples de τ, nous obtenons une série temporelle $\{\lambda_k\}_{k\in\mathbb{N}}$, où λ_k est la charge moyenne calculée sur l'intervalle $[(k-1)\tau, k\tau]$. Le SVR (Support Vector Regression) embedding process emploie alors un nombre arbitraire d d'observations du passé de la série défini ci-dessus afin de prévoir sa valeur future. Ainsi, une fois donné un input d-dimensionnel x ($S_X = \mathbb{R}^d$), une fonction SVR apprentie sur un ensemble d'entraînement retourne comme output $\hat{y} = f(x)$ une prévision de la cible y ($S_Y = \mathbb{R}$), qui dans notre cas est (voir Fig. 3.1) :

$$x = (\lambda_{k-(d-1)}, \ldots, \lambda_{k-1}, \lambda_k) \quad \text{and} \quad y = \lambda_{k+1} \tag{1}$$

Adoptant une approche empirique, nous évaluons l'efficacité de SVR en explorant un grand espace de paramètres et de conception. Notre but est double : d'abord, nous voulons évaluer l'exactitude de SVM et sa robustesse et, ensuite, nous voulons fournir des intuitions utiles sur la sélection des paramètres de SVM, un aspect pas toujours bien étudié dans les travaux précédents.

Nous comparons la performance de notre algorithme à ceux réalisables par des modèles paramétriques et non paramétriques tels que la moyenne glissante, les modèles autorégressifs ou l'estimateur de Nadaraya-Watson. Nos résultats montrent que, en dépit d'un bon accord avec les données réelles, le gain de SVR réalisable au-dessus des méthodes simples de prédiction n'est pas sufissant pour justifier son mise en oeuvre pour la prédiction de la charge sur un lien aux au moins pour des échelles temporelles courtes (voir Fig. 3.5 et Fig. 3.6).

Les Fig. 3.5 et Fig. 3.6 rapportent des résultats sur le RMSE (Root Mean Square Error) en fonction du nombre d d'observations du passé utilisées, pour les deux ensembles de données analysés en considérant une échelle temporelle égale à $\tau = 1$ seconde.

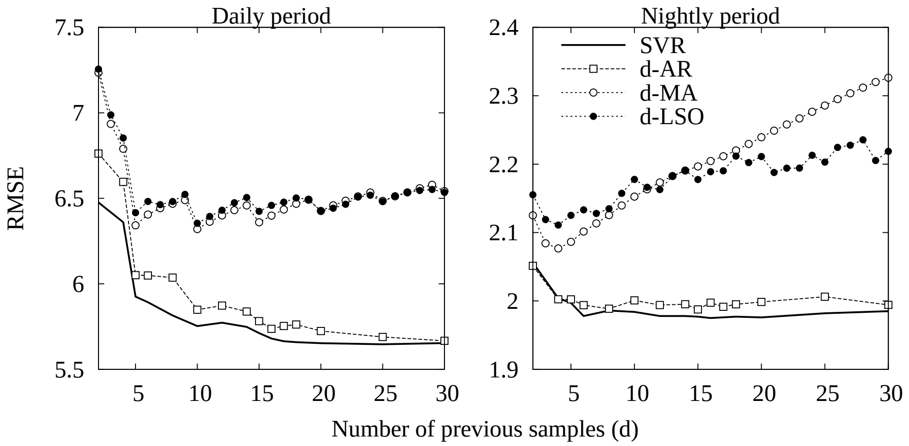


Figure 2: Impact du nombre *d* des observations passées sur l'exactitude de la prédiction.

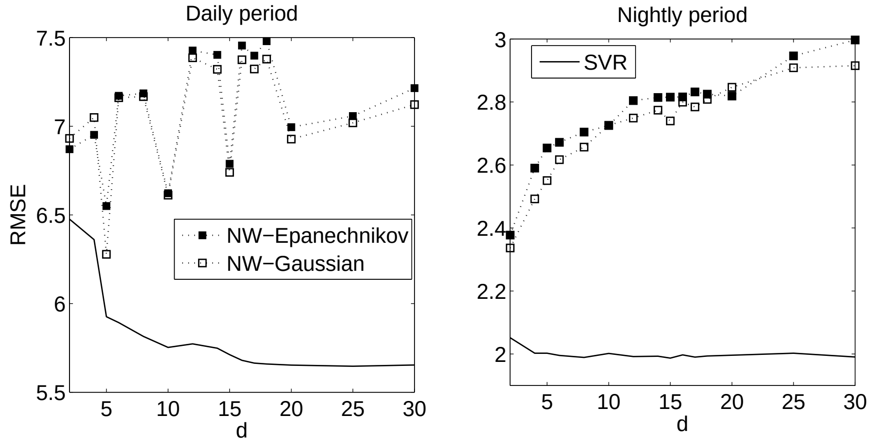
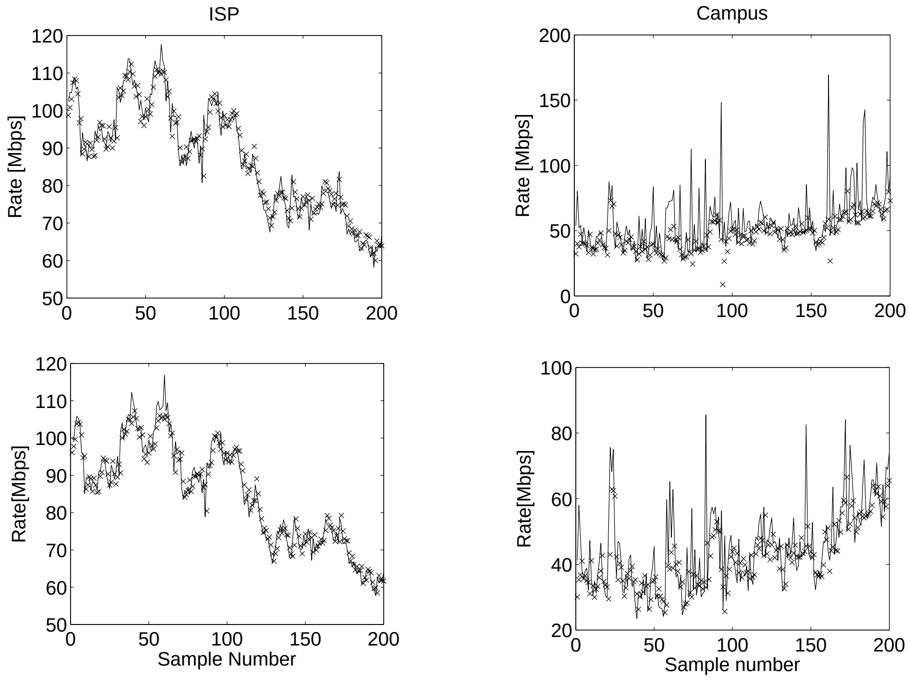


Figure 3: Impact du nombre *d* des observations passées sur l'exactitude de la prédiction de l'estimateur Nadaraya-Watson quand le même ensemble d'apprentissage et de validation que pour SVM est utilisé.

Les résultats pour l'estimateur de NW sont affichés séparément pour faciliter la comparaison. Chaque point correspond au résultat moyen de 10 répétitions de l'expérience. Comme première remarque, on peut affirmer que les résultats sont différents, SVR étant celui qui obtient toujours les meilleurs résultats. Cependant ils sont quantitativement très similaires : en d'autres termes, SVR ne semble pas offrir une amélioration significative, particulièrement par rapport aux modèles d-AR. Il faut avouer que les modèles AR sont en fait équivalents a des modèles de SVR avec un noyau linéal. Cependant, nous devons attribuer à SVR un certain nombre d'aspects extrêmement positifs : par exemple, les algorithmes basés sur SVR sont robustes à la variation des paramètres, et leur complexité informatique est loin d'être prohibitive, propriétés qui les rendent appropriés pour la prédiction en ligne.

Nous étudions également des méthodes pour prolonger l'horizon de prédiction en



(a)

(b)

Figure 4: Comparaison des valeurs réelles et les prédictions associées obtenues par SVR pour les traces (a) ISP and (b) Campus.

utilisant des valeurs prévues comme inputs pour une nouvelle prévision. Intéressante, cette approche de SVR recursive peut, de manière significative, prolonger l'horizon réalisable de prédiction, au prix d'une dégradation très limitée de l'exactitude de la prédiction.

En outre, nous prouvons que la performance de l'algorithme de prédiction peut être améliorée si une approche différente est adoptée. Nous employons toujours SVM mais nous définissons différents inputs/outputs : l'objet de la prédiction dans ce cas-ci est une fonction, par exemple le maximum ou le percentile, de la charge du lien (pendant un intervalle donné de temps) et l'input est constituée par un résumé des propriétés statistiques des observations mesurés dans l'intervalle passé de la même longueur. Dans ce cas-ci, nous prouvons que l'utilisation de plusieurs "machines" en parallèle peut améliorer le gain de performance et en même temps éviter la décision sur le choix d'input optimal.

Nous montrons dans la Fig. 3.15 les valeurs réelles et les prédictions obtenues avec SVR pour le maximum (en haut) et pour le 95th-percentile, pour un ensemble aléatoire de validation pour deux ensembles de données différents (ISP et Campus). Cette figure nous permet d'avoir une première idée (visuelle) de l'exactitude de la prédiction. Dans

		ISP		Campus	
Input		p_{95}	p	p_{95}	p
Naive	**RMSE**	**3.44**	**3.91**	**8.28**	**16.55**
SVM Gain	1st	5.1 % $\circ$	8.2 % $\star$	15.5 % $\bullet$	21.2 % $\bullet$
	2nd	5.0 % $\bullet$	8.0 % $\bullet$	15.1 % $\star$	20.9 % $\star$
	3rd	4.6 % $\star$	7.9 % $\circ$	14.4 % $\circ$	20.8 % $\circ$
	= 2 feat	4.7 %	7.7 %	14.6 %	20.1 %
	$\geq$ 3 feat	3.8 %	6.3 %	13.0 %	15.9 %
	All	4.3 %	7.2 %	13.9 %	18.3 %

Table 1: Comparaison de la prédiction Naive et de SVR pour p et p_{95}, pour les différents inputs ($\circ$=(μ),$\bullet$=(μ, p),$\star$=(μ, p_{95}))

ce cas nous avons utilisé comme input la combinaison (μ, p) (moyenne, maximum). Comme prévu, SVR ne peut pas prévoir les “pics” importants qui sont présents surtout dans les traces nommé Campus, mais il est capable de “suivre” la forme de la courbe. Intuitivement, la présence de ces pics rend la prévision du maximum plus difficile que celui du percentile. Ces différences sont reflétées dans le RMSE : pour la trace ISP, le RMSE est de 3.65 pour le maximum et de 3.39 pour le 95th-percentile, tandis que pour la trace Campus ces valeurs sont respectivement de 21.46 et 9.94 .

Nous avons aussi exploré l’utilisation d’autres inputs que (μ, p). Nous considérerons comme inputs toutes les combinaisons possibles des attributs μ (moyen), σ (écart type), p (maximum) et p_{95} (95th-percentile). Le Tab. 3.4 rapporte le RMSE obtenu par la prédiction naïve (où la prédiction est le maximum ou le percentile observé dans l’intervalle précédent) et le gain correspondant de SVR pour l’ensemble de données ISP et Campus et pour les différentes combinaisons des attributs.

Comme mentionné avant, il est en fait inutile d’inspecter quelle combinaison d’inputs donne les meilleurs résultats : l’idée fondamentale est d’utiliser plusieurs machines en parallèle, dont chacune est apprenti à partir des différents inputs pour le même output (de sorte qu’il est possible par exemple de *combiner* la puissance de prédiction de différentes machines, ou aussi de *choisir automatiquement* la meilleure combinaison d’inputs).

Classification des applications P2P-TV

En ce qui concerne la classification du trafic P2P-TV, nous proposons une méthodologie originale qui se fonde seulement sur le comptage de paquets et d’octets échangés parmi des pairs pendant de petites fenêtres de temps. Notre affirmation est que les différentes applications échangeront différentes quantités d’information au sujet de leur opération particulière (par exemple d’activités de signalisation ou de taille du video chunks), et que cette information exploitée de façon appropriée est sufissante pour identifier l’application particulière. Notre système de classification, qui se sert de SVM, iden-

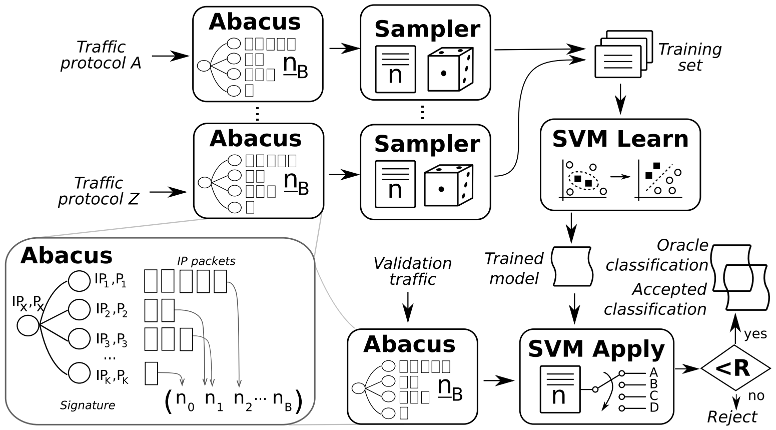


Figure 5: Cadre de classification.

tifie non seulement exactement le trafic P2P-TV (95% du trafic dans le pire des cas est correctement classifié), mais traite également le trafic qui n'est pas produit par des applications de P2P-TV, de sorte que les événements de fausse classification sont négligeables. Une campagne expérimentale très complète est exécutée pour valider nos résultats. Elle emploie des méthodologies actives et passives et est représentative d'un grand ensemble de scénarios possibles. D'ailleurs, nous analysons la portabilité de la signature définie, prouvant que notre cadre peut être prolongé à des situations très différentes, ce qui constitue un avantage très important.

Considérons le trafic reçu par un point final arbitraire $\mathcal{P} = (IP, port)$ pendant un intervalle de temps ΔT. Nous évaluons simplement la quantité d'information reçue par $\mathcal{P}$ par le nombre de paquets reçus. Après nous prolongeons ce concept pour compter également la quantité d'octets.

Nous divisons l'espace $\mathbb{N}$ du nombre possible de paquets envoyés à $\mathcal{P}$ par un autre pair dans $B_n + 1$ caissiers de taille exponentielle avec base 2, c.à.d :

$$I_0 = (0, 1], \quad I_i = (2^{i-1}, 2^i] \ \forall i = 1, \ldots, B_{n-1} \quad \text{et} \quad I_{B_n} = (2^{B_n - 1}, \infty].$$

Pour chaque intervalle ΔT, nous comptons le nombre N_i de pairs qui ont envoyé à $\mathcal{P}$ un certain nombre de paquets $n \in I_i$; i.e., N_0 compte le nombre de pairs qui ont envoyé exactement 1 paquet à $\mathcal{P}$ pendant ΔT ; N_1 compte le nombre de pairs qui ont envoyé 2 paquets; N_2 le nombre de pairs qui ont envoyé 3 ou 4 paquets et N_{B_n} est égal au nombre de pairs qui ont envoyé au moins $2^{B_n - 1} + 1$ paquets à $\mathcal{P}$.

Soit K le nombre de pairs qui sont entrés en contact avec $\mathcal{P}$ dans l'intervalle. La signature comportementale est donc définie par $\underline{n} = (n_0, \ldots, n_{B_n}) \in \mathbb{R}^{B_n + 1}$, où :

$$n_i = \frac{N_i}{\sum\limits_{j=0}^{B_n} N_j} = \frac{N_i}{K}.$$

	PPLive	TVAnts	SopCast	Joost	Unk
PPLive	**81.66**	0.58	9.55	2.32	5.90
TVAnts	0.49	**98.51**	0.18	0.77	0.04
SopCast	3.76	0.11	**89.62**	0.32	6.19
Joost	2.84	0.55	0.28	**89.47**	6.86

	PPLive	TVAnts	SopCast	Joost	TNR
CAMPUS	2.42	2.23	0.01	0.02	**95.3**
ISP	0.66	0.13	0.43	0.10	**98.7**

Table 2: Matrice de confusion pour la classification du trafic P2P-TV.

La signature $\underline{n}$ est la fonction de masse observée du nombre de pairs qui ont envoyé un nombre donné de paquets à $\mathcal{P}$ dans un intervalle de temps ΔT.

On appellera la signature “Abacus” pour ses initiales en anglais : “Automated Behavioral Application Classification Using Signatures”.

Pour produire le modèle de classification, SVM doit être appris a partir des entrées connues, comme il est montré dans la partie de dessous de la Fig. 4.4. Utilisant des traces de banc d'essai, nous produisons des signatures pour des applications connues de P2P-TV; spécifiquement, pour chaque trace, nous construisons une signature Abacus $\underline{n}$ pour chaque intervalle de ΔT secondes. La signature est peut être choisie, au hasard, pour faire partie de l'échantillon d'entraînement. Une fois que le SVM a été appris, l'outil de classification fonctionne comme indiqué dans la partie inférieure de la Fig. 4.4. Des signatures Abacus du trafic à classifier sont calculées et données au modèle de SVM, qui décide quelle application lui a produit : finalement, la classification donne par SVM est acceptée s'il passe un *critère de rejet*, utilisé pour rejeter correctement le trafic qui n'est pas du type P2P-TV.

Seulement pour montrer l'efficacité de notre méthode de classification, nous montrons ici sa performance pour le cas le plus simple, où la signature est calculée a partir du nombre de paquets échangés entre les pairs. Cette performance peut être améliorée si on considère aussi le nombre d'octets en plus du nombre de paquets.

Dans la partie supérieure du Tab. 4.6 on rapporte la performance de notre mécanisme de classification en adoptant une représentation type “matrice de confusion”. Chaque ligne considère le trafic d'une application particulière, et chaque colonne rapporte des résultats de classification. Les valeurs sur la diagonale correspondent aux vrais positifs (accentués en gras), tandis que les éléments en dehors de la diagonale correspondent à des faux négatifs, qui correspondent à des échantillons mal classifiés et à des échantillons rejetés (marqués comme inconnus, Unk). On peut voir que, dans le pire des cas, environ 81% de signatures sont correctement classifiées. L'application la plus difficile à identifier semble être PPLive, qui est confondue avec SopCast (9.55%) ou Joost (2.32%). D'autres applications montrent un plus haut taux de classification positive, avec une classification presque parfaite pour TVAnts. En moyenne, environ 4.5% de signatures de trafic P2P-TV sont rejetées, donc marquées comme inconnues.

La partie inférieure du Tab. 4.6 rapporte des résultats en considérant de traces qui ne contiennent pas de trafic P2P-TV. Le taux de vrai negatives est l'index principal à considérer (les caractères gras dans la colonne plus à droite de la table). Les résultats prouvent que le critère de rejet adopté est très robuste, de sorte que moins de 5% d'échantillons sont mal classés dans le pire des cas. La partie gauche du tableau détaille les faux positifs : PPLive et TVAnts sont la cause de la plupart de fausses classifications, alors que Joost ne cause pratiquement aucun faux positif.

Modèles pour CSMA

Dans la deuxième partie de la thèse, nous traitons deux problèmes différents liés aux mécanismes de contrôle d'accès au support dans les réseaux ad-hoc. Le premier est lié à la modélisation du mécanisme bien connu CSMA/CA. Nous avons observé que pour plusieurs des modèles proposés il n'est pas évident de savoir quelles sont les hypothèses fondamentales et quelle est leur corrélation avec le vrai protocole. Sans viser de un cours d'instruction sur la matière, nous présentons de manière clarifiant les multiples modèles qui, à notre connaissance, ont été proposés dans la littérature pour l'analyse de la performance de CSMA/CA. En particulier, nous discutons quand l'hypothèse sur le mode de CCA peut être dépassée et dans quels cas un modèle basé sur la définition des domaines d'exclusion est approprié.

En particulier, tous les modèles analysés sont basés sur l'hypothèse que la fonction d'atténuation est monotone en la distance entre les noeuds, et pourtant n'inclut pas des effets de fading/shadowing. Nous nous sommes alors concentrés sur la possibilité de généraliser de tels modèles afin qu'ils incluent les variations aléatoires des conditions de canal. En particulier, nous avons proposé une généralisation basée sur la définition des domaines d'exclusion aléatoires (RED). Nous avons montré que des résultats très différents sont obtenus par ce modèle une fois comparés aux précédents. Cette comparaison montre que les effets de fading/shadowing jouent un rôle important dans la performance du protocole et qu'ils doivent être pris en considération.

CED se réfère au modèle qui considère une fonction d'atténuation déterministe, ce qui implique que les domaines d'exclusion sont définis seulement en termes de distance. RED renvoie au modèle comprenant des effets de fading/shadowing qui se traduisent en domaines d'exclusion aléatoires. Dans ce qui suit nous comparons la réutilisation spatiale et le débit moyen obtenu par les deux modèles pour différentes valeurs de l'exposant α de la fonction d'atténuation (voir Fig. 5.6 et Fig. 5.7).

Plus que les résultats quantitatifs, le but de cette comparaison est d'indiquer que des résultats qualitativement différents sont obtenus avec les deux modèles considérés.

Cependant, certaines des techniques employées par les propositions analysées, qui permettent d'obtenir des résultats analytiques, ne peuvent pas être généralisées à ce cas plus général (par exemple le packing formalisme ou la troncature d'un processus

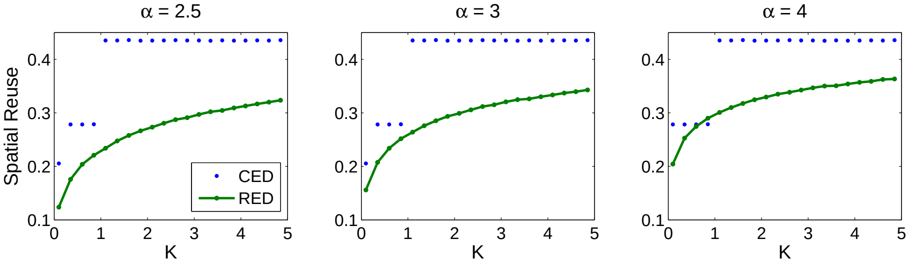


Figure 6: Comparaison de la réutilisation spatiale obtenue pour CED et RED avec l'hypothpèse de shadowing Lognormal.

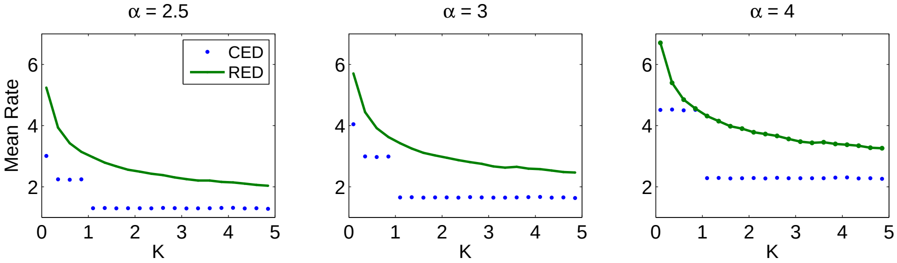
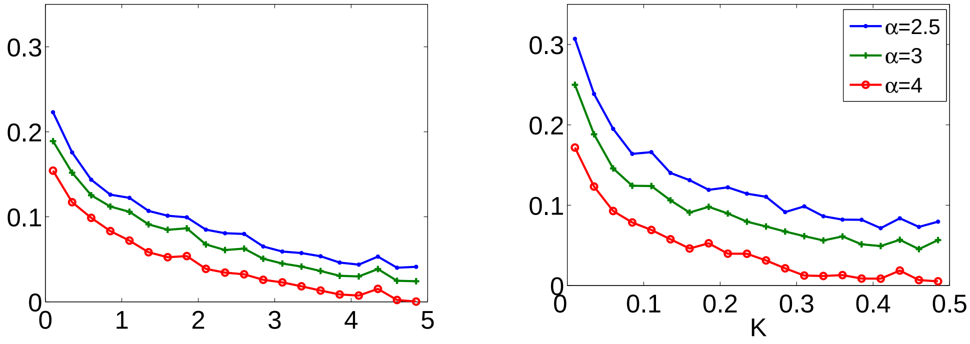


Figure 7: Comparaison du taux moyen obtenu pour CED et RED avec l'hypothpèse de shadowing Lognormal.

libre). Des résultats analytiques (même avec l'hypothèse d'une fonction d'atténuation déterministe) peuvent être obtenus seulement pour une topologie simple comme la ligne. Le seul modèle capable de fournir des résultats analytiques pour des topologies plus générales et pour des canaux aléatoires est un modèle de type Matèrn [20] (au moins à notre connaissance). Malheureusement, ces modèles sont connus pour être intrinsèquement conservateurs. Nous avons mesuré quelle est la magnitude de ce conservatisme, constatant que la différence relative est d'environ 30% en termes de réutilisation spatiale et taux moyen. Lorsque la densité spatiale du taux est considérée, les différences sont réduites: moins de 10% dans beaucoup de cas (particulièrement sous le modèle de shadowing Lognormal). En dépit de ces différences quantitatives, les résultats obtenus sont qualitativement semblables. La dépendance des indicateurs de performance (par exemple la réutilisation spatiale et le débit moyen) à l'égard du paramètre K est presque identique dans les deux modèles. La différence entre les résultats obtenus est grossièrement constante pour toutes les valeurs de K. Voir par exemple la Fig. 5.14, qui montre la différence entre les modèles RED et MRED c.à.d, la version Matèrn du modèle RED qui prend en compte les effets aléatoires du canal.

Une hypothèse fondamentale des modèles analysés est qu'ils supposent que le CCA est exécuté dans le mode Carrier Detection (CD). Cependant, d'autres modes peuvent être considérés, par exemple le mode Energy Detection (ED). Considérant que pour le premier cas le canal est rapporté comme occupé si au moins un signal est détecté, pour le second la condition est définie par rapport à la puissance totale reçue (interférence).



(a) Lognormal shadowing

(b) Rayleigh fading

Figure 8: Différence relative entre RED et MRED pour la densité spatiale de taux considérant shadowing Lognormal (a) ou fading Rayleigh (b).

Pour ce dernier cas, tous les modèles précédemment présentés ne sont plus valides. Nous proposons alors un cadre analytique, basé sur les outils provenant de la géométrie aléatoire, qui nous permet de fournir une comparaison profonde entre ces deux modes. En particulier, des formules analytiques sont données pour la probabilité d'accès pour les deux modes considérés. In mélange de ces deux modes est également analysé.

Supposant que l'ensemble des noeuds d'un réseau est la réalisation d'un processus ponctuel suivant une distribution Poisson dans $\mathbb{R}^2$, nous avons prouvé que l'ensemble d'émetteurs actifs sous ces deux modes (CD et ED) peut être rapproché par des shot noise extrêmes ou additifs (ESN et ASN) associés au processus ponctuel original. Ce cadre nous a permis de calculer la probabilité d'accès pour les deux cas et de mesurer les différences entre elles.

Φ_{add} et Φ_{ext} sont des processus ponctuel qui représentent l'ensemble d'émetteurs qui peuvent être actifs pour le ED et le CD mode respectivement. Nous avons supposé par défaut à fading Rayleigh et la fonction d'atténuation suivante : $l(r) = (Ar)^{-\beta}$ tel que $\beta > 2$. Nous avons aussi considéré deux cas différents : seuils déterministes et aléatoires pour définir l'accès au canal. On note λp_I et λp_M la densité du processus Φ_{add} et Φ_{ext} respectivement, pour chaque cas on indiquera si le seuil est aléatoire (rand) ou deterministe (det).

Puisque la somme est toujours plus grande que le maximum il devrait être clair que l'intensité de Φ_{add} sera toujours plus petite que celle de Φ_{ext}. Plus précisément, on a prouvé que la différence entre les deux intensités (pour λ suffisamment grand) est un facteur constant :

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_M^{rand}} = \frac{1}{\Gamma(1+2/\beta)\Gamma(1-2/\beta)} < 1. \tag{2}$$

Ce facteur, qui dépend seulement de l'exposant β de la fonction d'atténuation, est toujours plus petit que 1. Il peut être vu comme la perte sur la densité des émetteurs

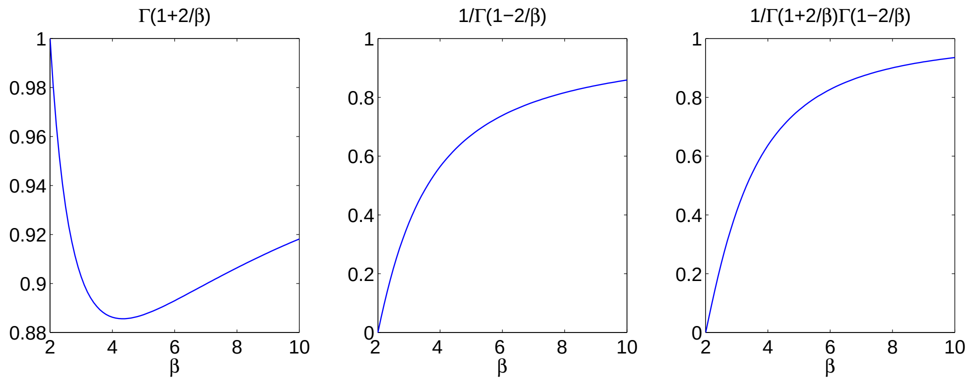


Figure 9: Fonctions de limite

actifs pour considérer l'interférence totale reçue par le noeud qui compte accéder au canal, au lieu de la puissance de réception maximale, pour décider si le noeud peut être activé ou pas (c.àd. pour considérer le CCA en mode de détection d'énergie au lieu de détection de porteur).

Quand $\beta \to \infty$ les deux fonctions $\Gamma(1 + 2/\beta)$ and $\Gamma(1 - 2/\beta)$ convergent vers 1. Ceci signifie que les intensités de Φ_{ext} et Φ_{add} seront égales. D'ailleurs, la fonction $1/\Gamma(1 + 2/\beta)\Gamma(1 - 2/\beta)$ est monotone croissante, ce qui signifie que la différence entre les intensités diminue avec β. Ces conclusions ne sont pas étonnantes : pour des valeurs grandes de β l'impact de chaque shot est moins significatif et ainsi la différence entre le maximum et la somme est moins importante. Cependant, (6.28) donne une quantification de cette différence quand l'intensité du processus original est suffisamment grande.

Dans la partie la plus à droite de la Fig. 6.2, on représente la fonction $1/\Gamma(1 + 2/\beta)\Gamma(1 - 2/\beta)$. Comme nous pouvons observer la limite est approchée pour des valeurs très grandes de β qui ne sont pas des choix raisonnables dans notre contexte. D'ailleurs les valeurs obtenues pour des valeurs de β près de 2 sont très petites ce qui implique que les différences correspondantes sur les intensités sont très grandes. Par exemple, pour $\beta = 2.1$ et $\beta = 4$ les valeurs obtenues sont respectivement 0.05 et 0.64. Ceci signifie que l'intensité asymptotique obtenue pour le shot noise additif sera respectivement 5% et 64% de cela obtenu par le shot noise extrême.

Si maintenant on considère le cas où le seuil est déterministe, nous pouvons obtenir des résultats analytiques pour le shot noise additif seulement pour $\beta = 4$. Dans ce cas, la même relation que pour le cas aléatoire peut être prouvé, c.à.d. que pour $\beta = 4$ on a la relation suivante :

$$\lim_{\lambda \to \infty} \frac{\lambda p_I^{det}}{\lambda p_M^{det}} = \frac{1}{\Gamma(1 - 2/\beta)\Gamma(1 + 2/\beta)} < 1. \tag{3}$$

Même si le résultat est prouvé seulement pour $\beta = 4$ nous faisons la conjecture de qu'il est en fait valide pour toutes les valeurs de β. Nous validons notre conjecture par

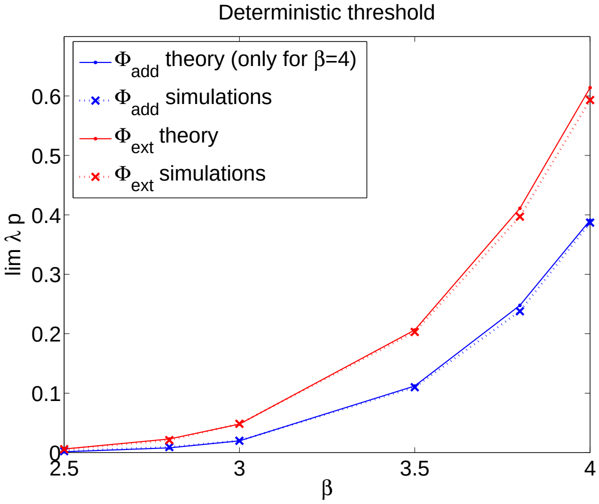


Figure 10: Validation de la conjecture.

des simulations comme le montre la Fig. 10. Les mêmes conclusions que pour le cas précédent sont donc valides.

Nous avons aussi comparé la relation entre le cas déterministe et le cas aléatoire, et nous avons obtenu le même résultat pour le shot noise additif que pour l'extrémal :

$$\lim_{\lambda\to\infty}\frac{\lambda p_I^{rand}}{\lambda p_I^{det}} = \lim_{\lambda\to\infty}\frac{\lambda p_M^{rand}}{\lambda p_M^{det}} = \Gamma(1+2/\beta). \tag{4}$$

Si nous nous concentrons sur la partie gauche de la Fig. 6.2, qui donne la relation entre le cas aléatoire et déterministe pour les deux processus, on peut observer que le minimum est atteint à $\beta = 4$ pour lequel la différence est de 12%. D'ailleurs, pour des valeurs plus petites que $\beta = 3$, la fonction est plus grande que 0.9, ce qui implique dont une différence de moins de 10%. Ceci signifie que la différence entre considérer un seuil aléatoire ou déterministe est marginale.

Finalement, des résultats obtenus en considérant une condition pour la somme et le maximum en même temps ont été également déduits et comparés aux modes précédents. Une observation importante est que la condition sur la somme est plus restrictive, du moins quand des seuils comparables sont considérés.

Contrôle d'accès au support avec des garanties de performance

La deuxième partie de cette thèse est consacrée à la conception et à l'évaluation quantitative des mécanismes de MAC avec des garanties de performance. Par ceci, nous voulons dire des mécanismes où chaque connexion admise obtient un taux minimum ou d'une manière équivalente un SINR (rapport signal sur bruit et interférence) minimum. Deux mécanismes sont définis et comparés. On propose ces mécanismes dans

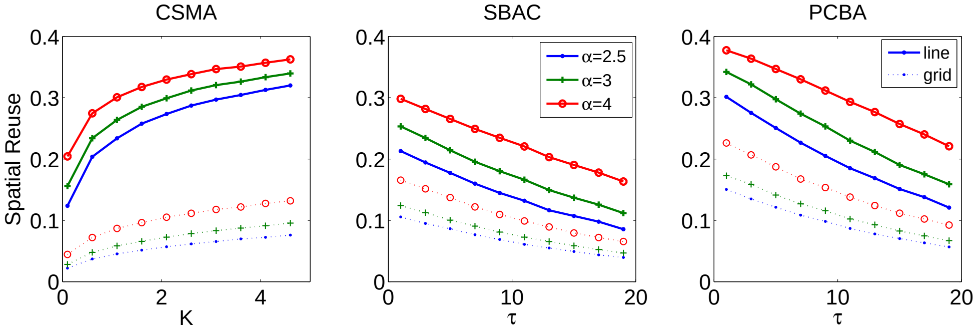


Figure 11: Comparaison de la réutilisation spatiale pour une topologie en ligne (lignes continues) et en grille (lignes pointillés) pour le shadowing Lognormal.

le cadre des réseaux ad hoc et pourtant, ils doivent être décentralisés : les problèmes d'implementation de cette contrainte sont analysées. Les deux mécanismes prennent en considération le niveau d'interférence pour décider sur l'ensemble de connexions qui peuvent accéder au canal de façon simultanée dans un moment donné. La différence principale entre les deux mécanismes est la possibilité d'ajuster la puissance de transmission des noeuds. Une comparaison approfondi de la performance de ces deux mécanismes et du CSMA/CA est présentée, basée sur un mélange de modèles analytiques et des simulations, et sur un ensemble complet de métriques de performance qui inclut la réutilisation spatiale et l'efficacité dans l'utilisation de la puissance de transmission. Différentes topologies de réseau, différents environnements de propagation et scénarios du trafic sont considérés. Dans tous les scénarios analysés, un des mécanismes proposés surpasse de manière significative CSMA, sauf pour quelques cas où les différences ne sont pas significatives. La distribution des taux obtenus est également plus juste pour chacun des mécanismes proposés que pour CSMA. Cependant, celui qui parmi eux est le "meilleur" dépend fortement du scénario de trafic.

Les mécanismes proposés sont nommés SBAC (pour SINR Based Access Control) et PCBA (pour Power Control Based Access). Dans SBAC, la puissance est constante et égale à P pour chaque émetteur. Une nouvelle connexion est acceptée si et seulement si le SINR que elle obtient (qui dépend des connexions déjà admises) est plus grand qu'un certaine seuil, et en même temps, le nouveau SINR que les transmissions déjà actives obtiennent, en prenant en considération ce nouveau connexion, est également plus grand que le seuil. Au lieu de simplement vérifier si le SINR est acceptable, nous supposerons maintenant que la puissance de transmission peut être ajustée. Dans notre deuxième proposition (PCBA), les connexions seront acceptées tant qu'il existe un vecteur faisable de puissance (c.à.d. puissance des connexions déjà actives plus la nouvelle) garantissant que le SINR obtenu pour toutes les transmissions admises est plus grand que le minimum.

Dans les figures Fig. 7.2 et Fig. 7.3 nous montrons la comparaison de CSMA, SBAC et PCBA en termes de réutilisation spatiale (SR) et de taux moyen (MR) pour deux topologies différentes (ligne et grille) quand le shadowing est supposé Lognormal.

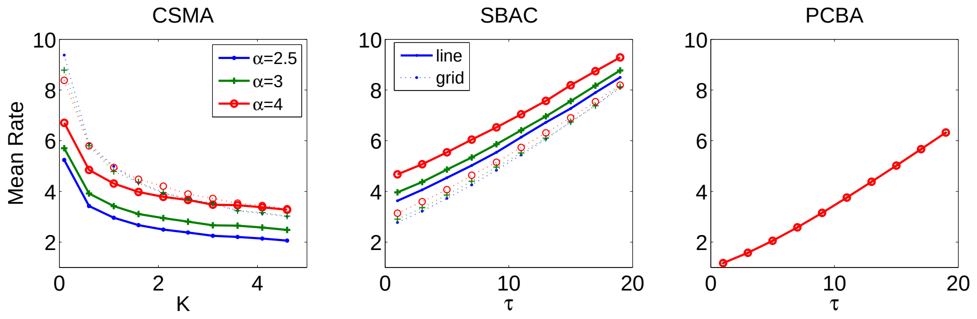


Figure 12: Comparaison de la taux moyenne ($\rho = \log_2(1 + \text{SINR})$) pour une topologie en ligne (lignes continues) et en grille (lignes pointillés) pour shadowing Lognormal.

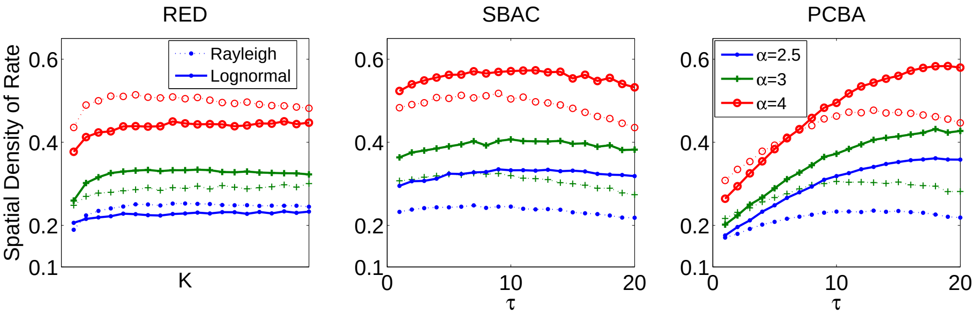


Figure 13: Densité spatiale de taux pour une topologie en grille pour Rayleigh fading (lignes continues) et pour Lognormal shadowing (lignes pointillés.

Concernant l'impact de la topologie sur le SR et le MR, l'analyse comparative entre les mécanismes proposés et CSMA (or RED) est semblable pour les deux topologies. On peut observer des différences principalement dans le SR, qui est plus petit pour la topologie en grille due à l'augmentation des potentiels interférents : dans la ligne chaque noeud loin du bord a deux voisins plus proches, qui deviennent quatre dans la grille. Cependant, l'impact sur le MR est moins significatif. Le nombre peu élevé des connexions actives simultanées a comme conséquence des niveaux de taux moyens semblables à ceux obtenus sur la ligne.

Il y a un compromis évident entre le MR et le SR. Par exemple, si nous choisissons la valeur de K qui maximise le SR, nous obtiendrons des valeurs petites pour le MR. Au contraire, si nous choisissons K afin de maximiser le MR nous obtiendrons un SR très faible. Quel est alors un bon choix de K qui considère les deux paramètres en même temps ? Pour évaluer plus exactement cette différence, nous considérons plusieurs fonctions d'utilité, selon le type de trafic présent dans le réseau. En particulier nous nous concentrons sur trois types de trafic : le trafic élastique (par exemple données), le trafic élastique avec un minimum SINR exigé (ce qu'on apprécierait beaucoup pour le trafic de données dans LANs sans fil fortement chargé) et le trafic à taux d'échantillonnage fixe (CBR pour ce sigle en anglais : Constant Bit Rate) (par exemple trafic téléphonique).

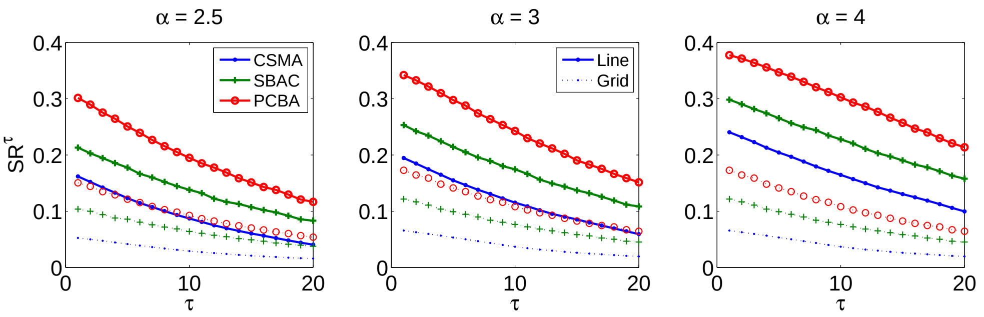


Figure 14: Réutilisation spatiale quand SINR $\geq \tau$ pour des topologies en ligne et grille avec shadowing Lognormal.

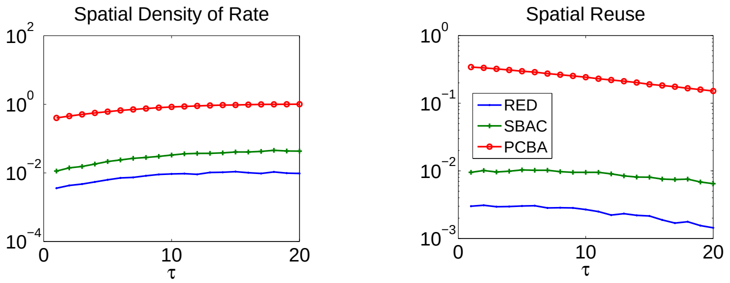


Figure 15: Densité spatiale de taux et réutilisation spatiale quand SINR $= \tau$ pour la topologie en ligne avec shadowing Lognormal et $\alpha = 3$.

Si le trafic élastique est considéré, SBAC est l'algorithme qui fournit les meilleurs résultats. Quand un taux minimum doit être garanti, le meilleur dépend de la métrique considéré, du niveau de taux minimum et aussi de α. Si la comparaison est faite en termes de réutilisation spatiale, PCBA surpasse largement le reste des mécanismes. Cette supériorité est dû à sa capacité de contrôler la puissance de transmission, ce qui le permet d'accepter plus des connexions simultanés. En même temps, puisqu'il donne exactement le même taux à tous les connexions, son performance diminue quand une autre métrique (qui inclut explicitement le taux) est considérée. Par exemple, SBAC est le mécanisme qui fournit la meilleure densité spatiale du taux. Exemples de ce type de comparaison sont montrés dans les figures Fig. 7.5 et Fig. 7.10.

PCBA obtient également les meilleurs résultats quand le trafic CBR est considéré, indépendamment de la métrique et du modèle de propagation. Quand on considère le rapport entre le taux et la puissance de transmission, PCBA obtient encore des résultats très bons tant que la distance entre l'émetteur et le récepteur reste limitée. La Fig. 7.11 montre la comparaison en termes de réutilisation spatiale et de densité spatiale de taux, pour le cas de trafic CBR pour un cas particulier représentatif des cas plus généraux.

Contents

Acronyms List

Notation	**Meaning**
ABACUS	Automated Behavioral Application Classification Using Signatures
ADSL	Asymmetric Digital Subscriber Line
ALR	Adaptive Link Rate
AR	Auto Regressive
ASN	Additive Shot Noise
CA	Collision Avoidance
CCA	Clear Channel Assessment
CDMA	Code Division Multiple Access
CED	Constant Exclusion Domain
CD	Carrier Detection
CoV	Coefficient of Variation
CPU	Central Processor Unit
CSMA	Carrier Sensing Multiple Access
CTS	Clear To Send
ED	Energy Detection
ESN	Extremal Shot Noise
FDMA	Frequency Division Multiple Access
FPR	Faux Positive Rate
FTTH	Fiber To The Home
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LRD	Long Range Dependence
MA	Moving Average
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
NAV	Network Vector Allocation
NC	Number of Contenders
NW	Nadaraya-Watson
OFDMA	Orthogonal Frequency Division Multiple Access

PCBA	Power Control Based Access
POP	Point of Presence
PSTN	Packet Switch Telephone Network
P2P	Peer to Peer
P2P-TV	Peer to Peer Television
QoS	Quality of Service
RE	Relative Error
RED	Random Exclusion Domain
RMSE	Root Mean Square Error
RTP	Real Time Protocol
RTS	Ready To Send
SBAC	SINR Based Access Control
SN	Shot Noise
SR	Spatial Reuse
SVM	Support Vector Machines
SVR	Support Vector Regression
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TPR	True Positive Rate
UDP	User Datagram Protocol
VideoIP	Video over IP
VoIP	Voice over IP
VPN	Virtual Private Network

Chapter 1

Introduction

1.1 Motivation

Internet is the first real multi-service network; voice, video and data services are all offered on the same network. Among the many consequences that this convergence brought with it, the change on the traffic structure is probably between the most prominent ones. Not only has the amount of traffic increased, but its nature has radically changed too, mainly due to the offer of new advanced services (with a socio-economic impact that is incontestably important). The popularity of these services, such as for instance file-sharing (e.g. KaZaa, eDonkey, BitTorrent), Internet telephony (e.g. Skype, Gtalk, VoIPbuster) and Internet television (e.g. Sopcast, Joost, Zatoo, Babelgum) has exploded in the last years. As a result, network data traffic is increasingly complex and dynamic.

These two characteristics translate into a high traffic unpredictability (at least with traditional and simple techniques). Predicting traffic volume is very useful for both capacity planning and self-management schemes (even if they involve very different time scales). For instance, a typical application of this prediction is the dynamic resource reservation in the context of Virtual Private Networks (VPN) provisioning [1]. Another application, related to the emerging discipline of *green networking* [2], is the Adaptive Link Rate (ALR) problem. ALR is proposed as a way to reduce energy consumption by means of adjusting the configured link rate on the router to the minimum required. It would be interesting then to design a prediction scheme that accurately predicts such variable traffic in an online manner, and at the same time minimizes the assumptions on the traffic structure (e.g. no assumption on the statistical properties of the measurements should be made).

However, the prediction of some traffic features as the mean or the maximum over a given time interval does not give information about the traffic itself. Certain traffic-aware applications require to actually identify the type of traffic traversing the network, or even the application generating it. For instance, traffic (application) classification is instrumental to the application of QoS policies, is necessary to block specific banned applications (e.g. P2P or chat in an office), or to identify popular applications with commercial purposes (e.g. advertising). An outstanding example of these new types of traffic is P2P-TV (Peer-to-Peer Television); large-scale real-time video-streaming services exploiting the peer-to-peer (P2P) communication paradigm. There are several currently deployed P2P-TV systems which feature low-quality streaming [3, 4, 5, 6], but high-quality systems will be soon of widespread use too [7, 8].

P2P-TV systems are candidates for becoming the origin of serious problems for the Internet, as testified by the growing success of commercial systems such as PPLive, SopCast, TVAnts and many others. Indeed, P2P-TV traffic may potentially grow without control, causing a degradation of the quality of service perceived by Internet users or even the network collapse [9]. While downlink is limited by the stream rate, uplink may grow unboundedly as observed in [10]. Furthermore, the proprietary (and closed) design of some successful P2P-TV applications turns the identification of such applications in a topic of growing importance. For instance, an ISP (Internet Service Provider) will certainly be interested in blocking these resource consuming applications, or at least in identifying them.

Although the traffic classification problem is not new, for different reasons some well known techniques are becoming obsolete. Examples of these are port-based or payload-based classification. Whereas the former one can be misled by, for instance, firewall blocking or dynamic port allocation, the computational complexity of the latter may be as high as to make it not implementable. As an alternative, very good performances are obtained by means of statistical classification [11, 12]. In addition, behavioural classification, based on the rationale that different applications generate different patterns, was used in the context of coarse-grained classification of Internet hosts [13, 14]. Thus, we interest ourselves in the design of a classification engine targeted to P2P-TV applications (i.e. fine-grained) based on the behavioural paradigm.

Another characteristic of today networks related to the above considerations is the use of new emerging technologies. Indeed, as traffic injected to the network is increasingly heterogeneous, so are the access technologies. In particular, wireless have increasingly become the edge technology of choice. It is expected that in the near future, with the proliferation of smart devices with new functionalities, communication between wireless terminals will usually take place without the core being involved. These terminals will act in a self-organized and adaptive manner, falling in the category of Mobile Ad-hoc NETworks (MANETs).

At the moment of analyzing the design or the performance of MANETs, the shared nature of the medium appears as one of the most difficult aspects to cope with. As such, the design of the Medium Access Control (MAC) mechanism plays a key role. Ideally, a well designed MAC algorithm allows the maximum number of simultaneous transmissions such that they do not interfere with each other. All of this, minimizing unfairness in the access opportunities. However, such design is a really difficult task when a decentralized mechanism is considered (which is the ideal case). The MAC mechanism is then ran locally by each node, which has only local information of the network state. The most widely deployed MAC mechanism is CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), used for instance in IEEE 802.11 and IEEE 802.15.4. Even if it has been relatively well studied from an empirical point of view, there is no widely accepted mathematical model for it. In this sense, an analysis of the protocol, and the identification of its relevant features for modeling purposes is, surprisingly, missing. For instance, current models do not take into account the randomness of channel conditions due to fading/shadowing effects [21, 22]. Furthermore, some aspect of CSMA/CA are simply ignored, as the Clear Channel Assessment (CCA). In this context we analyze whether these models can be extended to the non deterministic case, or if it is necessary and possible to define a new model for CSMA/CA.

In any case, CSMA/CA does not guarantee anything in terms of the quality obtained by the accepted transmissions. This lack of QoS (Quality of Service) could hinder the implementation of, for instance, advanced real-time applications. It would then be interesting to design and analyze a control access mechanism with performance guarantees. As mentioned above, the design of such a mechanism must cope with the decentralized nature and the shared resources (where the QoS of each connections depends on *all* active connections) of MANETs.

In a nutshell, this thesis presents four main contributions:

- We analyze the problem of link load prediction, and propose a new algorithm based on a novel Machine Learning technique, the so-called Support Vector Machines (SVM) [15]. Our proposal is compared with several parametric and non-parametric techniques, obtaining always the best results.

- A SVM variant is used on the design of a fine-grained P2P-TV application classification engine. The performance of our proposal is validated by testing it on several different traffic traces representative of a wide range of scenarios, from which excellent results were obtained.

- We analyze and discuss possible models for the different flavours of CSMA/CA, putting special emphasis on the correlation between the model assumptions and the real protocol features. A special section is devoted to the modeling and comparison of the different CCA modes.

- We propose two different access mechanisms that guarantee a minimum QoS (in terms of rate) for all accepted transmissions. The performance of these new mechanisms is compared with the classic CSMA/CA in several scenarios, considering various performance metrics. The main aim of our study is to identify which of the proposed mechanisms performs better depending on the scenario.

In the following section, we detail the main contributions of the thesis.

1.2 Main Contributions

This thesis is organized in two parts. The first one is dedicated to the design of tools based on Machine Learning, in particular SVM [15, 16], to cope with two of the problems previously mentioned: link load prediction and P2P-TV traffic classification. SVM are a set of methods for classification and regression that are grounded in the framework of statistical learning theory. Even if the first work on this topic dates from the seventies [17], it only gained the international academic community attention in the nineties. SVM have been extensively used mainly in the context of pattern recognition in which they have shown very good performance. In the context of networking applications, they have been used, for instance, for anomaly detection or throughput estimation [18, 19]. We interested ourselves in these techniques since they work well in many learning situations, due to its good generalization to unseen data. Moreover, they are amenable to continuous and adaptive online learning which constitutes an extremely desirable property for our purposes.

As a first step, we perform a deep and comprehensive study of the SVM techniques. We then analyze in detail its performance when applied to the above mentioned networking problems. As a general remark, we can say that in both cases SVM proved to be a very useful and powerful tool. Whereas for the link load prediction, SVM robustness and low computational complexity appear as its main assets, for the P2P-TV classification SVM provides impressively good results.

More precisely, for the link load prediction, we follow the methodology known as "embedding procedure". A time series of averaged link load values at the chosen time scale is considered, and a future value of this time series is predicted based only on a small number of past observations. In particular, we choose for this work small time scales, i.e. less than one minute. An extensive sensitivity analysis is presented which shows the SVM robustness in terms of parameter tuning and training, as well as its cost-effectiveness. Its accuracy is gathered by means of a thorough comparison phase that includes parametric and non-parametric models, ranging from simply moving averages to more sophisticated models such as the Nadaraya-Watson estimator [23]. We found that in any case, SVM provides the best results.

Furthermore, we show that the performance of the predictor can be improved if a different approach is taken. We still use SVM but we define different inputs/outputs: the prediction is in this case focused on a function of the link load (e.g. the maximum or percentile value over a given time interval) and the input is constituted by a summary of statistical properties of the observations taken in a past interval of equal length. In this case, we show that the use of several "machines" in parallel can further improve the performance gain and avoids the decision of a particular input choice.

Concerning P2P-TV traffic classification, we propose a novel methodology that relies only on the count of packets and bytes exchanged among peers during small time-windows. Our claim is that the different applications will exchange different amounts of information concerning their particular operation (e.g. signaling activities or video chunk size), and that this information conveniently exploited is enough to identify the particular application. Our classification framework, which makes use of SVM, not only accurately identifies P2P-TV traffic (95% of correctly classified traffic in the worst case), but also handles traffic that is not generated by P2P-TV applications, so that false classification events are negligible. A thorough experimental campaign is performed to validate our results, which uses both active and passive methodologies and is representative of a large set of possible scenarios. Moreover, we analyze the portability of the defined signature, showing that our framework can be extended to very different situations, which constitutes a very important benefit.

In the second part of the thesis, we deal with two different problems related to the multiple access mechanisms in ad-hoc networks. The first one is related to the modeling of the well known CSMA/CA mechanism. We found out that in many of the proposed models it is not clear which are the underlying assumptions and their correlation with the real protocol. Without aiming at providing a tutorial on the topic, we present a clarifying picture of the several models that, to the best of our knowledge, have been proposed in the literature for the analysis of the performance of CSMA/CA. In particular we discuss when the assumption on the CCA mode can be overridden and in which cases a model based on the definition of exclusion domains is appropriate.

In addition, we propose a new model for CSMA/CA that takes into account the shadowing/fading effects, and we compare it with previous proposals. This comparison allows us to prove that this aspect plays a major role in the performance of the mechanism. Furthermore, we show that models originated in the stochastic geometry context provide analytical formulae for some important performance indicators, such as the access probability or the spatial reuse. These models are known to be conservative, but up to now a quantitative analysis of the magnitude of this difference is still lacking. We analyze this gap assuming a slotted division of time for both deterministic and random attenuation functions. We show that, even if significant differences can appear in terms of spatial reuse (and so in the rate), when other performance metrics that include both the spatial reuse and rate are considered, these differences are reduced. Moreover, it should be said that results obtained by Matérn like models are qualitative similar to real ones.

An underlying hypothesis of the analyzed models is that they assume that the CCA is performed in Carrier Detection (CD) mode. However, other modes can be considered, for instance Energy Detection (ED) mode. Whereas for the first case the channel is reported as busy if at least one signal is detected, for the second one the condition is over the total received power (interference). For this last case, all previously introduced models are no longer valid. We propose then an analytical framework, based on stochastic geometry tools, that allows us to provide a deep comparison between these two modes. In particular, analytical formulae is given for the access probability under both considered modes. Moreover, a mixture of these two modes is also analyzed.

The second part of this thesis is also dedicated to the design and the quantitative evaluation of MAC mechanisms with performance guarantees. By this, we mean mechanisms where each accepted connection obtains a minimum rate or equivalently a minimum SINR (Signal to Interference and Noise Ratio). Two such mechanisms are defined and compared. These mechanisms are proposed in the context of ad-hoc networks and as such they must be decentralized: the implementations issues of this constraint are analyzed. Both mechanisms take the interference level into account to decide on the set of connections which can access the shared channel at any given time. The main difference between the two is the possibility or not of adjusting the transmission power of the nodes. A deep comparison of the performance of these two mechanisms and CSMA/CA is presented, based on a mix of analytical models and simulations, and on a comprehensive set of performance metrics which include spatial reuse and power efficiency. Different network topologies, propagation environments and traffic scenarios are considered. In all analyzed scenarios, one of the proposed mechanisms significantly outperforms CSMA, apart from a few cases where the differences are not significant. The rate distribution is also more fair for each of the proposed mechanisms than for CSMA. However, which of them is the “best” one strongly depends on the traffic scenario.

1.3 Outline

A brief summary of the main attributes of SVM is presented in Chap. 2. We choose to avoid mathematical proofs that could make this chapter hard to follow. We present instead the main results of this theory that are required for its understanding, and the emphasis is put on those characteristics that will be useful later for our specific applications. In particular, SVM can be divided into classification or regression according to the output type considered: discrete or continuous. Both cases are analyzed separately. The multiclass classification case is particularly emphasized since our application falls in this category.

The application of SVM in its regression mode (referred to as SVR) to the link load problem will be analyzed in Chap. 3. In this chapter, two different approaches are presented: the “embedding procedure” and the prediction of a function of the link load. In the first case, much effort is dedicated to the parameter selection and the comparison with several different techniques. The sensitivity analysis includes a thorough and careful tuning of SVR, but also the analysis of other aspects that influence its performance as, for instance, the training set size. Further details on the temporal evolution of the error, the computational complexity and the recursive use of SVR are also discussed in this chapter.

With the objective of improving the performance of the embedding procedure, we then analyze the possibility of predicting a function of the link load (e.g. the maximum). In particular we propose to use as input to fed SVR a vector with the summary of statistical properties of past observations. The use of several machines in parallel, i.e. different machines trained with the same output but with different inputs, is analyzed. In order to gather robust results we consider several real-world traffic traces, representative of very different network scenarios (such as ISP, Ethernet, WiFi LAN and enterprise networks) so as to be able to print out a fine-grained picture of the gain brought by SVR.

A classification framework tailored for P2P-TV traffic is defined and analyzed in Chap. 4. The classification engine defined in this chapter is based on a signature constructed from the count of packets and bytes that peers exchange between them, which we claim and show to be enough to identify the application that generated the traffic. A very important aspect of this kind of problems is to correctly validate the results, and special attention is dedicated to the treatment of non P2P-TV traffic. In our case, very different datasets are considered, which are representative of a wide range of possible scenarios. Another aspect, as important as the previous one, is the portability of the defined signature. In particular, we analyze this portability across different network sites, access technologies, channel popularity, time and network conditions.

Chapters 5 and 6 are dedicated to the analysis of the MAC mechanism for wireless ad-hoc networks. More precisely, the modeling of CSMA/CA is considered in Chap. 5. Firstly, a comparison of some of the most "popular" models is presented, with special emphasis on the correlation between the model assumptions and the real protocol features. In particular, Matérn like models (conceived on the stochastic geometry context) are introduced, for which analytical results of performance indicators are obtained. Secondly, a model that includes shadowing/fading effects is defined and compared with other proposals that do not include it. Finally, we address the problem of quantifying the subestimation of Matérn like models we mentioned above.

Chapter 6 is devoted to the modeling and comparison of the different CCA modes. Our analysis is based on a key element in the context of stochastic geometry: the shot noise (SN). As we will see, both modes of the CCA (ED and CD) can be described in terms of an additive or an extremal shot noise. This framework allows us to calculate the access probability of a typical node and so the density of nodes that can access the channel at the same time. Moreover, a quantification of the impact of considering the total interference instead of the maximum received power is provided. Finally, a third mode consisting on a combination of the previous ones is analyzed.

In Chap. 7 two MAC mechanisms with performance guarantees are defined and studied. A thorough comparison is performed, mainly based on simulations, including different scenarios and metrics defined accordingly. A special section is devoted to the discussion of the implementation issues, where we analyze the complexity of each of the proposed algorithms and discuss the question of whether they can be implemented in a decentralized and realistic way.

Finally, Chapter 8 concludes this dissertation and presents directions for future work. A list of publications disseminating the material covered in this thesis can be found in Appendix A.

Part I

Machine Learning for Traffic Prediction and Classification

Chapter 2

Support Vector Machines

In this section we introduce the basics ideas behind Support Vector Machines (SVM) [15, 16]. SVM represent a set of classification and regression techniques, that are grounded in the framework of statistical learning theory. Despite they were introduce in the seventies [17] only in the nineties attract the attention of the academic community. Since then, SVM have shown very good performance, for both classification and regression problems and in very different research areas. For the pattern recognition case, SVM have been extensively used for isolated handwritten digit recognition [24, 25], object recognition [26], face detection in images [27, 28] or text categorization [29]. But they have been also used more recently in the networking context [18, 30]. For the regression case, SVM have been used for time series prediction and benchmarked with known test, showing that in most cases the performance is similar or is significantly better than that of competitors methods [31, 32, 33]. In this case there exist also successful results in networking [19, 34].

We will apply SVM for two different networking problems. Firstly we will use SVM for regression to predict the link load over a given timescale based only in past measurements of its load. Results are presented in Chap. 3. Secondly, in Chap 4, we analyze the problem of P2P-TV traffic applications classification by means of SVM.

We present here definitions and results according to Vapnik's theory [15, 16]. We start by giving the context of the learning theory, in particular the (structural) risk minimization which is at the origin of SVM. We present only the main results without proofs. Then, we will introduce the classification problem; in particular the simpler case of linear separable data will be analyzed in detail. We will see that the general case of non linear machines trained with non separable data can be solved in a very similar way. Finally, we present how SVM can be used in the regression context. In each case, we will concentrate on the most relevant notions and parameters whose impact we investigate in detail later on and that are instrumental to the understanding of the performance evaluation of chapters 3 and 4.

2.1 Vapnik's Learning Theory

For any learning task, with a given finite amount of training data, there is a main tradeoff between the accuracy obtained for that particular training set and the "capacity" of the machine, that is the ability to learn any training set without error. As perfectly explained in [35]: "A machine with too much capacity is like a botanist with a photographic memory who, when presented with a new tree, concludes that it is not a tree because it has a different number of leaves from anything seen before; a machine with little capacity is like the botanist's lazy brother, who declares that if it's green, it's a tree. Neither can generalized well". The formalization of these concepts are central in the statistical learning theory of Vapnik and we will try to present here its main features.

Let us consider a training set, i.e. a set of observations $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x \in S_X$ is the input and $y \in S_Y$ is the output. Suppose that there exist some unknown probability distribution form which data is drawn, i.e. they are *iid*, independent and identically distributed according to $P(x, y) = P(y|x)P(x)$.

The objective is to define an algorithm (learning machine) that given a new input X predicts the output $\widehat{Y}$. In general the output $\widehat{Y}$ is found by minimizing some risk functional. If the output Y is discrete the problem is a classification one, being a regression problem if Y is continuous. Let be $L : S_X \times S_Y \times S_Y \to \mathbb{R}^+$ a risk (cost) function,

$$L(x, u, y) = \text{ cost of deciding } u \text{ for input } x \text{ when the real output is } y.$$

Different risk functions can be defined according to the type of problem. In the classification problem with binary output, the objective is to minimize the number of errors and so the risk function is simply $L(x, u, y) = 1_{\{u \neq y\}}$. Alternatively, if the output is a real variable and the problem is to find an approximation that minimizes the quadratic error, the risk function will be $L(x, u, y) = ||u - y||^2$.

Let $\mathcal{F} = \{f : S_X \to S_Y\}$ be the set of all possible functions $f : S_X \to S_Y$. A learning machine is a function $f \in \mathcal{F}$, i.e. given an input x returns an output y. We look for the function $f^* \in \mathcal{F}$ that minimizes the expected risk $R(f)$:

$$R(f) = E(L(X, f(X), Y)) = \int_{S_X \times S_Y} L(x, f(x), y) dP(x, y).$$

Then, the function f^* verifies that

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, R(f).$$

However the expected risk can not be calculated since the distribution $P(y|x)$ is not known. Instead an empirical risk is calculated from the training set:

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} L(x_i, f(x_i), y_i),$$

for which the dependence on the probability distribution disappears. The solution is defined then as the function f_n^* that minimizes the empirical risk function $R_n(f)$:

$$f_n^* = \operatorname*{argmin}_{f \in \mathcal{F}} R_n(f). \tag{2.1}$$

We will refer to this solution as ERM (Error Risk Minimization) principle. One may wonder about the validity of this principle as learning method (i.e. is f_n^* a good approximation of f^*? or more important, is $R_n(f_n^*)$ a good approximation of $R(f^*)$?), and many of the Vapnik's results are devoted to prove it. We present here only the main results, for which some definitions need to be introduced.

Definition 2.1.1. The ERM principle is *consistent* if *:*

$$R(f_n) \quad \underset{n\to\infty}{\overset{p}{\to}} \quad \inf_{f \in \mathcal{F}} R(f) \tag{2.2}$$

$$R_n(f_n) \quad \underset{n\to\infty}{\overset{p}{\to}} \quad \inf_{f \in \mathcal{F}} R(f) \tag{2.3}$$

which means that the expected risk and the empirical risk for f_n converge in probability when n goes to infinity to the same limit that its the minimum value of the risk $R(f)$ $(R(f^*))$.

Vapnik [15] proves the following result, referred as to the Fundamental Theorem of Learning.

Theorem 2.1.2. Suppose that there exist constants a and b such that $a \leq L \leq b$ for all $f \in \mathcal{F}$. Then the following statements are equivalent:

1. The ERM principle is consistent
2. The empirical risk $R_n(f)$ converges towards the expected risk $R(f)$ in the sense that:

$$\lim_{n\to\infty} P\left(\sup_{f \in \mathcal{F}}(R_n(f) - R(f)) > \epsilon\right) = 0 \quad \forall \epsilon > 0$$

Remark 2.1.3. A simply case where it can be easily proved that the second condition holds is when S_X and S_Y are finite. In this case the family $\mathcal{F}$ is also finite, and by the strong law of large number it results that $R_n(f)$ converges uniformly and almost surely to $R(f)$ and so the second condition is verified.

However, this theoretical results can not be directly applied since the risk $R(f)$ can not be calculated (remember that the distribution probability $P(x, y)$ is unknown). Instead other sufficient conditions are deduced that can be used in practice. The central idea is that the second condition of the previous theorem is verified if the family $\mathcal{F}$ is not too "big". Different measures are then introduced for instance the ϵ-entropy or the Vapnik-Chervonenkis (VC) dimension. There exist a lot of results based on these notions, but we present here only those results that are able to synthesize the main features of the theory.

Theorem 2.1.4. If the VC-dimension of the set $\mathcal{F}$ is finite, then the ERM principle is consistent.

Furthermore, if the risk function is bounded $a \leq L \leq b$, then for a given η, with probability $1 - \eta$ the following bound holds:

$$R(f) \leq R_n(f) + (b-a)\sqrt{\frac{h(\log(2n/h) + 1) - \log(\eta/4)}{n}}, \tag{2.4}$$

where h is the VC-dimension of $\mathcal{F}$. The second term is referred as to VC-confidence.

Remark 2.1.5. Consider a binary classification problem, i.e. the output y is either 1 or -1. If the risk function is defined as $L(x, f(x), y) = |y - f(x)|$, then $0 \leq L \leq 1$ and the previous bound holds with $b - a = 1$.

It can be seen that the bound does not depend on $P(x, y)$, it simply assumes that the observations are drawn independently according to some distribution. Also, and as mentioned before, the left hand side can not be calculated. However, if h is known (which it is the case for a certain number of models) then the second term of the right hand can be easily calculated. Then, by choosing a small η if we choose the learning machine that minimizes the right hand side, we are choosing the machine that gives the minimum upper bound to the expected risk. This gives a method for selecting a learning machine for a given task, and is the basis for the Structural Risk Minimization (SRM) that we will introduce later.

It is important to note that in case the VC-dimension h is high, nothing can be ensure about the performance of a selected machine, i.e. a high VC-dimension does not necessarily implies high values of the expected risk. On the other hand, if for a given fixed family of functions to choose from the bound is tight for at least one of them, then it is not possible to do better. Moreover, since we only have an upper bound, it is possible that a particular machine with the same empirical risk that belongs to a family set with higher VC-dimension, obtains better performance. Some examples in this sense are analyzed in [35].

2.1.1 The VC Dimension

To give more intuition to this notion, we limit ourselves here to a family of functions f that take values in $S_Y = \{-1, 1\}$. Given a set of n points, we say that the family $\mathcal{F}$ can *shatter* them if for each possible labeling (there are 2^n possible labelings), there is a function of the family that correctly assigns those labels. The VC-dimension of $\mathcal{F}$ is defined as the maximum number of points that can be shattered by it. For example, if the VC-dimension is h it means that there exist at least one set of h points that can be shattered; but it is not necessarily true for all set of h points.

Example 2.1.6.

1. Consider the family of characteristic functions of type $1_{[a,+\infty]}$. It is clear that two points in the real line can not be shattered by this family: if the left point is classified as 1 then the right point will be also classified as 1. Then the VC-dimension of this family is 1.

2. Consider points in $\mathbb{R}^2$ and a family of functions defined by straight lines, so that points on each side of the line are assigned to different classes. Clearly two points can be shattered by this family of functions. Some drawing must convince the reader that three points can be also shattered. However it is not possible to shatter four points. Then the VC-dimension of this family is 3.

3. The VC-dimension of the set of oriented hyperplanes in $\mathbb{R}^n$ is $n+1$ (see [36]).

It is important to highlight that the VC-dimension is in fact different that the number of parameters of a given model. An example of a learning machine with only one parameter and infinite VC-dimension is constructed in [15].

2.1.2 Structural Risk Minimization

If we concentrate on the right side of (2.4), we see that if the ratio $\frac{n}{h}$ is large enough then the term preponderant will be the empirical error. In this way the minimization of the empirical risk it is enough to ensure small values of the expected risk. However if this ratio is small (for instance the training set is small), then the VC-confidence plays a more important role and the simple minimization of the empirical risk may be not enough to ensure good performance. Since in general the training set size can be limited for real world applications, a way to ensure a small expected risk is to control the VC-dimension h (still remember that high values of h does not necessarily imply bad performance). Vapnik proposes then to use a different principle called Structural Risk Minimization (SRM).

The main idea is that whereas the expected and the empirical risk depends only on the selected function choose by the training procedure, the VC-confidence depends on the family of functions. Then, the objective is to find a subset of this family such that the bound for this subset is minimized. Since h is an integer (and so its variation is not smooth), a structure is introduced by dividing the family into nested subsets for which we can calculate (or at least bound) the VC-dimension. The SRM principle consist for instance in training a set of machines, one per subset with the objective of minimize the empirical risk. Then, the machine that minimizes the sum of the empirical risk and the VC-confidence between them, is selected as the solution.

The final step in the Vapnik learning theory is the definition of efficient algorithms that control the generalization capacity of the models they produce. It is on the basis of the SRM that Vapnik proposes Support Vector Machines.

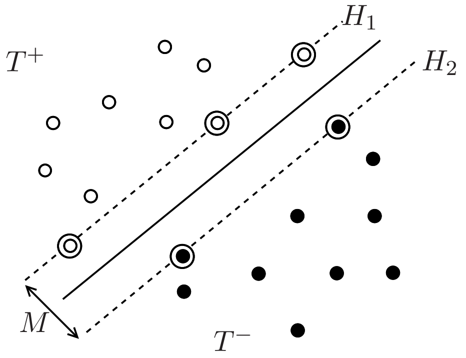


Figure 2.1: SVM for linear separable data.

2.2 SVM for Classification

We devote this section to analyze in detail the binary classification problem for which the output $y \in \{-1, +1\}$. In particular we will start with the simplest case where the data are separable, which means that there is a hyperplane that separates the data with label -1 from those with label 1.

2.2.1 The Linear Separable Case

Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the training set where $x \in \mathbb{R}^d$ and $y \in \{-1, +1\}$. Suppose that there exist a "separating" hyperplane, i.e. and hyperplane that separates the positives from the negative data.

Let H be the hyperplane defined by $H = \Big\{x \in \mathbb{R}^d : f(x) = \langle w, x\rangle + b = 0\Big\}$, where $\langle , \rangle$ represents a dot product in $\mathbb{R}^d$, $w \in \mathbb{R}^d$ with $||w|| = 1$ (w is the normal vector) and $b \in \mathbb{R}$, being $\frac{|b|}{||w||}$ the distance (perpendicular) to the origin. Let be $T^+ = \{x_i \in \mathbb{R}^d : y_i = 1\}$ and $T^- = \{x_i \in \mathbb{R}^d : y_i = 1\}$. The assumption that the data are separable by H means that the sign of f is constant over T^+ and T^-.

Let d_+ (d_-) be the distance from H to the nearest positive (negative) data. The margin of H is defined as $M = d_+ + d_-$ (see Fig. 2.1). We look for a separating hyperplane that maximizes the margin. This can be formulated as follows. Suppose that all points in the training set verifies:

$$\langle x_i, w\rangle + b \geq 1 \quad \text{if} \quad y_i = 1, \tag{2.5}$$

$$\langle x_i, w\rangle + b \leq -1 \quad \text{if} \quad y_i = -1, \tag{2.6}$$

which can also be expressed as:

$$y_i(\langle x_i, w\rangle + b) - 1 \geq 0 \quad \forall i. \tag{2.7}$$

The points that verify the equality in (2.5) belong to the hyperplane $H_1 = \{x : \langle x, w\rangle + b = 1\}$, which is an hyperplane with normal w and distance to the origin $\frac{|1-b|}{||w||}$. Analogously, points that verify the equality in (2.6), belong to the hyperplane $H_2 = \{x : \langle x, w\rangle + b = -1\}$ whose distance to the origin is $\frac{|-1-b|}{||w||}$. Hence $d_+ = d_- = \frac{1}{||w||}$ and the margin is $\frac{2}{||w||}$. The problem of maximizing the margin is then equivalent to minimize $||w||$ (equivalently $\frac{1}{2}||w||^2$ which simplifies future calculus) subject to constraints (2.7):

$$(P) = \begin{cases} \text{minimize} & \frac{1}{2}||w||^2 \\ \text{subject to} & y_i(\langle x_i, w\rangle + b) - 1 \geq 0 \quad \forall i. \end{cases}$$

This problem is a convex quadratic programming problem, and so there exist a unique optimum. Thus, the KKT (Karush-Khun-Tucker) conditions [37] are necessary and sufficient. This means that a point (w^*, b^*) is an optimal solution for (P) if and only if, there exist values $\{\lambda_i\}_{i=1,\ldots,n}$ such that the following conditions are satisfied:

1. $y_i(\langle x_i, w^*\rangle + b^*) - 1 \geq 0$
2. $\lambda_i \geq 0$
3. $\lambda_i(y_i(\langle x_i, w^*\rangle + b^*) - 1) = 0$
4. $\nabla L(w^*, b^*, \lambda) = 0$, where $L(w^*, b^*, \lambda) = \frac{1}{2}||w^*||^2 - \sum\limits_{i=1}^{n} \lambda_i(y_i(\langle x_i, w^*\rangle + b^*) - 1)$ is the Lagrangian associated to (P)

The values λ_i are called Lagrange multipliers. From condition 4, we obtain that:

$$\begin{aligned} \frac{\partial L(w^*, b^*, \lambda)}{\partial w_j^*} &= w_j^* - \sum_{i=1}^{n} \lambda_i y_i x_i^j = 0 \quad \Rightarrow \quad w^* = \sum_{i=1}^{n} \lambda_i y_i x_i \\ \frac{\partial L(w^*, b^*, \lambda)}{\partial b^*} &= -\sum_{i=1}^{n} \lambda_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \lambda_i y_i = 0 \end{aligned} \tag{2.8}$$

From these equations we obtain an explicit expression for w^*, but not for b^*. However b^* can be easily obtained from condition 3, by choosing any value $\lambda_{i_0} \neq 0$:

$$b^* = \frac{1 - y_{i_0}\langle x_{i_0}, w^*\rangle}{y_{i_0}}$$

For numerical reasons, it is safer to compute b^* as the mean value resulting from all such equations.

Observe that there is a Lagrange multiplier λ_i for each training point (x_i, y_i), but the solution (w^*, b^*) is determined only for those training points which corresponding multipliers are positive. These points are called “Support Vectors”. If $\lambda_i > 0$, condition 3 implies that $y_i(\langle x_i, w^*\rangle + b^*) - 1 = 0$, which means that the point (x_i, y_i)

lies in the hyperplane H_1 or H_2 (see Fig. 2.1 where the support vectors are indicated with extra circles). The important points are the support vectors, if any of the other points (with $\lambda_i = 0$) is removed (or even moved but without changing of subspace), the solution does not change: the same separating hyperplane will be obtained after the training process.

Then, the solution can be written as:

$$f(x) = \langle w^*, x\rangle + b^* = \sum_{i=1}^{n} \lambda_i y_i \langle x_i^*, x\rangle + b^* = \sum_{j=1}^{n_{SV}} \lambda_j y_j \langle x_j, x\rangle + b^*, \tag{2.9}$$

where n_{SV} is the number of support vectors. This means that the solution depends on the number of the support vectors and not in the training set size. It can be seen that training (x_i) and test points (x) are related only by its inner product. As we will see later, this property is the key to the non linear extension of the procedure.

Despite we have obtained an expression for the optimal separating hyperplane, the problem is not solved: we need to find the values λ_i ! Finding solutions for real world problem usually require the use of numerical methods. However we will show how this can be done by writing problem (P) in its dual version. Let $L^D(w, b, \lambda)$ be the Lagrange dual function:

$$L^D(w, b, \lambda) = \inf_{w,b} L(w, b, \lambda)$$

The function L is convex in the variables w, b then its has an minimum that can be founded by imposing that its partial derivatives must vanish at that point. These derivatives are the same we already calculate in (2.8), and by replacing them in the definition of L we obtain that:

$$L^D(w, b, \lambda) = -\frac{1}{2} \sum_{i,j=1}^{n} \lambda_i \lambda_j y_i y_j \langle x_i, x_j\rangle + \sum_{i=1}^{n} \lambda_i \tag{2.10}$$

Since the original problem (P) is convex (the objective function in (P) is convex and the constraints are linear), it is equivalent to solve its dual version:

$$(D) = \begin{cases} \text{maximize} & L^D(w, b, \lambda) \\ \text{subject to} & \lambda_i \geq 0 \quad \forall i \\ & \sum\limits_{i=1}^{n} \lambda_i y_i = 0. \end{cases}$$

The dual problem (D) is also a convex optimization problem, since the objective function is concave and the constraints are linear. This is always the case no matter whether or not the original problem (P) is convex. This means that the dual problem always has a solution that can be founded also by means of the KKT conditions. We will see later some methods for solving this problem for more general cases. Note that the solution of the SVM training is always global (there is no other feasible point at which the objective function takes a lower value); this is in contrast to neural networks where many local minima usually exist.

Once we have trained a Support Vector Machine (i.e. we found w^* and b^*), a new data x will be classified accordingly to on which side of the decision boundary it lies, or what it is the same, accordingly to the sign of the function $f(x)$.

2.2.2 The Linear Non Separable Case

In this section we will show that a similar solution is obtained when there is not an hyperplane that perfectly separates positives from negatives data. In this case we cannot apply the same procedure as before. The idea is to relax conditions (2.5) and (2.6), but only when necessary. Positive slack variables $\{\xi_i\}_{i=1,\dots,n}$ are then introduced as a cost that will increase the objective function of (P):

$$\begin{array}{rcll} \langle x_i, w\rangle + b & \geq & 1-\xi_i & \text{if} \quad y_i = 1, \\ \langle x_i, w\rangle + b & \leq & -1+\xi_i & \text{if} \quad y_i = -1, \\ \xi_i & \geq & 0, & \end{array}$$

which can also be expressed as:

$$y_i(\langle x_i, w\rangle + b) - 1 + \xi_i \geq 0 \quad \text{and} \quad \xi_i \geq 0 \quad \forall i.$$

When an error occurs the corresponding ξ_i must be larger than one, then $\sum_{i=1}^n \xi_i$ is an upper bound for the number of training errors. The objective function is changed to consider an extra cost due to errors as follows:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i,$$

where C is a constat chosen by the user to assign more or less penalty to errors (we will evaluate its impact over the performance in Chapters 3 and 4). Then training the Support Vector Machine in this case correspond to the following, again quadratic convex programming problem:

$$(P) = \begin{cases} \text{minimize} & \frac{1}{2}||w||^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{subject to} & y_i(\langle x_i, w\rangle + b) - 1 + \xi_i \geq 0 \quad \forall i \\ & \xi_i \geq 0 \quad \forall i. \end{cases}$$

As in the previous section the KKT conditions are necessary and sufficient to find the optimum. Let $L(w, b, \xi, \lambda, \mu)$ the Lagrangian associated to (P):

$$L(w, b, \xi, \lambda, \mu) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\lambda_i(y_i(\langle x_i, w\rangle + b) - 1 + \xi_i) - \sum_{i=1}^{n}\mu_i\xi_i.$$

Then the KKT conditions are:

1. $y_i(\langle x_i, w^*\rangle + b^*) - 1 + \xi_i^* \geq 0$ and $\xi_i^* \geq 0$ $\forall i$

2. $\lambda_i \geq 0$ and $\mu_i \geq 0 \ \forall i$

3. $\lambda_i(y_i(\langle x_i, w^* \rangle + b^*) - 1 + \xi_i^*) = 0 \ \forall i$

4. $\mu_i \xi_i^* = 0 \ \forall i$

5. $\nabla L(w^*, b^*, \xi^*, \lambda, \mu) = 0$

Null gradient for the Lagrangian (last condition) implies that:

$$\begin{aligned}
\frac{\partial L(w^*, b^*, \xi^*, \lambda)}{\partial w_j^*} &= w_j^* - \sum_{i=1}^{n} \lambda_i y_i x_i^j = 0 \quad \Rightarrow \quad w^* = \sum_{i=1}^{n} \lambda_i y_i x_i \\
\frac{\partial L(w^*, b^*, \xi^*, \lambda)}{\partial b^*} &= -\sum_{i=1}^{n} \lambda_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \lambda_i y_i = 0 \qquad (2.11) \\
\frac{\partial L(w^*, b^*, \xi^*, \lambda)}{\partial \xi_j^*} &= C - \lambda_j - \mu_j = 0 \quad \Rightarrow \quad \lambda_j + \mu_j = C. \qquad (2.12)
\end{aligned}$$

We found in this case that w^* has the same form as before. To calculate b^* it is enough to use condition (3) with $0 < \lambda_{i_0} < C$: inequality $\lambda_{i_0} < C$ implies that $\mu_{i_0} > 0$ (see condition (2.11)) and from condition 4 we obtain that $\xi_i^* = 0$. As before it is recommended to use the average over all such equations.

The solution can be written then as before:

$$f(x) = \sum_{i=1}^{n_{SV}} \lambda_i y_i \langle x_i, x \rangle + b^*, \qquad (2.13)$$

Finally, to find λ_i and μ_i, again the dual problem of (P) is considered. It is easy to verify that the dual Lagrangian function is given by (2.10) and that only the constraints on λ_i change:

$$(D) = \begin{cases} \text{maximize} & L^D(w, b, \lambda) \\ \text{subject to} & 0 \leq \lambda_i \leq C \quad \forall i \\ & \sum_{i=1}^{n} \lambda_i y_i = 0. \end{cases}$$

Thus the only difference with the separable case is that the λ_i have now an upper bound equal to C.

2.2.3 The Nonlinear Case

In this section we show how the procedure introduced for the linear case can be easily extended to the non linear case, i.e. to the case where the decision function is not a linear function of the data. In [38], authors show that a quite old technique [39], can be used to treat this case in a straightforward way. The fundamental idea to define a

Support Vector Machine in this case is that we can transform the original data space into a high dimensional space such that in this new space the data become separable for a linear function. The main result of [38] that allows to apply this idea in practice is that it is possible to find the separating hyperplane in this new space without considering it explicitly or the transformation itself. This results is based in the fact that the training data appears on the solution only by means of its dot product $\langle x_i, x_j\rangle$. It is enough to known the dot product between support vectors and vectors of the new space. Let us explain this idea more in detail.

Suppose that data is transformed by a map $\Phi : R^d \to \mathcal{H}$ where $\mathcal{H}$ is an Euclidean space (possibly infinite dimensional). Then the training algorithm would depend on the dot products of the mapped data $\langle\Phi(x_i), \Phi(x_j)\rangle$. Let us define $K(x_i, x_j) = \langle\Phi(x_i), \Phi(x_j)\rangle$. If we replace $\langle x, y\rangle$ by $K(x, y)$ in the training algorithm, it will produce a machine that lives in $\mathcal{F}$ and all the previous holds since it is a linear separation but in a different space. Then the solution will only depend on K and takes the form:

$$f(x) = \sum_{i=1}^{n_{SV}} \lambda_i y_i K(x_i, x) + b^*, \tag{2.14}$$

Example 2.2.1. Let us give a simple yet illustrative example; suppose that data live in $\mathbb{R}^2$ and that are separable by a quadratic function. If we consider the map $\Phi(x) : \mathbb{R}^2 \to \mathbb{R}^3$ such that $\Phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, then a linear function of $\Phi(x)$ yields a quadratic function of x. In this case $K(x_i, x_j) = \langle x_i, x_j\rangle^2$. Note also that different maps or spaces $\mathcal{H}$ could result in the same function K: consider for instance $\mathcal{H} = \mathbb{R}^3$ and $\Phi(x) = 1/\sqrt{2}(x_1^2 - x_2^2, 2x_1x_2, x_1^2 + x_2^2)$, or $\Phi(x) = (x_1^2, x_1x_2, x_1x_2, x_2^2)$ with $\mathcal{H} = \mathbb{R}^4$.

However the approach of the previous example becomes computationally infeasible for higher dimensions and other solutions must be founded. The question is whether we can forgot about the map Φ and consider only the function K. The function K can not be any function, since it must represent an inner product in the space $\mathcal{H}$, this means that K must be a "kernel function". So the question turns to be for which kernel functions there exist a pair $(\mathcal{H}, \Phi)$ with the properties described above. The answer is given by Mercer's condition [40]: there exists a mapping Φ such that $K(x, y) = \sum_i \Phi(x)_i\Phi(y)_i$ in some feature space $\mathcal{H}$ if and only if:

$$\int K(x, y)g(x)g(y)dxdy \geq 0 \quad \text{for any function } g(x) \text{ such that } \int g(x)^2dx < \infty.$$

It must be noted that this condition (which it is not trivial to apply) allows us to verify whether or not a given function is a kernel, but it says nothing about the mapping Φ or even the space $\mathcal{H}$.

In particular, some examples of kernels are the following:

- Linear Kernel $K(x, y) = \langle x, y\rangle$: produces a linear classifier.

- Gaussian Kernel $K(x, y) = e^{-\gamma||x-y||^2}$: produces a Gaussian radial basis function classifier (translation invariant).
- Polynomial Kernel $K(x, y) = (\langle x, y\rangle + 1)^p$: produces a classifier that is a polynomial of degree p.
- Hyperbolic Tangent Kernel: $K(x, y) = \tanh(\kappa\langle x, y\rangle - \delta)$, only for some values of κ, δ and of data $||x||$ (resemblance with neural networks).

It must be noted that both training and test functions depend on the data only through de kernel function K. Even if the kernel represent an inner product in $\mathcal{H}$ whose dimension can be high, the complexity of computing K itself may be far smaller. For instance, for the homogeneous polynomial kernel $K(x, y) = (\langle x, y\rangle)^p$, an inner product in $\mathcal{H}$ requires order of C_p^{d+p-1} operations, whereas the computation of K only requires $O(d)$ operations. It is this special property that allows to construct hyperplanes in very high dimension spaces with tractable number of computations. This property in addition to the fact that SVM depend only on the number of support vectors (and not on the whole training set), shows how SVM avoid well known forms of "curse of dimensionality". However, the choice of the best kernel for a given problem is still an open research issue.

2.2.4 Multi-Class Classification

So far, only binary classifiers have been considered, however they can be combined to include the multiclass case. We devote here some attention to this case since in Chap. 4 we will apply SVM in its multiclass version. The passage from two class to k classes is not so simple and different methods to do it has been proposed. We described here some of them; from their comparison it seems that methods of kind "one-versus-one" are more suitable than "one-versus-rest" or methods that consider all classes simultaneously.

In "one-versus-rest" implementation of SVM multiclass [38] with k classes, k models are constructed. The ith machine is trained with data corresponding to the ith class which is labeled as positive, whereas all other data is labeled as negative. This correspond to train k machines each one with a training set of size n (the original training set size). Given a new input, the assigned class will be the one that maximizes the value of the decision function $f_i(x)$ $i = 1, \dots, k$.

The method called "one-versus-one" [41], constructs $N = k(k-1)/2$ machines where each one is trained with data corresponding only to two different classes. The decision for a new input is usually done by voting; for each machine one vote is added for the assigned class and the final decision correspond to the class with the maximum number of votes. If all classes are equally represented (each class has more or less k/n data in the training set) then this method imply to solve $k(k-1)/2$ quadratic programming each one with a training set of $2n/k$ points. This method is computationally more expensive than one-versus-rest. Moreover, each machine is trained with data corresponding to two classes but they will be applied to any class point. Still, a defense of this approach is presented in [42].

Another approach correspond to the directed acyclic graph SVM (DAGSVM) proposed in [43]. For this method the training is as in the "one-versus-one" method i.e. N machines are trained. For the test phase, a rooted binary directed acyclic graph with N internal nodes and k leaves is constructed; each node is a binary machine between two classes. Given a new data, at each node a binary decision is taking that indicate the next node and the final decision corresponds to the reached leaf node.

Solutions to the multiclass SVM problem in one single optimization problem are proposed in [16, 44]. The idea is similar to the "one-versus-rest" approach; there are k decision functions but they are all obtained by solving only one optimization problem (instead of k).

A comparison of the above methods is presented in [45]; the authors conclude that "one-versus-one" and DAGSVM must be preferred over the other methods. Furthermore it is the former method that it is generally used in the available implementations of SVM such as MySVM [46] or LibSVM [47].

2.3 SVM for Regression

SVM can be also used to function approximation [48], and it is referred to as Support Vector Regression (SVR). In ϵ-SVR [15] the goal is to find a function $f(x)$ whose deviation from each target $y_i \in \mathbb{R}$ is at most ϵ for all training data, and at the same time, is as "flat" as possible. See Fig. 2.2(a). Again for the sake of clarity, we consider the linear case since we already know that it can be easily extended to the non linear case by introducing a kernel.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be such that:

$$f(x) = \langle w, x \rangle + b, \qquad \text{with} \qquad x \in \mathbb{R}^d,\ b \in \mathbb{R} \tag{2.15}$$

Flatness in the case of (2.15) can be ensured by minimizing the norm $\|w\|^2$, leading to the following convex optimization problem:

$$(P) = \begin{cases} \text{minimize} & \frac{1}{2}\|w\|^2 + C \sum\limits_{i=1}^{S} (\xi_i + \xi_i^*) \\ \text{subject to} & y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \forall i \\ & -y_i + \langle w, x_i \rangle + b \leq \epsilon + \xi_i^* \forall i \\ & \xi_i, \xi_i^* \geq 0 \forall i \end{cases} \tag{2.16}$$

In the above formulation, slack variables ξ_i and ξ_i^* are included to cope with otherwise infeasible constraint of the optimization problem, whereas the constant $C > 0$ determines the trade off between the flatness of f and deviations from target greater

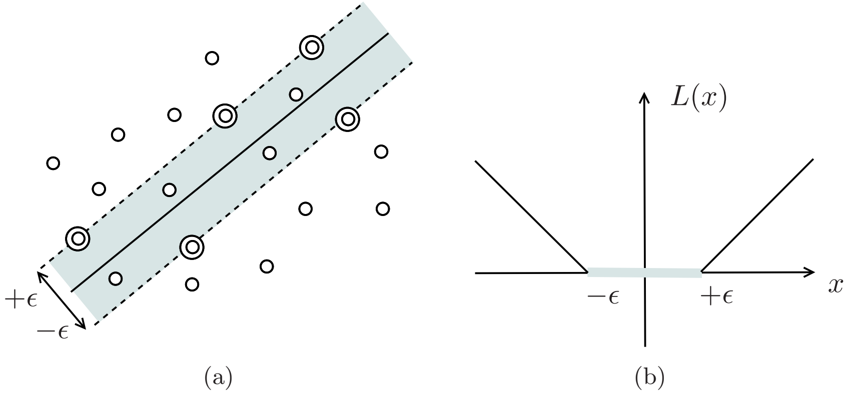


Figure 2.2: (a) SVM for linear regression and (b) ϵ-insensitive loss function.

than ϵ. Notice that this tradeoff makes SVM rather different from traditional error minimization problems, and very robust to outliers.

The above formulation is equivalent to the use of the ϵ-insensitive loss (see Fig. 2.2(b)) function in the theory of error risk minimization with regularization [48]. This means that problem (P) is equivalent to minimize:

$$R_{reg}(f) = R_{emp}(f) + \frac{\alpha}{2}||w||^2, \tag{2.17}$$

where $\alpha > 0$ is a regularization constant. The rationale is that if the family of functions $\mathcal{F}$ is very rich (for instance, small training set in very higher dimensional space), it can lead to overfitting and bad generalization properties of the solution; then a control term is added, in this case it is $|w||^2$.

Loss function (2.18) represents the fact that there is no loss (or cost) for deviations smaller than ϵ and that larger deviations will be linearly penalized.

$$L(x) = \begin{cases} 0, & \text{if } |\xi| \leq \varepsilon \\ |x| - \varepsilon, & \text{otherwise.} \end{cases} \tag{2.18}$$

It is easy to show that minimizing (2.17) is equivalent to minimize (2.16) with $C = n\alpha$.

SVR can be seen as an extension of more traditional regression techniques: for instance, when $L(\xi) = |\xi|^2$ is used as loss function, then we fall into the case of a minimum square error regression problem.

As for the classification problem, the training problem (2.16) it is solved more easily in its *dual* formulation. The solution of the dual problem yields the function $f(x)$, which can be written as a linear combination of the training data, the Lagrange multipliers λ_i, λ_i^* (associated to the two first constraints of problem (2.16)), and the constant term b, whose computation stems from the KKT conditions:

$$f(x) = \sum_{i=1}^{n}(\lambda_i - \lambda_i^*)\langle x_i, x\rangle + b = \sum_{i=1}^{n_{SV}}(\lambda_i - \lambda_i^*)\langle x_i, x\rangle + b, \tag{2.19}$$

The Lagrange multipliers verify the constraints:

$$\sum_{i=1}^{n}(\lambda_i - \lambda_i^*) = 0 \quad \text{and} \quad 0 \leq \lambda_i, \lambda_i^* \leq C. \tag{2.20}$$

The KKT conditions also imply that if $\lambda_i, \lambda_i^* \neq C$ and $|f(x_i) - y_i| < \varepsilon$, then λ_i, λ_i^* must be zero. Intuitively, as errors lower than ϵ are tolerated, training data lying inside the so called "ε-tube" will not contribute to the problem solution (nor to its cost). As before, not all x_i are needed to calculate $f(x)$, but only the $n_{SV} < n$ training points x_i whose $\lambda_i, \lambda_i^* \neq 0$, already referred to as support vectors. Once again, the problem complexity is independent of the dimension of the input space, and depends only in the number of support vectors.

An interesting characteristic of the SVR is that it is possible to choose the kernel K such that the solution results in a regression based on the Fourier series development or the interpolation by "splines" [49].

2.3.1 Methods of Solution

In this section we briefly present how the optimization problems defined by the training of SVM can be solved. This aspect of the SVM it is not our main interest since there are a lot of very good available implementations and we will use them in the next chapters. In particular we use JMySVM [46] an open source SVM implementation distributed along with the Rapidminer [50] software tool and LibSVM in its version for MATLAB.

Basically, a quadratic convex problem must be solved for which the existence of a unique minimum is guaranteed. In [35], three main steps are identified: (i) set out the KKT conditions, (ii) maximize the dual object function (subject to its constraints) to approach optimality, (iii) define a decomposition algorithm so that only portions of the training set are handled at a given time. Some of the techniques described in [35] are (refer to it for more references or details):

- Conjugate gradient with constraints: is a classical conjugate gradient where the directions are projected over the subspaces defined by constraints $\sum_i \lambda_i y_i = 0$. If the problem is seen as a sequence of equality constrained problems, a Netwon method could solve it in one step or at most in n steps using conjugate gradient descent.

- Projection methods: similar to conjugate gradient, but limiting the moves so that the actual move remains in the feasible region.

- Bunch-Kaufman decomposition: takes advantage of the fact that the Hessian can be diagonalized and that most of the lagrange multipliers λ_i are nulls.

- Interior Point methods: solves the dual and the primal problem simultaneously by only gradually enforcing the KKT conditions to find a feasible solution and using the duality gap [37] to determine the quality of the solution. This method is specially useful if the support vectors proportion is expected to be high.

2.4 The VC Dimension of SVM

In this section we will show that the VC dimension of SVM, contrary to its initial motivation, can be very large, even infinite. However, several arguments are explored to explain that despite the VC dimension is not small, SVM can and usually have very good generalization performance. As mentioned before, it is not always easy or even possible to compute the VC dimension. The following results whose demonstrations can be founded in [35], correspond to the VC dimension of the SVM with polynomial and Gaussian kernel.

Theorem 2.4.1.

1. *Let d be the dimension of the input space, the VC dimension of SVM with polynomial kernel is $C_p^{d+p-1}+1$ (note that it quickly get very large).*
2. *The SVM with Gaussian kernel for which the error penalty can take any value, has infinite VC dimension.*

This results can be seen as contrary with the original ideas introduced in Sec.2.1; however it must be noted that the effective error can be in practice much smaller than the value given by the bound 2.4. In fact, different number of experiments show that SVM obtain very good performances when compared with other learning algorithms, in particular when the Gaussian kernel is considered [51]. For example in [35] an example of a SVM with Gaussian kernel that can classify without errors any number of training points is constructed (and the VC-dimension is infinite!). The good results for the SVM seems to be based on the properties of the optimal separating hyperplane more than the VC-dimension. In this sense the following result suggest that algorithm that minimizes D^2/C^2 may obtain better generalization performance (i.e. more tight bounds for the expected error) [16].

Theorem 2.4.2. If the training points verify that $||x_i|| \leq D \ \forall i = 1, \ldots, n$, and they admit a separating hyperplane optimal with $b = 0$ and margin M, then:

$$E(P(\text{ error })) \leq \frac{E(\frac{D^2}{M^2})}{n}$$

where $P(\text{Error})$ is the probability of error in the test set, the expectation on the left side is over all training sets of size $n-1$ and the one on the right side is over all training sets of size n.

However, this theorem will be useful only if the diameter of the minimal enclosing sphere D can be calculated for any kernel function and for any training set size. In [35] an algorithm to calculate it is presented; which in fact is very similar to the training of a support vector machine.

Chapter 3

Link Load Prediction

In this chapter we explore the use of Support Vector Regression (SVR) for the purpose of link load prediction. As we explained in Chap. 2, SVR works well in many learning situations because it generalize to unseen data, and are amenable to continuous and adaptive online learning – an extremely desirable property in network environments. Motivated by the encouraging results recently gathered by means of SVR on other networking applications, our aim is to enlighten whether SVR is also successful for the prediction of network links load at short time scales. This problem is of great interest in networking for both capacity planning and self-management application (e.g. bandwidth provisioning, admission control, trigger of backpressure mechanisms, etc.).

It is fairly well accepted that, as a result of network services and Internet applications evolution, network traffic is becoming increasingly complex. On the one hand, transport networks are challenged by the current convergence trend of voice/video/data services on an all-IP network, and by the fact that user-mobility will likely translate into service-mobility as well. On the other hand, the explosion of Internet telephony, television and gaming applications implies that we may be forced to re-think what we mean by “data” traffic. Moreover, the widespread usage of application layer overlays directly translates into a much higher variability of the data traffic injected into the network. We wonder whether such variability can be efficiently forecasted, and if so, with what level of accuracy. In contrast with most of the work related to network load forecast, which are based on the analysis of time series properties, we prefer to focus on techniques that avoid to make any assumption on the phenomenon under observation such as SVR.

In Sec. 3.2, we focus first on link load forecast based only on past measurements, following an approach known as “embedded process” [31]. Though the SVM approach fits well to longer time-scales as well, which are more of a concern for capacity planning, we focus here on the estimation of load variation at short time scales. Basically, the link load is treated as a *series*, and the future value is forecasted based on an arbitrary number of its past measurements. Adopting a hands-on approach, we evaluate the effectiveness of SVR for this purpose by exploring a rather extensive parameter and design space. Our aim is twofold: first, we want to evaluate the SVM accuracy and robustness and, second, we want to provide useful insights on the tuning of the SVM parameters, an aspect not always clear in previous work.

We compare the performance with those achievable by parametric and non parametric models such as Moving Average, Auto-Regressive models and the Nadaraya-Watson estimator. Our results show that, despite a good accordance with the actual data, the SVR gain achievable over simple prediction methods is not enough to justify its deployment for link load prediction at short time scales. Yet, we have to tribute SVR of a number of extremely positive aspects: for instance, SVR models are rather robust to parameter variation, and their computational complexity is far from being prohibitive, which makes them suitable for online prediction. We also investigate methods to extend the forecast horizon using forecasted values as input for a new prediction: interestingly, this approach of recursive SVR may significantly extend the achievable forecast horizon, entailing only a very limited accuracy degradation.

Given the results of this first approach, in Sec. 3.3 we take a radically different approach and we devise techniques that significantly improve the accuracy of SVR. Firstly, we consider a wider framework, targeting the maximum or the percentile of the link load over a given temporal horizon (larger than the one considered before) instead of the single value predicted by the embedding procedure. Secondly, we consider as SVR input a *summary* of statistical properties (e.g.,mean, variance, quantiles, peak, etc.) of the link load, as opposite to the past measurement of the link load *series* itself. Furthermore, in order to gather more robust results we extend the evaluation to a more complete set of traffic traces.

In this case, we quantify the impact of several factors such as forecast timescale, samples aggregation strategy, input feature combination, forecast target, type of traffic, etc., in the SVR forecast accuracy. We also show that the use of an “intelligent” combination of different machines, can bring further advantages in both the forecast accuracy as well as in the tuning of SVM, which allows us to construct a very robust model. As mentioned before, we evaluate our model over different real world traffic traces representing very different network scenarios (Ethernet and WiFi LAN, ISP and enterprise). Our results show that SVR may provide accurate predictions and very significant gain over both naive estimation techniques and also over the “embedding process”.

Before presenting in detail both approaches and its results, some related works are discussed in the next section.

3.1 Related Work

Most of the work related to network load forecast is based on the analysis of time series properties. In this context, a number of very different models [52, 53, 54] have been proposed, ranging from very simple to very complex ones. However, the majority of these approaches relies on specific assumptions and underlying models for the network traffic (e.g., they are tailored to capture Long Range Dependence (LRD) [55] at short and long timescales, etc.). A first drawback is that such models will no longer be applicable if the assumption no longer holds (e.g., considering other timescales). A second drawback is that such models usually rely on the precise estimation of some traffic parameters, whose computation can be a very intensive and delicate task (e.g., Hurst parameter of the arrival time series). Rather, as in [1, 56], we prefer to focus on techniques that, avoiding to make any assumption on the phenomenon under observation, allow for intrinsically more robust and flexible prediction. A simple local Gaussian predictor is provided in [1] as a core tool to guide the bandwidth provisioning in the hose model: interestingly, the model is able (but not *forced*) to embed assumptions on the LRD properties of the traffic, by an appropriate tuning of the parameters. TCP throughput prediction is the object of [56], where authors compare formula-based versus history-based prediction schemes, showing that even simple moving-average models are able to yield satisfactory results (provided that one copes with major error sources).

While the use of SVM for classification is relatively more popular in networking research, especially in the context of anomaly and intrusion detection [18, 30], the use of SVM for regression is largely left unexplored. To the best of our knowledge, the only works that explore the use of SVR techniques in the networking field are [19, 34].

TCP throughput prediction on a given path is the object of [19], where the prediction is based on a combination of path properties (such as queueing delays and available bandwidth) and on the performance of prior file transfers as well. Authors show that when the path properties are *precisely* known (e.g., when they are provided by an “oracle”), SVR is able to predict TCP throughput within 10% of the actual value in 90% of the cases – which represent nearly a 3-fold improvement in accuracy over prior history-based methods. Also, in more realistic scenarios and using less accurate measurements of path properties (e.g., gathered by means of active probes), the predictions can be made within 10% of the actual value nearly 50% of the time – which still represents a 60% improvement, with a furthermore much lower impact on end-to-end paths.

The authors of [34] focus instead on the prediction of the latency toward an unknown IP address, based on the latency knowledge toward other previously contacted IP addresses. Using as input features vectors of IP addresses bits (transformed into a 32 dimension input space, where each bit of the address corresponds to a different dimension), authors show that the estimation performance is within the 30% of the true value for approximately three-quarters of the latency prediction on a large Internet data set. More in details, SVM regression on a large randomly collected data set of 30,000 (IP,latency) couples, yield a mean prediction error of 30 ms (25 ms) using only 6% (20%) of the samples for training.

3.2 Embedding Process

This section details how we apply the SVR framework to the load forecast problem. Other techniques, such as Moving-Average (MA) and Auto-Regressive (AR) models, whose performance will be compared with SVR's, will be introduced in Sec. 3.2.1.

We stress that we deliberately avoid comparison with other unsupervised predictors such as the Local Gaussian Predictor [1], as we experimentally verified that in the short timescales considered in this work it systematically overestimates the incoming traffic rate. While in some applications this is actually a desirable feature (e.g., as in VPN bandwidth provisioning, where over-provisioning translate in fewer losses and thus in a greater service QoS), in many other contexts it is not (e.g., when admission control is performed, over-estimating the incoming load unjustifiably increases the flow reject ratio): thus we prefer to avoid introducing any a priori bias.

We also point out that, in principle, an option to accommodate short timescales variability due to Internet traffic burstiness, could be to exploit the *a priori* knowledge of the scaling relations between rate and variance at different timescales as in [1]. However, as previously stated, we prefer to avoid any assumption on the phenomenon under observation.

In the context of time series prediction by means of SVR, there is no a priori restriction on the type and number of input features. A known approach, which we adopt here and explain in the following, is the so called "embedding process" [31].

Let be $\lambda(t)$ the traffic load measured in the time interval $[t-\tau, t]$. By quantizing the time in multiples of τ, we obtain a time series $\{\lambda_k\}_{k\in\mathbb{N}}$, where λ_k is the average traffic load measured in the interval $[(k-1)\tau, k\tau]$. The SVR embedding process then uses an arbitrary number d of past measurement of the above series in order to predict its future value. Thus, when given a d-dimensional input x ($S_X = \mathbb{R}^d$), a trained SVR function returns as output $\hat{y} = f(x)$ a forecast of the target y ($S_Y = \mathbb{R}$), which in our case are (see Fig. 3.1):

$$x = (\lambda_{k-(d-1)}, \dots, \lambda_{k-1}, \lambda_k) \quad \text{and} \quad y = \lambda_{k+1} \tag{3.1}$$

In order to construct all the possible x input tuples, we use a sliding window of length d over the time series to construct all possible input/output pairs, obtaining the set $\{(x_i, y_i)\}_{i=1\dots L-d}$ where L is the time series length.

A subset of this set will be used as training set, i.e., to construct the SVR forecast function $f(x)$ (see (2.19)); then, the model accuracy will be evaluated over the *complement* of the training set, i.e., on unknown data. More precisely, the SVR training set is constructed by randomly selecting a few out of all the possible x input tuples: in other words, SVR training can be thought as realized by means of a jumping window. The impact of the training set selection will be thoroughly examined later.

In general terms, τ can be thought as the observation timescale, the dimension d as the minimum number of state variables required to describe the system and their

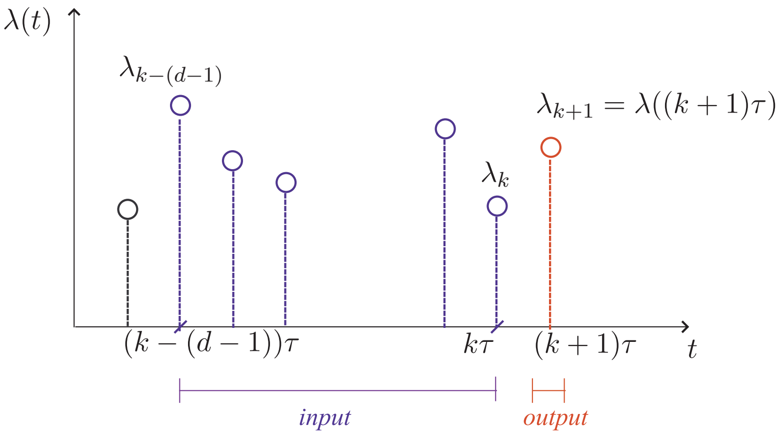


Figure 3.1: Link load time series with SVR inputs/outputs for the embedding process.

product $d\tau$ corresponds to the average system memory length. The traditional "embedding process" assumes that the observed time series is the projection of a deterministic dynamic operating in a high-dimensional state space: in this case, the parameters d and τ can be obtained by running (rather computationally intensive) geometric heuristics on input data. However, we point out that the above assumptions do not apply directly to our context, as the network load dynamic is clearly not well represented by a deterministic process.

Moreover, our aim is rather to build a robust engine for the online estimation of traffic load on *arbitrary* timescales: thus, we prefer to avoid constraints on the selection of the operation point (d, τ) – or at least on the operation timescale τ. Therefore, we prefer to cross-check the impact of the embedding parameters choice *a posteriori*, based on the empirical results of the regression: as a side effect of this choice, we will extend the sensitivity analysis of SVR to a wider range of parameters. In what follows, we assume a Gaussian kernel, to which an infinite dimensional mapping space corresponds (c.f. Sec. 2.2.3). This choice is motivated by the good performance shown in both the time series prediction [57] and more general network [19] contexts:

$$K(x, x') = e^{-\gamma\|x-x'\|^2}. \tag{3.2}$$

3.2.1 Reference Techniques

In this section we describe those methods whose performance will be compared with the one obtained by SVR.

Moving-Average Models

Given a time series $\{\lambda_k\}_{k\in\mathbb{N}}$, the one-step d-order Moving Average (d-MA) predictor can be defined as:

$$\hat{\lambda}_{k+1} = \frac{1}{d} \sum_{i=k-d+1}^{k} \lambda_i \tag{3.3}$$

As a general remark, if d is too small the predictor cannot smooth out the noise in the underlying measurements, whereas a too large value of d makes it slow to adapt to non-stationarity properties of the data. The predictor (3.3) is the simplest among the unsupervised forecast methods; yet, in a slightly different context, the authors of [56] showed that, despite its simplicity, d-MA is able to provide accurate results provided that it copes with the two major error sources: namely, Level-Shift and Outliers (LSO). We implement the LSO heuristics as in [56], and denote with d-LSO the corresponding predictor. Basically, outliers are just ignored, whereas the detection of level-shift triggers a filter restart. More on details, considering a set of measurements $\{\lambda_1, \ldots, \lambda_d\}$, a sample λ_k is said to be an *outlier* whenever it differs from the median of the set by more than a relative difference ψ. Moreover, an increasing (decreasing) *level-shift* is detected in correspondence of λ_k whenever the following conditions jointly holds:

- the measurements $\{\lambda_1, \ldots, \lambda_{k-1}\}$ are all higher (lower) than $\{\lambda_k, \ldots, \lambda_d\}$
- the median of the first portion $\{\lambda_1, \ldots, \lambda_{k-1}\}$ is higher (lower) than the median of the second portion of the set by more than a relative difference χ
- $k + 2 \leq n$, to avoid classifying an outlier as a level shift

When a level shift is detected, all measures prior to λ_k are ignored and the predictor is restarted from λ_k. In the following, we select $(\chi, \psi) = (0.3, 0.4)$ as in [56], which correspond to a good parameter choice in our data set as well.

Auto-Regressive Models

We will compare SVR performance also with an Auto-Regressive (d-AR) models. d-AR models are similar to d-MA models: the main difference is that they take into account not only previous observations, but previous *predictions* as well.

$$\lambda_k = \sum_{i=1}^{d} \varphi_i \lambda_{k-i} + \epsilon_k \tag{3.4}$$

d-AR predictors takes the general form of (3.4), where φ_i are parameters of the model and ϵ_k is a noise factor. In the embedding process context, a linear prediction function leads to a *class* of autoregressive models, and fitting procedures can be used to extract the most appropriate regression function within the class. Indeed, it has been shown [57] that the use of a *linear kernel* yields to an d-AR model equivalent to the one estimated through other means such as, for instance, the Yule-Walker equations [52]. At the same time, the advantage of using SVR to provide d-AR models lie in its simpler fitting procedure and robustness in the presence of outliers: thus, in the following we will use a SVR with linear kernel (c.f. Sec. 2.2.3) to evaluate the performance of autoregressive models.

Nadaraya-Watson Estimator

The last class of forecast model that we will compare SVR performance with is the Nadaraya-Watson estimator [23] [58]. Given a random vector $X \in \mathbb{R}^d$ (as for the SVR inputs) and a random variable $Y \in \mathbb{R}$ (as for the SVR output), the regression problem consists of finding an estimation of the function Φ such that:

$$Y = \Phi(X) + \epsilon,$$

where ϵ is an independent and centered random variable representing errors (the noise factor of the AR models).

The function Φ can be estimated from observations of the pairs (X, Y). Several models can be assumed for Φ such us linear or parametric ones. However, since we are already considering this kind of model (d-MA and d-AR) , we assume a non parametric one, i.e. the function cannot be determined with a finite number of parameters. In particular, the Nadaraya-Watson(NW) estimator is defined by the function $\Phi_n : \mathbb{R}^d \to \mathbb{R}$ such that:

$$\Phi_n(x) = \frac{\sum\limits_{i=1}^{n} Y_i K(\frac{||x-X_i||}{h_n})}{\sum\limits_{i=1}^{n} K(\frac{||x-X_i||}{h_n})}, \tag{3.5}$$

with the convention that $\Phi_n(x) = 0$ if $\sum_{i=1}^{n} K(\frac{x-X_i}{h_n}) = 0$. The function K is a kernel, in the sense that $K \geq 0$ and $\int_{\mathbb{R}} K(x)dx = 1$ and h_n is a sequence of positive numbers that goes to zero with n. The selection of the window h is of crucial importance since a non optimal selection may strongly degrade the accuracy of the estimation. For instance, a large value of h will imply a large averaged estimation whereas a small one, will imply a very noisy one. It must be noted the definition of kernel introduced here is different to the one defined in Sec.2.2.3 which relates with the existence of an inner product.

An intuitive interpretation of (3.5) is that $\Phi(x)$ is obtained by averaging the outputs values Y_i corresponding to inputs X_i that are “near” x, where the sense of near is given by the kernel function. More formally, assuming that the vector (X, Y) has a density in $\mathbb{R}^{d+1}$, this estimator can be seen as the plug-in estimator obtained by substituting this density by its kernel estimation. More details can be found in [59], in particular about the hypothesis for the different types of convergence of Φ_n to Φ.

We will consider two different kernel functions: Gaussian (3.6) and Epanechnikov (3.7), which are defined as:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{3.6}$$

$$K(x) = \frac{3}{4}(1 - x^2)\mathbf{1}_{\{|x|<1\}} \tag{3.7}$$

An important difference between SVR and NW estimator is that for the former, a model is constructed from the training phase, whereas for the latter each time a new value is to be predicted, the rest of the time series could be used as observations (or at least the past observations for online predictions). However, to achieve a fair comparison, we will use the same training and validation sets for both models. For SVR, once the training is done and the model constructed the prediction is very easy and results of a simply function evaluation. Moreover as we have seen before, it depends on the number of support vectors and not on the whole training size. Instead, for each NW prediction the whole training set is considered in the calculation of (3.5). This characteristic is a serious drawback of NW when used for online prediction, as in our case, since it results to be more computationally expensive.

3.2.2 Sensitivity Analysis of SVR Performance

Support vector embedded process regression is affected by many parameters, pertaining to two different areas. A first set, related to the SVR itself, includes the training size S and the smoothing factor C of (2.16), the tolerance ϵ of the loss function (2.18), and the parameter γ of the kernel function (3.2). The second set is instead related to the embedding process parameters, i.e., the timescale τ and dimension d (3.1).

In the following, we provide a very thorough and careful tuning of SVR, with the twofold intent of i) evaluating the extent of SVR accuracy for link load prediction, as well as ii) assessing the sensitivity of SVR forecast performance to the above parameter variation. Results reported in this chapter are gathered through JMySVM [46] an open source SVR implementation distributed along with the Rapidminer [50] software tool.

Input Data

Prior to inspect the impact of the above parameters on SVR performance, we need to provide details on the input data, that were collected at the POP of a major Italian ISP. This dataset is very interesting since it refers to an innovative ISP which is providing end users (residential, SoHo or large companies) with data, voice and video over IP by means of either an ADSL or a FTTH link (no PSTN link is offered). Traffic is therefore composed of data transfers over TCP, VoIP and VideoIP traffic over RTP/UDP. Moreover, as users make extensive use of P2P applications, VPN services, etc., the resulting traffic mix is therefore very heterogeneous.

	i		0	2	4	6	8	10
	τ_i	[ms]	1	4	16	64	256	1024
Day-time	μ	[Mbps]	116.4	118.3	120.4	119.5	118.9	121.3
	σ	[Mbps]	28.9	20.7	16.4	12.7	10.3	7.9
Night-time	μ	[Mbps]	60.8	62.4	68.1	69.6	68.2	66.2
	σ	[Mbps]	32.6	21.2	15.4	9.6	6.8	8.7

Table 3.1: Input trace: link load at different timescales

We sniffed a one-day long trace on Monday the 15th May 2007, and consider a single traffic direction, namely the downlink one. We then extracted several 10000 second long (about 2h45) subsets of the trace: here, we report results referring to two different subset, namely a daily-busy period and nightly-idle one. In the daily subset (D), average link load is 121.3 Mbps, whereas in the nightly subset (N), average load was 66.2 Mbps. For each subset, we consider different timescales $\tau_i = 2^i$ ms with $i \in [0, 10]$, and for the sake of brevity, in the description we approximate $\tau_{10} = 1024$ ms with $\tau = 1$ s.

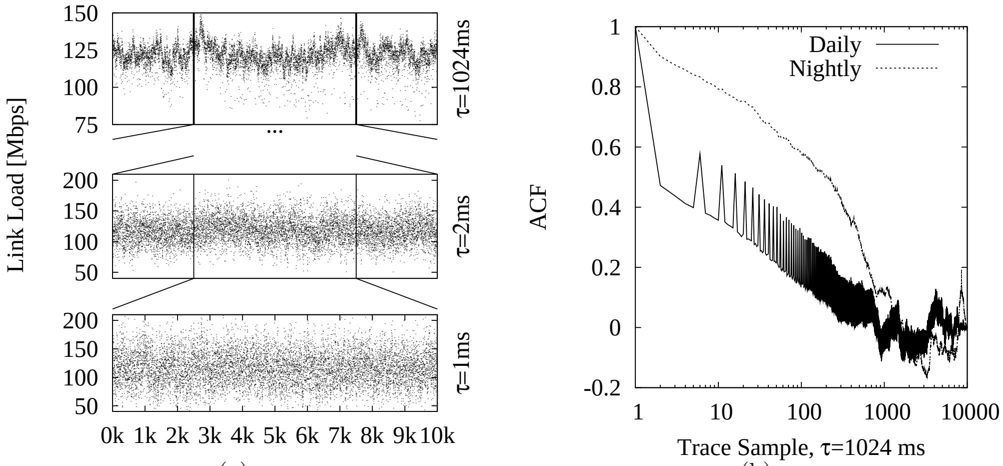
Further details relative to the mean and standard deviation of the load at different timescales are reported in Tab. 3.1 for even values of i: it can be seen that in general, the lower the timescale the higher the load variance. Also, considering the coefficient of variation CoV $= \sigma/\mu$, we notice that load variation is more important during the night, where CoV is roughly the double with respect to the daily subset. For all different timescales, we construct a $N = 10000$ long dataset with the partitioning criterion illustrated in 3.2(a): the subset for the (i-1)-th timescale corresponds to the central portion of the i-th one.

The daily and nightly datasets present another interesting difference. 3.2(b) depicts the autocorrelation function of the load at timescale $\tau = 1$ s: the peak of the busy trace exhibits a periodic fluctuation on the range of 5 s (and multiples of 5 s) which is absent in the nightly trace. Clearly, this dependence will affect any 5-lag samples when $\tau = 1s$ (i.e., λ_k and λ_{k-5}) and more generally any two samples that are $5n$ seconds apart.

SVR Parameters

To tune the SVR performance, we start by performing a *grid optimization* process, which boils down to the selection of a tuple $(C^*, \epsilon^*, \gamma^*)$ of SVR parameters. The best tuple is chosen as the one that minimizes the Root Mean Square Error (RMSE) of the prediction, a metric which asses the quality of the estimator in terms of its variation and unbiasedness. RMSE has the same units as the quantity being estimated (specifically, Mbps in our case), and is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2} \tag{3.8}$$



(a) (b)

Figure 3.2: (a) Subset choice at different timescales for the daily dataset and (b) Autocorrelation function of the Daily and Nightly trace datasets

We point out that while RMSE is more suited than the Relative Error (RE) to assess the quality of estimation, at the same time its interpretation is somewhat harder than that of the RE index. As such, in the following we will mainly use RMSE to drive the parameter tuning and selection, but will use both RMSE and RE metrics to quantify the forecast accuracy.

Fixing for the moment $d = 5$ and $\tau = 1\,\mathrm{s}$, we construct a dataset with all possible inputs/outputs by means of a sliding window as described in Sec. 3.2. We selected 20% of the dataset at random to train the SVM, and tested the prediction accuracy over the remaining 80%.

We choose 10 values for each of the SVR parameters, for a total of 1000 tuples (C, ϵ, γ). To select the boundary of the parameter space to be explored, we apply the following reasoning. A prescription for the regularization parameter C follows from (2.19): if we consider that, $|\lambda_i - \lambda_i^*| \leq C$ and $|K(x_i, x_j)| \leq 1$, we have that $|f(x)| \leq Cn_{SV}$, which yield to $C \leq |f(x)|/n_{SV}$. While for $|f(x)|$ a reasonable choice is $\max(|y{+}3\sigma_y|, |y{-}3\sigma_y|)$ (to avoid outliers influence), since the number of support vector cannot be known a priori we consider the boundary cases where either all training data are support vectors $S_V = S$, or only a single data point is a support vector $S_V = 1$. Finally, [60] suggest that ϵ should be proportional to the input noise level: however, as the definition of link load "noise" is questionable, we are forced to resort to an empirical choice – and we proceed similarly for γ.

This grid optimization process for the daily trace yields to a minimum RMSE=5.9 when $(C^*, \epsilon^*, \gamma^*) = (30, 5, 0.05)$, to which a relative error RE=3.5% corresponds. Fig. 3.3 shows the whole (C, ϵ, γ) parameter set explored, conditioning over each of the three parameters. For the sake of clarity, let us consider the leftmost plot of Fig. 3.3, whose x-axis represents the C parameter values. Each point in the plot represents a single

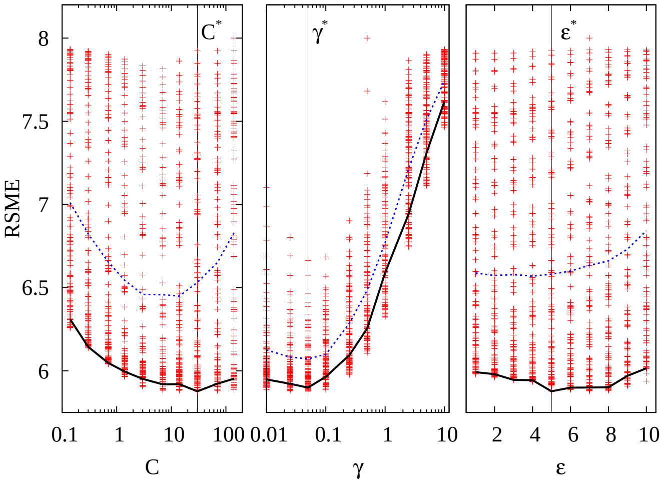


Figure 3.3: Grid optimization process for the selection of $(C^*, \epsilon^*, \gamma^*)$

experiment, and for any value of the C parameter on the x-axis, 100 points are plotted that correspond to the 100 combinations of the other two parameters ϵ and γ.

The plot also reports some reference lines: the vertical thin line refers to the best value C^*; the dotted thick line represents, for any given C, the average of the RMSE achieved for the 100 possible combination of ϵ and γ and the solid thick line refers instead to the RMSE achieved as a function of C when the other parameters are set to their best values (i.e., ϵ^* and γ^*).

From the Fig. 3.3, it can be gathered respectively that i) as the RMSE is convex in C, the value of C should be neither too big nor too small, ii) the prediction error exhibit a (roughly exponential) increase with γ value, and iii) that the impact of ϵ is less significant with respect to C and γ.

A similar operation on the nightly trace yielded to a different best combination of parameters, namely $(25, 0.1, 0.001)$. At the same time, the daily parameters $(C^*, \epsilon^*, \gamma^*)$ ranked 50th in the night trace, with an RMSE increase of 4%: thus, even in extremely different load conditions, the SVM prediction is rather robust to the parameter choice.

Training Size Impact

By fixing the parameters $(C^*, \epsilon^*, \gamma^*)$, we explore now the impact of the training set size S on SVR performance. We build a training set with S randomly chosen samples and evaluate the prediction accuracy over the remaining samples. The process is repeated 10 times for each value of S, changing the training and validation set every time. RMSE results are reported in the boxplot Fig. 3.4 as a function of the training size S (top x-axis) and ratio S/N over the total trace size (bottom x-axis). The boxes report the lower quartile, median, and upper quartile values; the lines extending from each end of

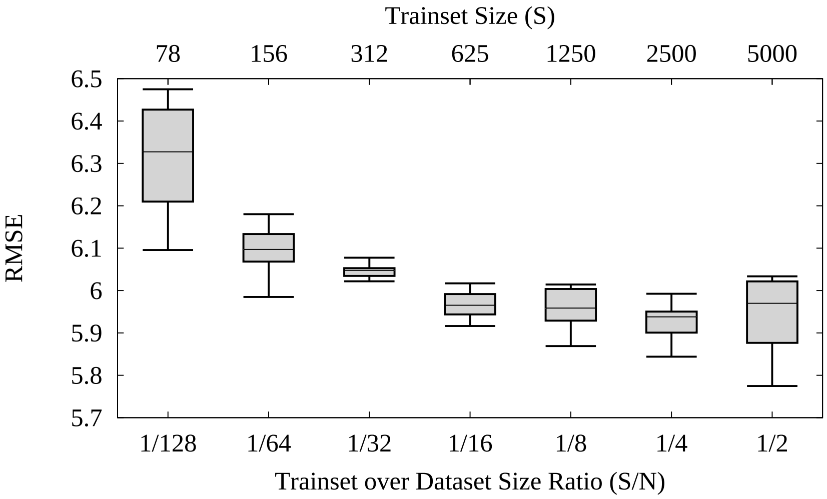


Figure 3.4: Impact of training size on prediction accuracy

the boxes represents the extent of the rest of the data (i.e., the maximum and minimum values), and outliers are not filtered out.

First, it can be observed that there exists a noticeable variation in the RMSE performance for a given training set size, emphasizing the importance of the training set in the model construction. Then, it can be observed that whenever the number of samples is exiguous ($S/N < 1/16$), the SVM is *under-trained* and prediction error is large. For instance, relative error, not shown on the picture, grows beyond 10% when $S/N < 1/128$.

Afterwards, SVM rapidly learns and both the RMSE and RE errors quickly drop (specifically, RMSE=5.9 and RE=3.5% when $S/N = 1/4$), until the SVM is *over-trained* ($S/N \geq 1/2$) and the error slightly increases (RE=4.2%). In our situation, in the zone $S/N \in [1/8, 1/4]$ the RMSE stays at acceptable values, with a minimum in $S/N = 1/4$, which validates our original choice of $S/N = 0.2$.

Embedded Parameters Impact

In this section, we explore the impact of the embedding parameters d and τ in the prediction accuracy of SVM versus d-MA, AR model and NW (c.f. Sec. 3.2.1). As earlier explained in Sec. 3.2.1, AR models are evaluated through a SVR with a linear kernel. In this case, the parameters that must be set are only C and ϵ: by performing a new grid optimization process, we obtain that the best choice is keeping the values C and ϵ already obtained for the Gaussian kernel. For the NW estimator, a grid optimization was made to find the optimal value of the window h. It results to be much less robust than SVR, since it strongly depends on the value of d, specially when the Epanechnikov kernel is considered. Remember that for SVR the parameters were optimized for $d = 5$.

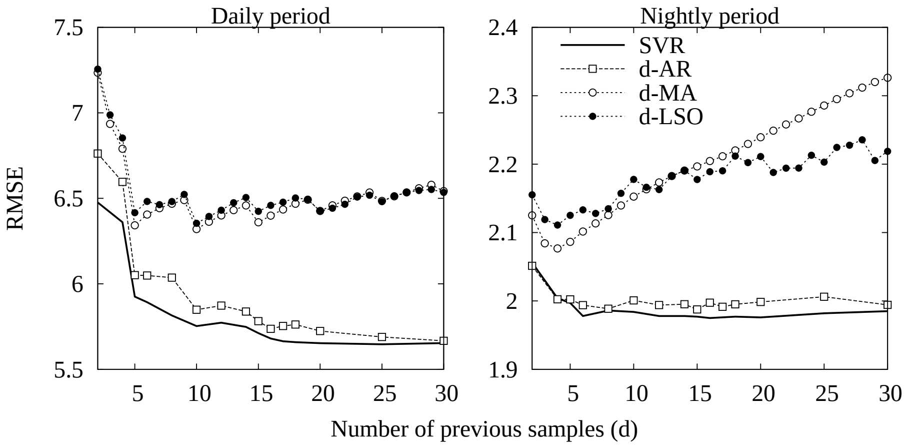


Figure 3.5: Impact of the number d of previous samples on the forecast accuracy

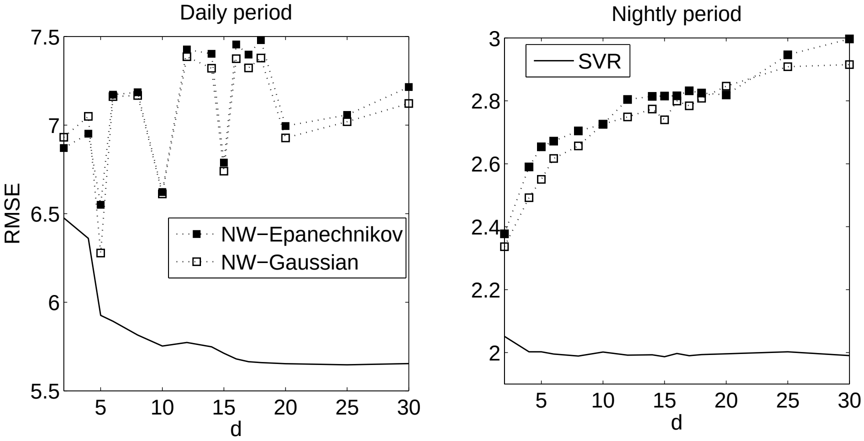


Figure 3.6: Impact of the number d of previous samples on the forecast accuracy for the Nadaraya-Watson predictor using the same training and validation set as for SVM.

Fig. 3.5 and Fig. 3.6 reports the RMSE results as a function of the number d of previous samples, for both nightly (right) and daily (left) periods at the $\tau = 1\,\mathrm{s}$ timescale. The results for the NW estimator are shown separately to ease the comparison. Each point in the plot correspond to the average result over 10 repetitions of the experiment. As a first remark, results are different, being SVR the one that always obtain the best results. However they are quantitatively very close: in other words, SVR does not appear to offer a significant improvement, especially over d-AR models.

Nevertheless, let us investigate more closely the daily period, where the d-MA and the NW estimator (for both analyzed kernels) resent much of the periodical fluctuation: intuitively, the error is minimum whenever the forecast is exactly a multiple of the periodical lag (indeed, for $d = 5$ samples, two of them thus 2/5 are correlated, while for $d = 6$ only 2/6 are correlated, and so on) and grows in between two multiples. Interestingly, the LSO heuristic is not helpful in this case (as in this case the periodic

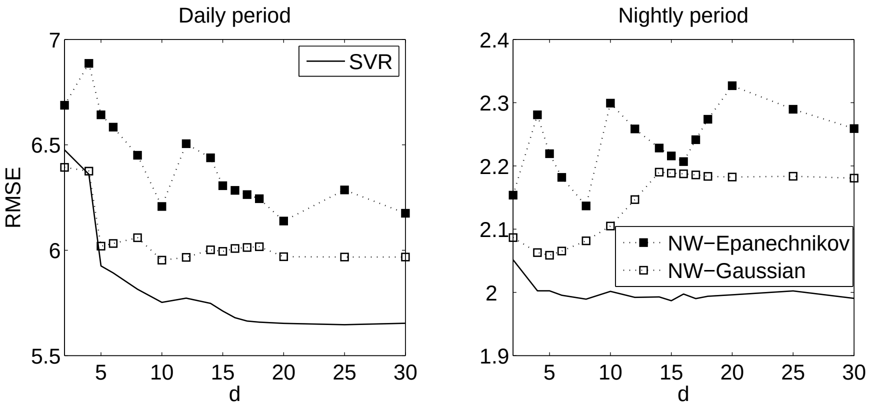


Figure 3.7: Impact of the number d of previous samples on the forecast accuracy for the Nadaraya-Watson predictor training with a fixed range (10% of the time series length) of past values.

fluctuation may be seen as a level shift and thus disregarded), while the robustness of SVR models is preserved. It has to be noted that both SVR and d-AR models are similarly affected by variations of d, but that the use of a Gaussian kernel yields to slightly more accurate results.

Considering the nightly period, it can be seen that the knowledge of a few elements is useful for the d-MA prediction, as long as the number of previous observation is small: indeed, when $d > 5$ the d-MA filter is averaging not useful information, actually worsening its accuracy. At the same time, while the d-LSO limits the error for high values of d, it may actually *worsen* the accuracy at low values of d (and is furthermore sensitive to its parameter tuning). A similar behaviour is observed for the NW estimator. Conversely, SVR outperforms d-MA for all values of d, and is also naturally robust even to unreasonable choices of d – i.e., increasing the number of features neither ameliorate nor degrade the result accuracy. When considering d-AR models, we find out that the performance is very close to that SVR: this can be expected as in this case, the very small value of the $\gamma = 0.001$ means that linear kernel is a good approximation of the Gaussian one. Still, the slightly better performance achieved by the Gaussian kernel confirms it to be a good choice, other than for throughput [19] and latency [34] prediction, even for the purpose of link load forecast.

From Fig. 3.5 and Fig. 3.6, it results that the NW estimator gives results that are qualitatively similar to those obtained by d-MA, suggesting that perhaps the predictor potential of NW is not being fully exploited. We propose then to change the training set used for NW (in the previous we used the same as for SVR) to include only the history (past data) of the time series. More precisely, for each point of the time we use a fixed range of past values to construct the training set; in what follows we use 10% of the total length of the time series times d, so as to obtain the same training set size for all values of d. We decide to use a fixed range instead of all the past values since we observed that the accuracy does not degrade and the time required to obtain the

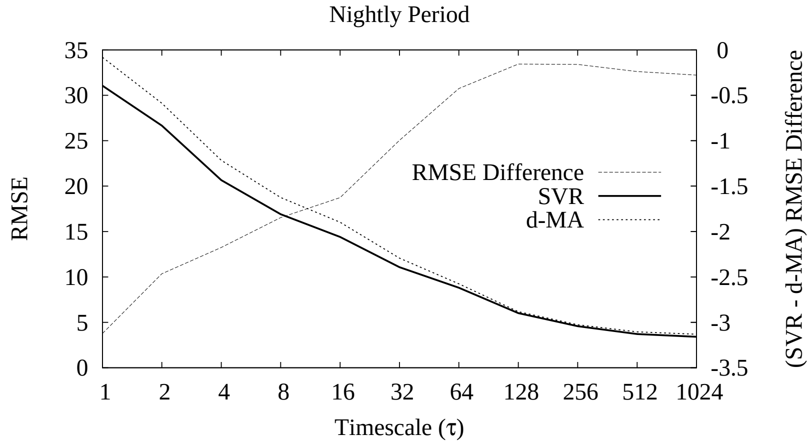


Figure 3.8: Impact of the timescale τ on the forecast accuracy

predictions was significantly smaller. Results for this case are reported in Fig. 3.7 where the value of the window h was again optimized for each value of d (cf. Sec. 3.2.1). It can be seen that results have improved but they are still worse than for SVR. Moreover, the results depend now on the considered kernel, being again the Gaussian kernel the one that obtain better results.

It should be noted that even if the differences with SVR are not significant, the NW estimator is intrinsically expensive for online predictions as explained earlier in Sec. 3.2.1. For instance, the point of Fig. 3.6 corresponding to $d = 10$ for the NW estimator with Gaussian kernel takes 60 times more time than for SVR. It should be said that we used our own MATLAB implementation for NW, but even if it can be optimized the difference with SVR would be still high. The multiplication factor for the kernel Epanechnikov is approximately 20. Furthermore, the variance on the errors for the NW estimator is about twice the one obtained for SVR. We will not dig further in the performance of NW but a more detailed analysis of the computational complexity of SVR is presented in Sec. 3.2.3.

Finally, setting $d = 5$ (thus, the best case for the d-MA filter but *not* for the SVM), we investigate whether SVM forecast bring any improvement at short time-scales. RMSE of the prediction is depicted in Fig. 3.8 as a function of τ, where we report only the nightly period to avoid cluttering the pictures. Behavior of both predictors is similar, with short time scales constituting a stiffer scenario, as it can be expected in reason of the much higher traffic variability shown earlier in Tab. 3.1. The picture also reports the RMSE difference of the two forecast techniques, from which it can be gathered that at very short time-scales (1 ms), SVM brings about a 10% improvement over d-MA.

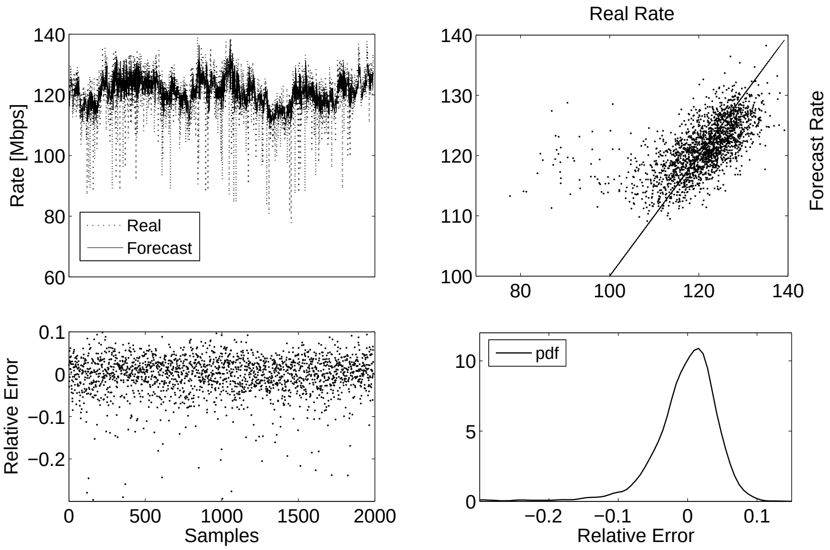


Figure 3.9: Time evolution of SVR prediction

3.2.3 A Closer Look on Forecast Performance

Temporal Evolution

After having performed a sensitivity analysis of SVR, let dig deeper into SVR performance, starting by an investigation of the temporal evolution of SVR prediction. Let us start by an example where, using the SVR parameters obtained through the grid optimization, we fix $\tau = 1$ and $d = 10$ and devote the first 2000 sample of the daily dataset for training and the last 2000 consecutive samples for testing.

The left portion of Fig. 3.9 reports the temporal evolution of real data and SVR forecast (top) as well as the relative error RE (bottom), from which it can be gathered that SVR prediction closely follows the real link load evolution, though the latter exhibits several load "drops", whose duration is on the order of a seconds, which SVR is unable to predict. We highlight, however, that from a provider perspective it may not be critical to be able to precisely predict such short load drops, since no specific action needs to be undertaken as a reaction. This is clearly in stark opposition with respect to load spikes, which we argue would be more critical to be able to anticipate, in order to promptly trigger possible counter-measures.

Impact of load drops on SVR performance is further highlighted in the right portion of Fig. 3.9, which depicts a scatter plot of the real and forecasted load values (top) and the probability distribution function of the SVR relative error (bottom): from comparison of these pictures is clear that the bias toward over-estimation is entailed by the early noticed load drops, rather than by a systematic bias introduced by the model.

This asymmetry due to the load drops, slightly bias the mean relative error toward negative values $\mu = -0.0065$. At the same time, we point out that performing a randomness test for the sequence of relative errors measured over the test set, we find that errors are independent and identically distributed. In particular, results of a Runs Test state that we could not reject the hypothesis of randomness for this sequence (with a p-value of 0.3161): this allows as to calculate a 95% confidence interval for the error $I = [-0.0092, -0.0039]$.

This fact has very important consequences in the case of online predictions, as it could be used in two rather different contexts. For instance, when the forecast accuracy starts to degrade (as many predictions slightly fall out the confidence interval), this may imply that a new training phase should be triggered. Conversely, if errors differ drastically from the previous ones, this might implies that something unusual is occurring with the traffic load – which may find useful applications in the field of anomaly detection.

Computational Complexity

The SVR forecast performance can also be described in terms of their *cost*. First of all, it is important to observe that SVR involves two rather different tasks, namely model *training* and the actual *forecast* operation: the former translates into the solution of an optimization problem, whereas the latter only involves a limited number of simple operations. The above decoupling of SVR computational complexity is a very desirable property. Indeed, training and forecast tasks have to be performed at intrinsically different timescales: model training is an offline operation, that has to be done episodically (or at most periodically) and in the background, while forecast need to be performed constantly and in an online fashion.

Concerning the model training, we need to stress that there exist very efficient algorithms for the solution of the SVR problem: for example, JMySVM implements a Sequential Minimal Optimization decomposition technique, whose computationally complexity is *linear* in the number of support vectors.

The computation of the forecast itself only involves a number of simple operations as can easily be gathered by observing the SVR regression function (2.19). Moreover, an interesting point is that it is actually possible to *upper bound* a priori the number of support vectors returned as a solution to the problem. This can be done by adopting a slightly different formulation of the SVR problem, called ν–SVR [48]: as opposite to the ε–SVR case, though, we would no longer be able to tune the ε-tube, i.e., the margin ϵ below which errors are tolerated. Thus, rather than tuning the tolerable error, one could choose to fix the maximum cost that can be afforded – which could be extremely useful to limit the amount of CPU resources devoted to the online forecast operation.

Still, in the case of ε–SVR, the online forecast cost can be evaluated a posteriori, and as we will show, it does not clearly constitute a performance bottleneck. The primary complexity indicator for the SVR forecast operation is clearly the number of support

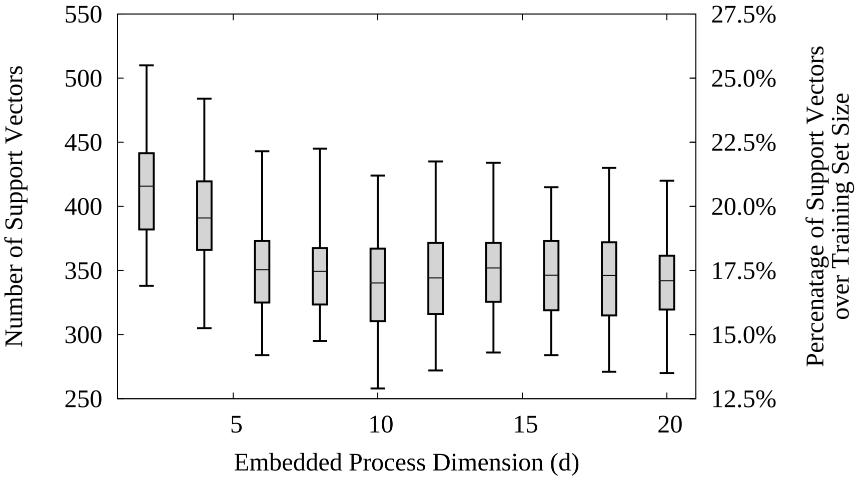


Figure 3.10: Number of support vectors for different values of d

vectors. Fig. 3.10 reports, for the daily dataset and Gaussian kernel, the number of support vectors generated by the model as a function of the embedded parameter d. Box-plots report the median, first and third quartiles, minimum and maximum number of support vectors obtained over 100 different training per value of d; on the left y-axis, we also report the percentage of support vector over the training set size (which is again set to 20% of the dataset).

For small values of d, more support vectors are needed while for values greater than 5, the mean percentage remains almost constant around 17%, though we can still observe a variability of the results, which is due to the random selection of the training set. For the AR models (evaluated via the use of SVR with linear kernel) results are similar, with an increase of about 5% on the number of support vectors – which would thus imply a slightly higher computational complexity and resource requirement in both the offline training and the online forecast operations.

Finally, we give a very rough but nevertheless useful feeling of the actual CPU requirement, reporting the typical execution time performance of our experimental campaign. Experiments, which were run on a Linux PC featuring a 2.0 GHz Intel Core 2 Duo processor equipped with 2 GB of RAM, show that:

- the offline training time is about 0.83 seconds per thousand of support vector – which means that the mean training time of a single SVR model of Fig. 3.10, averaged over all d values, is less than a third of second;

- the online forecast rate is above 9000 forecast per second when $d = 10$ – thus much faster than real-time, which further testify the viability of online SVR forecast.

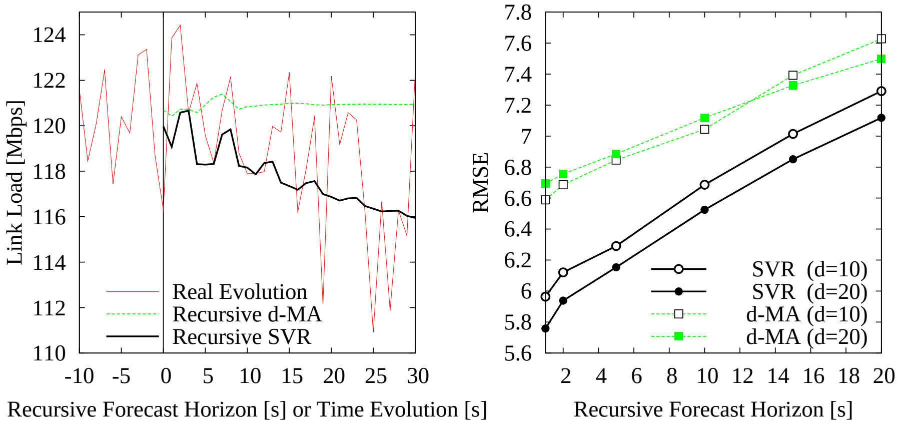


Figure 3.11: Example of temporal evolution of recursive forecast (left) and RMSE error for different values of the forecast horizon (right)

Extending the Forecast Horizon

Finally, an interesting question that we want to assess is whether it is possible to extend the forecasting horizon by *cascading* a series of SVR predictors – in other words whether recursively using forecasted results as an input for a new forecast is viable and promising in the case of link load forecast. We stress that the approach is feasible and intriguing, as results on computational complexity show that, when $\tau = 1\,\mathrm{s}$, it would be possible to actually cascade more than 9000 SVR predictors in a real-time online forecast. Thus, we want to assess how fast the prediction accuracy degrades.

More formally, at time $t = d\tau$ we forecast the load at the next time step $t = (d+1)\tau$, feeding the SVR function $f(\cdot)$ with the last d observation of the series, getting as result the prediction $\hat{y}_{d+1} = f(\hat{x}_d)$. Always at time $t = d\tau$, we then forecast the load at time step $t = (d+2)\tau$ using the last $d-1$ observation of the series and the *forecasted* value $\hat{y}_{d+1}$ in the input $\hat{x}_{d+1}$, thus:

$$\begin{aligned} x_d = \hat{x}_d = (\lambda_1, \ldots, \lambda_d), &\qquad \hat{y}_{d+1} = f(\hat{x}_d) \\ \hat{x}_{d+1} = (\lambda_2, \ldots, \lambda_d, \hat{y}_{d+1}), &\qquad \hat{y}_{d+2} = f(\hat{x}_{d+1}) \\ \hat{x}_{d+2} = (\lambda_3, \ldots, \lambda_d, \hat{y}_{d+1}, \hat{y}_{d+2}), &\qquad \hat{y}_{d+3} = f(\hat{x}_{d+2}) \end{aligned}$$

This procedure can be iterated arbitrarily, and we define as *forecast horizon* H the number of SVR forecast that are cascaded: notice that when $H \geq d$, this means that we are using only forecasted values as input features. In what follows, to avoid the influence of the training set, we train and validate the model over the same dataset: by doing so, we will be able to isolate the impact of the forecast horizon.

For illustration purpose, the left plot of Fig. 3.11 gives an example of recursive forecast for SVR and d-MA models when $d = 10$ and for an horizon up to $H = 30\,\mathrm{s}$.

H	1	2	5	10	15	20
d=10	0%	0.7%	1.1%	1.9%	2.5%	2.8%
d=20	0%	0.8%	1.3%	1.9%	2.4%	2.8%

Table 3.2: Relative Error RE between H-horizon and “one-step” SVR forecast

At time $t = 0$, we apply the cascading process and plot the $\hat{y}_h$ prediction for a given horizon h: notice that the x-axis represent the forecast horizon for both SVR and d-MA, whereas it represents the time (in the future) as far as the real series is concerned. In the picture, the $d = 10$ real load samples immediately preceding time $t = 0$ are also shown: these points determines the initial load forecast for time $t = 1$, after which the recursive process starts (and so SVR and d-MA models start to be fed with their own forecasts).

From this simple example, it can be seen that SVR is able to more closely follow the real data with respect to d-MA, at least for small values of H. Conversely, d-MA predictor quickly converges to a fixed point, though the precise value to which an ∞-cascaded d-MA filter converges depend on the actual *order* of the series[1]. Aiming at assessing the average RMSE and RE of the recursive forecast, and in order to gather results that are valid to a more general extent, we perform an exhaustive set of experiments, repeating the process 10 times for different training sets. The right portion of Fig. 3.11 depicts the RMSE error as a function of the forecast horizon for both SVR and d-MA and two values of $d \in \{10, 20\}$. RMSE is evaluated as the error between the cascaded value $\hat{y}_{d+H} = f(\hat{x}_{d+H-1})$ forecasted at time $t = d\tau$ and the real value λ_{d+H} at time $t = d\tau + H$. Results show that in all cases the RMSE error linearly increases with the forecast horizon H, but that cascaded SVR performance exhibit a lower error. Moreover, it can be gathered that $d = 20$ yields better results: this is not surprising, since in this case we are using more “real values” as input features.

Finally, we compare the H-horizon cascaded SVR forecast (i.e., the recursive SVR forecast for the series value at time $t + H$, performed at time t), with the “one-step” forecast (i.e., a normal SVR forecast). This comparison helps us in assessing the extent of the degradation in the prediction quality, which can be solely imputed to the cascading process itself. These results, expressed in terms of RE, are reported in Tab. 3.2 for different values of H: rather surprisingly, results state that the prediction of a cascaded SVR with a rather large horizon of $H = 20\,\text{s}$ is only about 3% worse than a “one-step” prediction, and that furthermore this holds irrespectively of d. Interestingly, this suggests that, if good accuracy is obtained for $H = 1$, the recursive process would not affect much the quality of the prediction.

[1] For example, it is easy to verify that the d-MA cascading process over $x = (1, 2, 3)$ converges to 2+1/3, whereas it converges to $2 - 1/3$ in case of $x = (3, 2, 1)$.

3.3 Prediction of a Function of the Link Load

In this section we address the problem of forecasting a function of the link load, such as the maximum or the percentile of its distribution, during an arbitrary time interval. The motivation to analyze this new strategy is twofold, in one hand we search to improve the performance obtained by the "embedding process" and on the other hand we believe that it is more useful to know for instance the maximum load over the next time period than the exact value for the next second.

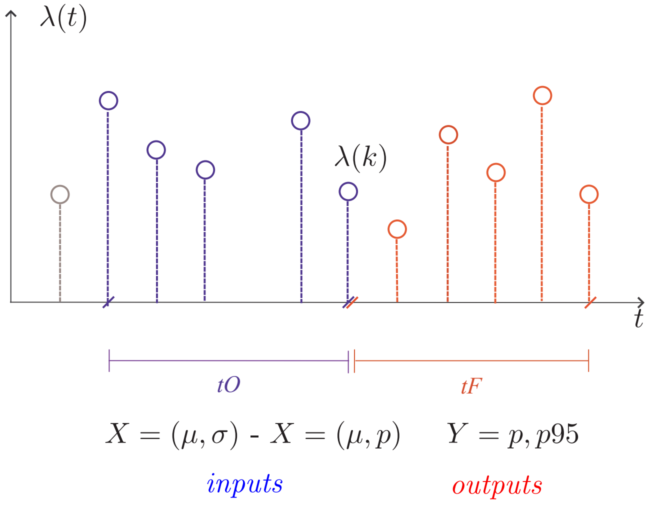


Figure 3.12: Example of possible SVR inputs/outputs.

As before, for a time scale t_S, for each dataset let $\{\lambda_k\}_{k=1,\dots,n}$ be the time series such that λ_k is the average traffic load measured in the interval $[(k-1)t_S, kt_S]$. Let us denote the temporal horizon of the forecast by t_F, and let us further assume that predictions for the time frame t_F are based on observations of a time frame t_O of equal length, i.e., $t_O = t_F$ (see Fig. 3.12). If we compare with the embedding process of Fig.3.1 we can see that the output is not anymore a single value but a function of the observations in a given interval. Also the inputs change, even if the considered values are the same the input is now a summary of statistical properties, for instance the combination of the mean and the standard deviation or the maximum of these values, as indicated in the figure.

More precisely, assume (for the moment) that we want to predict the peak network load on a given temporal horizon, using past observations of the load mean μ and standard deviation σ as input features. Let us now introduce, with the help of Fig. 3.13, the notation that will be used throughout this work. As sketched in the top of the figure, a single pair of features (μ, σ) can be gathered from the whole time frame t_O, which can then be used as SVR inputs to predict the maximum load in the subsequent time frame t_F. Alternatively, the samples constituting the time frame t_O can be *aggregated* into several windows of duration t_W, where a separate set of input features (μ_i, σ_i) can then

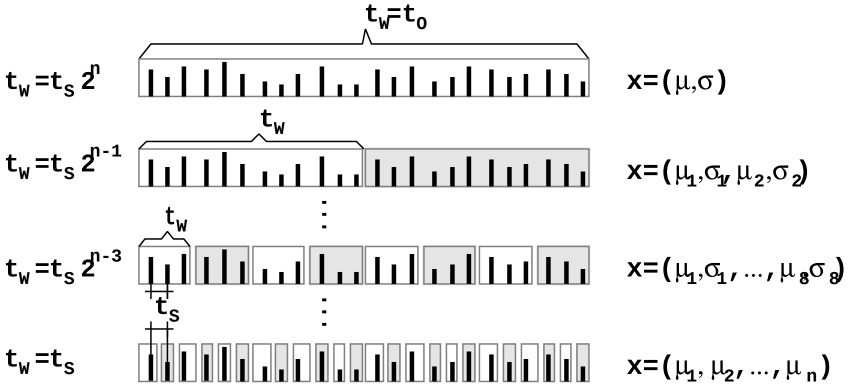


Figure 3.13: Synoptic of the general framework used throughout the rest of the section.

be calculated for each window i. For instance, going down one step, the window length is halved $t_W = t_O/2$, which doubles the number of features, producing $(\mu_1, \sigma_1, \mu_2, \sigma_2)$. Potentially, this dichotomic splitting procedure could continue until the window size reaches a minimum, given by the traffic sampling time (i.e.,. $t_W = t_S$). However, as 1-sample windows do not allow us to evaluate σ, in the following we disregard the degenerated case $t_W = t_S$ and consider $t_W \geq 2t_S$. It is worth noting that the case with 1-sample windows when the input is defined as the mean, is equivalent to the "embedding process" analyzed in Sec. 3.2. This means that the input is defined as a vector $x \in \mathbb{R}^d$, where the fact that $t_S = 1$ second, implies that $d = t_O$. In Sec. 3.3.3 we will show that the approach presented in this work brings better results (10% in the worst case) than the "embedding process".

For the sake of clarity, in the following we will refer to a forecast *horizon* t_F, based on an *observation* time frame t_O, which is possibly split into several time *windows* t_W, containing a number of samples collected using a constant *sampling time* t_S. Also, samples contained in the i-th window can be *consolidated* into a set of features, for example, (μ_i, σ_i) in this example: therefore, the SVM input will be the *union* of all feature sets, over all windows in which the observation time frame has been split. The impact of the above variables will be studied at length in the following, but, unless otherwise stated, we will refer to $t_F = t_O = t_W = 64$ seconds (i.e., the power of 2 closest to 1 minute interval), using a sampling interval of $t_S = 1$ second. In this case, no splitting is performed.

For the SVR training, we use a sliding window of length $t_O = t_F$ over this time series to build all possible inputs/outputs pairs (x_i, y_i). Again, a subset of this dataset is used for training, i.e., to solve the SVR problem and gather the forecast function $f(\cdot)$. The model accuracy is then evaluated over the complement of the training set, i.e., on unknown data. In the remainder, for each experiment 60% of the available data is used for SVM training and 40% for validation. Also, each experiment is repeated 30 times (using different training and validation sets), so that each experimental point corresponds to the average result over different SVR instances.

	ISP	Campus	WiFi	Ent
Period	May'06	May'06	Sep'07	Dec'04
Duration	15h	15h	15h	1h
Rate [Mbps]	60.1	30.0	28.8	7.7
Flows	$3.4{\cdot}10^6$	$6.2{\cdot}10^6$	$3.3{\cdot}10^6$	$58{\cdot}10^3$
Pkts	$461{\cdot}10^6$	$329{\cdot}10^6$	$195{\cdot}10^6$	$9.7{\cdot}10^6$
Bytes	$0.4{\cdot}10^{12}$	$0.2{\cdot}10^{12}$	$0.2{\cdot}10^{12}$	$3.5{\cdot}10^9$
Pkts/Flow	134	53	59	167
Bytes/Pkts	864	615	997	361
IP src	$2.8{\cdot}10^3$	$11.4{\cdot}10^3$	$3.4{\cdot}10^3$	$3.2{\cdot}10^3$
IP dst	$493{\cdot}10^3$	$699{\cdot}10^3$	$431{\cdot}10^3$	$2.7{\cdot}10^3$

Table 3.3: Traffic Traces Properties

3.3.1 Experimental Results

In this section we explore SVM performance for different inputs and outputs, for varying values of the temporal parameters and for different traces and traffic types.

In this case, SVR performance is affected by many parameters, belonging to two different classes: a first set is related to the SVR regression method and kernel function, whereas a second set pertains to the input (i.e., features) and output (i.e., target) spaces. As in Sec. 3.2.2, a "grid optimization" routine is used to systematically explore parameters of the first set, such as the smoothing factor C, the tolerance ϵ and the kernel parameter γ, and to select the tuple $(C^*, \gamma^*, \epsilon^*)$ that minimizes the prediction error. We have already shown that SVR is rather robust to the parameter selection, provided that the above parameters are selected in a "reasonable" range. Conversely, modification of the input features, the output target, etc., can significantly affect the forecast accuracy: therefore, in the following we will restrict our attention to parameters belonging to the latter class.

To quantify the forecast accuracy, we consider as before the RMSE (3.8). For reference purposes we consider the *naive* estimation, for which it is assumed that the value in the next horizon t_F will remain equal to the value achieved in the last observation time frame t_O. Thus, we may also express the relative RMSE gain of SVR forecast with respect to the naive prediction as:

$$G = \frac{(RMSE_{naive} - RMSE_{SVM})}{RMSE_{naive}}.$$

Other possible techniques to estimate a statistical property as the peak or the 95th-percentile implies the assumption of a model; we focus instead on a methodology that avoids making any assumption of the underlying phenomenon.

Experimental Dataset

Prior to investigate the SVM performance, let us briefly introduce the different real-world traces used throughout this work: we directly monitored an ISP access link and the campus egress router ports of our Ethernet and WiFi LANs, but we also make use of the enterprise network data made available by the LBNL/ICSI tracing project [61]. Details on these datasets are reported in Tab. 3.3, such as traces duration and bitrate, amount of flows, packets and bytes observed, average packet and flow lengths, count of distinct IP source and destination hosts.

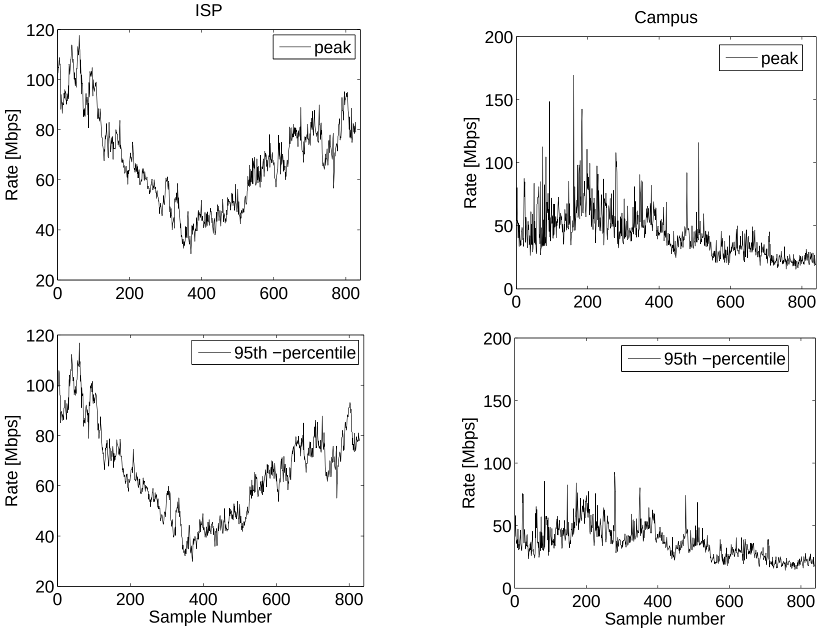
All traces are 15 hours long and were collected between May 2006 and September 2007, with the exception of the enterprise traffic, which is 1 hour long[2] and was gathered during December 2003. The ISP dataset is very peculiar, as it refers to an innovative ISP which is providing end users (residential, SOHO or large companies) with data, voice and video over IP by means of either an ADSL or a FTTH link, whereas no PSTN link is offered: clearly, all flavors of p2p applications are present in this downstream dataset. Ethernet and WiFi traces are typical examples of campus LAN downstream traffic, measured at the campus egress router, representing the aggregated traffic of the hosts having Ethernet or Wireless access respectively. Ethernet Campus traffic consists of a mix of Web, intranet services and Internet applications (a firewall tries to block p2p traffic, although some as, e.g. Skype, still manage to go through) whereas WiFi access is mostly used for Web browsing, mail and instant messaging. Finally, Enterprise traffic is also particular, as intranet services constitute the most important part of the traffic – for a thorough analysis of the LBNL/ICSI traffic, we refer the reader to [62].

Intuitively, these different traffic characteristics will translate into different prediction accuracy. For explanatory purposes, let us show in Fig. 3.14 the peak p and 95-th percentile p_{95} computed over 1 second long time-windows for both ISP and Campus traces. We easily realize that the prediction will be more difficult for the Campus trace since there are a lot of uncorrelated "spikes" in both series, especially considering the peak load. Conversely, it can be seen also, that for the ISP trace, both series are very similar, which make us expect the ISP trace to be a relatively easier forecast scenario with respect to the Campus one. It should be noted that the ISP and Campus dataset were already considered in the previous section (c.f. Sec. 3.2.2)

Features and Target Impact

Let us start by considering different statistical properties of the link load variable as target (i.e., *output* of the model), and by feeding SVR with different combinations of their previous observations (i.e., *inputs* of the model). As statistical properties, we consider the mean μ, standard deviation σ, peak p and 95-th percentile p_{95}. Also, as far as output is concerned, in the following we consider the problems of peak p and 95-th percentile p_{95} prediction. We show results only for the ISP and Campus dataset since they reflect the variety of performance results that can be obtained; we recall that we expect Campus to be a stiffer scenario, especially when the peak load prediction is considered.

[2] This is due to the measurement methodology in [62], where different switch ports are monitored every hour.



(a)

(b)

Figure 3.14: Peak p and 95-th percentile p_{95} of the link load for (a) ISP trace and (b) Campus trace.

As explained before, we preliminary fix SVR parameters (C,ϵ,γ) for each trace by performing a grid optimization for each output, using the pair (μ, p) as input (which is a reasonable choice given SVR robustness). We recall that $t_O = t_F = t_W = 64$ seconds, whereas $t_S = 1$ second. For this parameter setting we show in Fig. 3.15 real values and SVR predictions for a random validation set for both ISP and Campus traces, to get a first (visual) idea of the prediction accuracy. As expected, the predictions of both peak and 95th-percentile are more accurate for the ISP trace: SVR is unable to forecast the important "spikes" present in the Campus traces, despite it is able to "follow" the curve. Also intuitively, the presence of these spikes makes peak prediction harder than 95-th percentile one. These differences are reflected in the RMSE: for the ISP trace RMSE is 3.65 for the peak and 3.39 for the 95th-percentile, whereas for the Campus trace these values are 21.46 and 9.94 respectively.

In what follow we will explore the use of different inputs in addition to (μ, p). More precisely, we will consider as inputs all possible combinations of input features μ, σ, p and p_{95}. Tab. 3.4 reports the RMSE obtained by the naive prediction and the corresponding SVR gain for the ISP and Campus dataset and for these different combinations of input features. Specifically, we select:

- the three best combinations of 2-features inputs (i.e., over all *pair* of statistical properties)

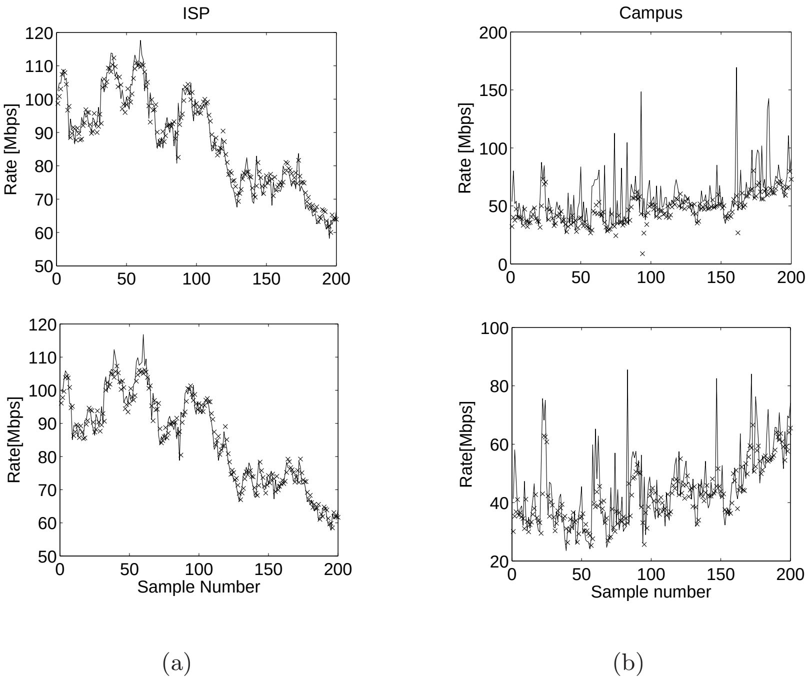


Figure 3.15: Comparison of real and predicted values for (a) ISP trace and (b) Campus trace.

- the average of all 2-features inputs
- the average of inputs with 3 or more features
- the average of *all* possible combinations irrespectively of their length

First, we observe that the performance of both SVR and naive forecast significantly changes when different *traces* are considered: the RMSE change by about a multiplicative factor between 2.5 and 4. Second, considering different *outputs* as p and p_{95}, the RMSE variation can be either important (roughly, a factor of 2 in the Campus case) or irrelevant (as in the ISP dataset): as we early noticed this can be explained by the presence of significant load spikes in the Campus trace, which are harder to predict and thus yield to a larger forecast error. Third, the *input* features combination does influence the results, although with a smaller magnitude. Interestingly, it can be seen that is better to use a small number of features to describe the statistical properties of the trace rather than a large one. Moreover, the three best input combinations yield to very similar results, although no clear winner can be identified, as the best input varies across traces and outputs. Therefore, in what follow, we will limit ourselves to consider (μ),(μ, p),(μ, p_{95}) as inputs for the SVM prediction. Finally notice that, rather surprisingly, the (μ, σ) combination yields to *worse* results with respect to the simplest input choice (μ), and that this holds furthermore for both traces and outputs. Also, the combination (σ, p) provides the *worst* results of all combinations, which suggest

		ISP		Campus	
Input		p_{95}	p	p_{95}	p
Naive	**RMSE**	**3.44**	**3.91**	**8.28**	**16.55**
	1st	5.1 % $\circ$	8.2 % $\star$	15.5 % $\bullet$	21.2 % $\bullet$
	2nd	5.0 % $\bullet$	8.0 % $\bullet$	15.1 % $\star$	20.9 % $\star$
SVM	3rd	4.6 % $\star$	7.9 % $\circ$	14.4 % $\circ$	20.8 % $\circ$
Gain	= 2 feat	4.7 %	7.7 %	14.6 %	20.1 %
	$\geq$ 3 feat	3.8 %	6.3 %	13.0 %	15.9 %
	All	4.3 %	7.2 %	13.9 %	18.3 %

Table 3.4: Comparison of Naive and SVR forecast of p and p_{95}, for different input features, where $\circ$=(μ),$\bullet$=(μ, p),$\star$=(μ, p_{95})

that the mean μ should always be considered as input feature, irrespectively of the statistical properties considered as target of the forecast.

In Sec. 3.3.2 we will show that it is actually unnecessary to inspect which combination of input yield the best results: the underlying idea is to use several machines in parallel, each of which is trained with different inputs for the same output (so that it is either possible to e.g., *combine* the forecast power of different machines, or also *automatically select* the best input combination).

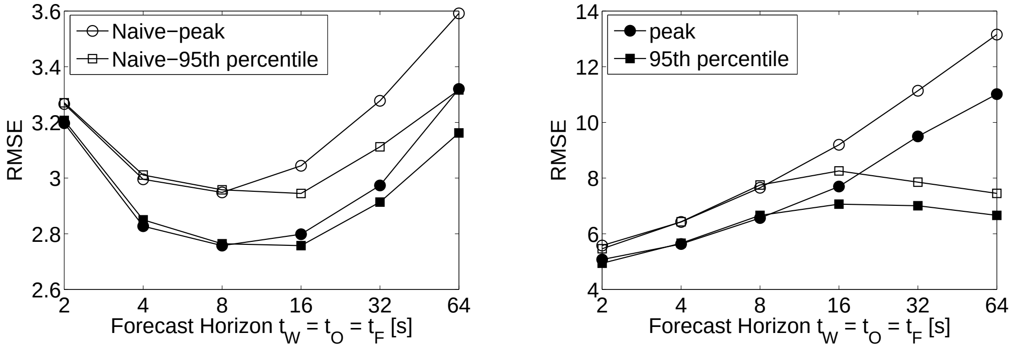
Aggregation and Timescales Impact

In this section, we explore the impact of *timing* related parameters, such as the forecast horizon t_F and the window length t_W. It is worth to note that we have fixed $t_S = 1$ second, since this choice implies more flexibility in the selection of the the forecast horizon and will allow us to explore a wider range of values for the aggregation time window t_W parameter. We focus again on the prediction of p and p_{95} load, considering only ISP and Campus datasets, and report the SVM results averaged over 30 repetitions for each of the three best 2-features input combinations described so far.

We first consider the forecast horizon $t_F = t_O$, assuming that the window t_W is consolidated into a single set of features (i.e. $t_W = t_O = t_F$). Neglecting the degenerated case $t_F = 1\,\text{s}$, we explore values of $t_F = 2^i\,\text{s}$ for $i \in [1, 6]$, where for instance $t_F = 16$ means that we observe an interval of 16 seconds (or 16 samples, since $t_S = 1\,\text{s}$) and predict the output value over the next 16 seconds interval. Results for naive and SVR prediction of p and p_{95} are reported in Fig. 3.16 for both the ISP and Campus traces [3].

Interestingly, the forecast accuracy is diversely affected by the forecast horizon for the different traces: this reflects the fact that network usage is much different across the datasets, and so are the temporal traffic dynamics. For example, considering

[3] Clearly, the results for small time scales are equal for peak and percentile, since in such time scales, they usually coincide.



(a)

(b)

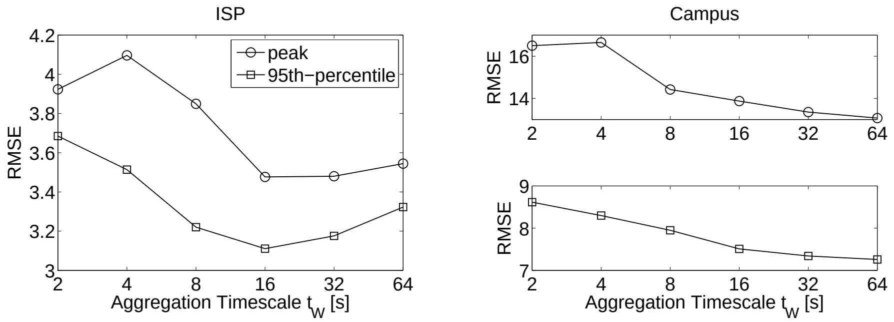
Figure 3.16: Impact of forecast horizon t_F on forecast accuracy for (a) ISP trace and (b) Campus trace.

the ISP trace, both p and p_{95} can be predicted with higher accuracy over a medium temporal horizon (i.e., $8 \leq t_F \leq 16$). Conversely, in the Campus dataset the p RMSE error monotonously increases with t_F, whereas the p_{95} RMSE exhibits an opposite symmetrical behavior with respect to the ISP case (i.e., medium values of t_F yield to worse performance).The increasing error for peak is in accordance with the important spikes shown in Fig. 3.15 since it could be easier to predict values in the near future being not possible to extend the forecast horizon. This fact have less impact when the percentile is considered as output since the spikes are not so important. Note that both predictors (Naive and SVR) are equally affected by the variations on the forecast horizon (i.e. the minimum is achieved at the same value of t_F).

While it is hard to draw general conclusions from these specific behaviors, nevertheless one can gather that some time horizon are definitively easier to predict. Pushing this intuition a bit further, we are interested in answering whether, aiming at predicting link load over *arbitrary* time horizons, it could be beneficial to aggregate data into windows corresponding to timescales where forecast is known to be more accurate. For instance, consider the Campus case: in order to predict next minute's p_{95} load, would it be better to use (i) a single set of features from the last minute window, or (ii) several sets of features gathered from separate smaller windows ? To answer this question, fixing $t_F = t_O = 64$, we consider the impact of the aggregation window t_W by splitting the observation time frame t_O into several windows of duration $t_W = 2^i\,$s with $i \in [1, 6]$ as described in Fig. 3.13. For instance, $t_W = 32\,$s means that the observation period t_O is split into two intervals of 32 samples, each of which gets consolidated into a different set of features: SVR is then fed with the *union* of these sets.

Results are reported in Fig. 3.17 in terms of the RMSE as a function of the window duration t_W. Moreover, we point out that, being the naive estimation RMSE constant for a fixed $t_O = t_F = 64$, the SVR gain is thus directly proportional to the error.

From comparison of Fig. 3.16 and Fig. 3.17, we have a partial confirmation of our



(a) (b)

Figure 3.17: Impact of aggregation timescale t_W on the SVM accuracy for (a) ISP trace and (b) Campus trace.

intuition: for instance, in both ISP p and p_{95} cases the minimum error at $t_O = t_F = 64$ is achieved when $t_W = 16\,\mathrm{s}$, which is precisely the value that minimized the RMSE in Fig. 3.17. In other words, there are cases where for an *arbitrary* forecast horizon t_F, it is preferable to split the observation time frame t_O into different windows $t_W < t_F$, whose duration t^*_W should be selected so as to minimize the forecast error.

At the same time, interpretation of Campus results is more complex, and undermine the generality of the above observation. Considering p_{95} estimation, it can be seen that Fig. 3.16 would suggest the use of small ($t_W \leq 4$) or even large ($t_W \geq 32$) windows – but not intermediate values of t_W, as the RMSE is concave in t_W. Fig. 3.17 states that smaller errors can be achieved by using large $t_W \geq 32$ windows (and consequently, a smaller number of features), which does not contradict our intuition, despite t_W value do not precisely correspond to the optimum of Fig. 3.16.

Prediction results of p are of harder interpretation, as the minimum error can be achieved when the observation time frame is *not* split (i.e., $t_W = 64$), despite the prediction error monotonously increases with $t_W = t_F$ as shown in Fig. 3.16. Thus, it seems as though the specific network scenario may play a very significant role in determining SVR forecast performance, and that moreover this happen in a non trivial way.

A possible, partial, explanation lies in the fact that shorter windows t_W translate into a higher number of features, which as early noted tends to “confuse” the SVR prediction: therefore, there may be cases where this increased number of features simply offsets the potential benefits brought by the splitting procedure.

Traces and Traffic Breakdown Impact

In this section, we analyze the different traces of Tab. 3.3 in the attempt to quantitatively bound the extent of SVR benefits. We limit the analysis to the peak load estimation at $t_F = t_O = 64\,\mathrm{s}$ and report in Tab. 3.5 the naive RMSE (R), as well as

	CoV		All		Srv/All		Services		Web/Srv		Web	
	∞	64 s	G%	R	B%	F%	G%	R	B%	F%	G%	R
I	0.29	0.05	12	3.7	1	2	22	1.1	8	2	15	0.5
C	0.40	0.17	21	13.4	44	21	19	14.2	66	38	18	12.5
W	0.88	0.54	11	10.6	92	89	11	10.6	91	80	10	10.6
E	0.64	0.56	25	10.2	62	37	27	10.0	27	33	10	4.9

Table 3.5: Naive RMSE error (R) and SVR gain (G%) for different traces (I,C,W,E) and for different traffic breakdown in terms of service type (All, Srv, Web), bytes (B%) and flows (F%)

the maximum gain (G%) brought by SVR (over all possible window values t_W).

First of all, the table reports the coefficient of variation (CoV), defined as the ratio of the standard deviation over the mean load and measured at two different timescales, as a statistical compact index of the link load variation. More specifically, CoV(∞) is evaluated over the whole trace, and represents the long-term load variability, whereas CoV(64 s) is the mean CoV evaluated over 64 s long windows and is representative of the traffic "burstiness" at short timescales. Notice that there is only a weak correlation between the CoVs and the forecast accuracy: for instance, RMSE for Campus is the highest despite its CoV is lower than both WiFi or Enterprise ones – which is reasonable since relevant forecast errors, corresponding to Campus load spikes, are quadratically penalized by the RMSE metric.

In order to test whether forecast accuracy not only depends on the traffic trace but on the traffic class as well, we partition each trace into different traffic aggregates and apply the SVR load estimation to each aggregate separately. Specifically, we first extract all "server" traffic from the trace, considering only those packets that involve well-known transport layer ports, and then extract a second aggregate constituted by Web traffic only (i.e., port 80).

Aggregates have a very different importance depending on the trace we are considering: e.g., ISP traffic volume is dominated by p2p traffic, while server traffic plays a major roles in the Enterprise trace and HTTP constitutes the dominant fraction of the WiFi dataset. Tab. 3.5 details the traffic breakdown across traces reporting the relative volume, expressed in terms of bytes (B%) and flows (F%), of Server/Total and Web/Server traffic.

We stress that this difference in the relative volume of traffic aggregates affects the RMSE value, which indeed also depends on the actual *scale* of the data. For example in the case of Web traffic for the ISP trace, the fact that RMSE is very small is also tied to the fact that Web browsing only represents a small fraction of the total traffic. For this reason, and to allow a more direct comparison across traffic aggregates, the table also reports the gain in percentage with respect to the naive prediction method.

From Tab. 3.5, is easy to gather that the gain over the naive estimation ranges between 10% and 30%, with an average of 17% – which is clearly a very significant gain. Moreover, results confirm that the specific dataset considered strongly affects the SVR performance, whereas the forecast accuracy does not exhibit a significant variation across the different traffic aggregates[4].

An important remark is that the optimal t_W^* value (i.e., the one reported in Fig. 3.16 to which the most accurate prediction when $t_W = t_F = t_O$ corresponds) only varies across *traces*, but remains the same over all *aggregates* for a given trace.

More precisely, the values of t_W^* that yield to the highest gain are $t_W = 64\,\mathrm{s}$ for Campus, $t_W = 16\,\mathrm{s}$ for WiFi or ISP and $t_W = 2\,\mathrm{s}$ for Enterprise respectively. Unfortunately, as early pointed out, there is no generally applicable guideline to properly select t_W. A possible strategy could be to use the *largest* possible window $t_W = 64$, which correspond to the *most concise* summary. A natural question is then to quantify the accuracy loss whether $t_W = 64$ is used instead of t_W^*.

Interestingly, the loss in SVR accuracy under this sub-optimal heuristic choice is actually very limited. Specifically, results are on average only 4% above the optimum, with a maximum of 11% for Enterprise traffic: furthermore, if we neglect the latter (too short) dataset, the mean accuracy loss drops to a very limited 2%. Therefore, we can conclude that, although properly setting the SVR parameters may be a cumbersome task, at the same time SVR models are robust enough to have very good performance even under non optimal settings.

3.3.2 Parallel SVR

In this section we study the performance obtained by using several SVMs in parallel, i.e. for the same output several machines are trained and the final prediction is a *combination* of the results obtained by the single machines. The idea is that the use of several parallel machines can increase the forecast power. In this case we limit ourselves to consider the peak p as output. In particular, a different SVM is trained for each of the following inputs: $(1) = (\mu)$, $(2) = (\mu, \sigma)$, $(3) = (\mu, p)$ and $(4) = (\mu, p_{95})$ (which are chosen according to Tab. 3.4). Different predictions are obtained form these machines and combined according to the following strategies:

I. the final prediction is the average of all predictions except the one that obtained the worst RMSE over the validation set

II. idem, but for each point in the validation set, we neglect the prediction that is *furthest* from the mean prediction of all machines

III. idem, but the prediction is defined as the one that is *closest* to the mean prediction

For the first case (I) the final prediction is defined depending on the performance (RMSE) of the different machines over all the validation set: the same combination is

[4] The most important variations appears for the Enterprise traffic, but the results may be negatively biased by the excessively short trace duration.

	Best	Worst	Most distant	Closest
ISP	(3) - 100%	(2) - 100%	(2) - 39%	(4) - 37%
Campus	(3) - 76%	(2) - 100%	(1) - 38%	(4) - 34%
WiFi	(3) - 69%	(1) - 100%	(1) - 60%	(2) - 35%

Table 3.6: Best and worst choice and percentage of most distant and closest to the mean prediction

	ISP		Campus		WiFi	
Strategy	Best	Worst	Best	Worst	Best	Worst
I	-0.5%	5.4%	0.1%	2.1%	0.1%	6.8%
II	-0.1%	5.7%	0.4%	2.4%	0.1%	6.8%
III	0.0%	5.8%	0.1%	2.1%	0.2%	6.9%

Table 3.7: Gain of parallel SVR over the best and the worst input

used for all points in this set. Conversely, in the last two cases the best combination is redefined for each point, i.e. the decision can be taken online with only punctual knowledge of the system.

It is our aim here to compare the performance of these three strategies with respect to the best and worst input combination, which possibly changes between datasets and outputs. Thus, for a given validation set, we determine the best and worst machine (in terms of RMSE). For each point in the validation set, we also identified which was the machine that predicted the value nearest and furthest to the mean of all predictions. This procedure is repeated 200 times and results are shown in Tab. 3.6 for ISP, Campus and WiFi datasets. The first two columns indicate the machine (or features combination) that obtained the best and worst RMSE in the majority of the repetitions (this majority is shown as a percentage in the table). The last two columns indicate the machine whose prediction was the closest and the most distant for the majority of points in the validation set and repetitions (again this majority is shown as a percentage).

Some remarks are in order. Regarding the first strategy, we find that the best input changes over the different repetitions, in accordance with the results already shown in Tab. 3.4. However the worst machine does not change, motivating the first considered strategy (I). More in details, since the worst machine is different across the traces, but not changes over a particular trace, a startup phase can be used to evaluate the worst machine which is then not considered for future predictions. Moreover we find out that the best (worst) input not necessarily coincides with the closest (most distant) predictor. This last observation justifies the consideration of the last two strategies (II and III).

We report in Tab. 3.7 the gain of using parallel SVMs over the best and worst results obtained by a single SVM. Performance are very similar for all the strategies, with strategy III obtaining slightly better results. Interestingly, *any* strategy employing

ISP		Campus		WiFi	
RE	RMSE	RE	RMSE	RE	RMSE
11.6%	10.3%	25.7%	14.8%	12.2%	11.8%

Table 3.8: Gain of parallel SVR over the "embedding procedure"

parallel SVM is also better than a single machine using all the input (μ, σ, p, p_{95}). Moreover, with respect to the best 2-feature input, the results of parallel SVR are generally better but sometimes worse (i.e. strategies I and II in the ISP case), with a difference always smaller than 0.5%. This means that although no real gain is obtained, at the same time no noticeable performance loss happens either.

In other words, the use of several machines in parallel can be an excellent strategy when the best input is not clear. A possible objection is that this approach may raise the computational cost of training several machines at the same time: to this extent, we stress that a common PC can support *several thousand* of such SVR machines in real-time 3.2.3.

It should be mentioned that more sophisticated techniques such as Boosting [63] or Bagging [64] can also be applied in this context. However, since they are more sophisticated they are more complex too, and the trade off between accuracy and simplicity (remember that the target is to do online predictions) should be carefully analyzed.

3.3.3 Comparison with the Embedding Process

Finally, we compare the performance of parallel SVM previously described and the "embedding process" described in Sec. 3.2. More precisely, we use the cascading procedure described and analyzed in Sec. 3.2.3 to predict all values on the forecast interval t_F, from which the maximum is then calculated.

We consider again the dataset ISP, Campus and WiFi, and the peak as the output, since it is the most difficult to predict. Given the results of the previous section, for parallel SVM we limit ourselves to the strategy III. Roughly, with this comparison we aim at knowing which approach is better to predict the maximum load in the next minute: the "embedding process" which uses as input a vector containing all the observed samples ($x \in \mathbb{R}^d$ with $d = 64$), or the approach presented in the previous section which uses an intelligent combination of statistical summary as input.

Results are shown in Tab. 3.8, averaged over 100 repetitions, where in each case 60% of the data is used as training and the remainder as validation set. Considering both the relative error (RE) or the RMSE, results indicate that for all traces an important gain can be obtained with the new approach: 15% on average and 10% in the worst case. It can be observed that a substantial RE gain is obtained for the Campus trace, which we already knew to be a stiffer scenario. This is an encouraging result, which shows that the drawbacks of the "embedding procedure" can be overcome, confirming that SVM is a very interesting technique for the purpose of link load prediction.

3.4 Conclusions

In Sec. 3.2 we have explored the use of Support Vector Regression for the purpose of link load forecast by means of the "embedding process". In this case, a series of averaged load at a given timescale is constructed and a future value of this series is forecast based on a given number of past measures. A very extensive sensitivity analysis of parameters impact was performed, including those of the SVR model and the embedding procedure, as well as the training set size. We compared SVR performance with those achievable by using Moving Average (MA), Auto-Regressive (AR) models and the Nadaraya-Watson estimator (NW). We found that, despite a good accordance with the actual data and that no one of the other considered methods can improve the results obtained by SVR, the gain achievable over simple prediction methods such as MA or AR is not sufficient to justify its deployment for link load prediction at short time scales. The gain over NW is also not significant, but the computational complexity of NW showed to be very much larger than that of SVR. Moreover, SVR showed a number of extremely positive aspects: for instance, SVR models are (i) rather robust to parameter variation, (ii) their computational complexity is far from being prohibitive, and (iii) the cascading of SVR models may significantly extend the achievable forecast horizon, entailing only a very limited accuracy degradation.

These results motivate us to the search of other approaches that we have analyzed in Sec. 3.3. In this case, a function of the link load over a given interval is predicted (in particular we concentrate in the peak and the 95th-percentile) based on a summary of statistical properties of observations taken from a past interval of equal length. We have investigated the impact of several parameters on the forecast accuracy and considering several real-work traces we gathered results that are representative of rather different network environments. Our main result is that combinations of several parallel SVR using different statistical summary as input, referred to as "parallel SVR", can provide accurate prediction. Indeed, a significant gain can be obtained not only over the naive estimation technique used as reference, but also with respect to the "embedding procedure". This work further provides some useful insights to tune SVR predictors: first, and as a rule of thumb, SVR accuracy improves when "compact" statistical summaries are used as inputs. Then, despite the best input feature set possibly depends on the forecast output, we have shown that the selection can be avoided by using parallel SVM. Concerning time-related parameters, our experiments show that their optimal tuning may not be easy task. Yet, we point out that the best setting only depends on the network measurement point but is insensitive to the traffic breakdown. Second, the use of large input time-scales (corresponding to compact input summary) yield to a near-optimal forecast accuracy. Third, we argue that the use of parallel SVM could bring benefits also concerning the tuning of time-related parameters.

Overall, we can conclude that SVR is an interesting technique for its flexibility, cost-effectiveness, accuracy and robustness. SVR is flexible since it applies to the forecast of different targets, while cost-effectiveness stems from the fact that both the offline training and the online forecast operations have a linear cost in the number of support vector. Moreover, given that a number of software tools are readily available, SVR techniques can be deployed right away. Finally, SVR has the potential to bring accurate results under a number of different scenarios: while parameter tuning may not be a trivial task, we stress that SVR performance is rather robust even under non optimal settings and that the use of parallel SVM helps in relieving this problem.

Chapter 4

P2P-TV Traffic Classification

The Internet proved to have an awesome capability of adapting to new services, migrating from the initial pure datagram paradigm to a real multi-service infrastructure. One of the most recent steps of this evolution is P2P-TV, i.e., large-scale real-time video-streaming services exploiting the peer-to-peer (P2P) communication paradigm. There are several currently deployed P2P-TV systems [3, 4, 5, 6], which feature low-quality streaming, while high-quality systems are just beyond the corner [7, 8].

In P2P-TV systems, hosts running the application, called *peers*, form an *overlay topology* by setting up virtual links over which information is transmitted and received. A source peer is responsible for injecting the video stream, by chopping it into segments of a few kilobytes, called *chunks*, which are then sent to a few other peers, called *neighbors*. Each peer then contributes to the chunk diffusion process by retransmitting chunks to its neighbors (accordingly to some scheduling scheme) following a swarming like behavior, somehow inspired from file sharing P2P systems like BitTorrent.

The main motivation to the deployment of P2P architectures is their ability to function, scale and self-organize in the presence of large number of nodes (whose behaviour is highly dynamic) and network failures, without the need of a central entity. The addition of new nodes requires more bandwidth but they also contribute to increase the available resources. Furthermore they are instantaneous to deploy and can enable the support of applications with minimal cost [65]. The major differences between P2P-TV systems and traditional P2P file sharing applications are that (i) the source is generating the stream in real time, (ii) data must be received by peers at constant rate, and (iii) chunks must arrive almost in sequence so that they can be immediately played at the receiver.

P2P-TV systems are candidates for becoming the next Internet killer application, as testified by the growing success of commercial systems such as PPLive, SopCast, TVAnts and many others. This raised the interest of the research community to understand their behavior and improve their performance, while the Internet Service

Providers (ISP) community have some concerns about them. Indeed, P2P-TV traffic may potentially grow without control, causing a degradation of quality of service perceived by Internet users or even the network collapse [9]. Unfortunately, many of the successful P2P-TV applications follow a closed and proprietary design, so that the algorithms and protocols they adopt are unknown. As such, the identification of P2P-TV applications is a topic of growing interest, which we address in the present chapter.

Much valuable effort has already been devoted to the problem of Internet traffic classification, and a large literature is available on the topic. After the classical *port-based* classification (which is no longer reliable), different approaches have been considered in the literature. These approaches can be roughly divided in: *Payload-based* [66, 67, 68], *Statistical-based* [69, 12, 11, 70, 71, 72] and *Behavioral-based* [13, 14] classification. In Sec. 4.1 a more detailed description of these approaches is presented, and a taxonomy of the most relevant works in the area is defined. In this chapter we will focus on the last approach (i.e. *Behavioral-based* classification).

This approach is very light-weight, as it requires neither to inspect portions of the packet payload as in [66, 68], nor to perform operations on a per-packet basis as in [72, 71]. Moreover, given the current tendency toward flow-level monitors such as NetFlow [73], the possibility to operate on the sole basis of behavioral characteristics is a very desirable property for classification engines. However, behavioral classification has so far attempted only a coarse-grained classification of Internet applications, identifying broad application *classes* (e.g., interactive, P2P, Web, etc.) rather than discriminating different applications within the same class. Thus, the design of a fine-grained classification engine that only exploits behavioral characteristics remains, to date, an open problem. Furthermore, it turns out that, given the implementation differences between the different P2P-TV applications, it may be easier to precisely identify a single application rather than the more coarse P2P-TV application class. As a beneficial effect, the fine-grained identification may allow both researches and ISP to conduct detailed analysis, e.g., explicitly comparing different systems or establishing a popularity ranking of the P2P-TV applications running on its network.

Motivated by the previous remarks, we designed a novel classification framework tailored for P2P-TV applications, filling thus a gap in the classification of Internet traffic. Our work fits in the *Behavioral-based* classification category, introducing at least three novel aspects. First, we perform *fine-grained* classification that distinguishes each particular application. Second, while much research has been devoted to P2P-TV application *characterization*, we are not aware of any P2P-TV classification engine. Third, we employ Support Vector Machines (Chap. 2) which have been rarely exploited (see for instance [74]) in the context of Internet traffic classification, using furthermore traditional transport layer information.

More precisely, our proposal relies only on the count of packets and bytes exchanged among peers during small time-windows: the rationale is that these two counts convey a wealth of useful information, concerning several aspects of the application and its inner working, such as, signaling activities, video chunk size, etc. It must be noted that this information is easily obtained by any peer in the network.

To validate the proposed classification engine, we carry out a thorough experimental campaign using both testbed traces and passive measurements collected from real network environments. Our results show that the percentage of correctly classified traffic exceeds 95% in the worst case. Moreover, the engine correctly labels as "unknown" the traffic generated by non P2P-TV applications, keeping the false positive rate (i.e., wrong classification of not P2P-TV traffic as such) below 1% in the worst case.

4.1 Related Work

Despite Internet traffic classification is a relatively recent research field, there is already a large literature on the topic. Since *port-based* classification has become unreliable (e.g. applications changing traditional ports to circumvent access controls, firewalls blocking, dynamic allocation of server ports, etc), three different classes of approaches have tried to solve this issue. *Payload-based* techniques [66, 67, 68], inspect the content of packets, looking for distinctive payload signatures. *Statistical-based* classification [69, 12, 11, 70, 71, 72] is based on the rationale that, being the nature of the services extremely diverse (e.g., Web vs VoIP), so will be the corresponding traffic generated (e.g., short packets bursts of full-data packets vs long, steady throughput flows composed of small-packets). Initially, work in this area focused on late traffic classification [69, 12, 11, 70] while early traffic classification has been also addressed more recently [71, 72] considering size and direction of the first few packets of a flow. Finally, Behavioral-based classification [13, 14]] targets a coarse-grained classification of Internet hosts on the sole basis of the transport layer traffic patterns they generate: e.g., P2P hosts will contact many different hosts typically using a single port for each host, whereas a Web server will be contacted by different clients with multiple parallel connections. The idea is that different classes of services and applications generate different patterns.

Table 4.1 aims to present a taxonomy of that we consider are the most relevant works in the area. The three main approaches mentioned above (i.e. Payload, Statistical and Behavioral) are further divided depending on the granularity and timeliness of the classification. Granularity refers to the level of detail known about the application. Generally speaking, applications can either be precisely individuated (e.g., discriminating between different VoIP applications, discriminating between P2P-filesharing versus P2P-TV applications, discriminating different database services, etc.) or only the coarse application class can be identified (e.g., data versus interactive versus P2P, without further discrimination possibility). The timeliness of the application refers instead to the duration of the classification process - from very short durations (e.g., a single or a few packets for early classification), to moderate durations (e.g., 10-100 packets or 1-5seconds for online classification), to post-mortem classification (i.e., at the end of the flow). Considerations on the computational cost and further comments are also reported.

Approach	Subcategory	Granularity	Timeliness	Comp. Cost	Comments
Payload Based	[66, 67] DPI	Fine-grained Individual appl.	Early first few packets	Moderate	Deterministic technique
	[70] SPI	Fine-grained Individual appl.	Online 100 pkts window	High	Robust technique
Statistical Based	[69, 12, 11, 70]	Coarse-grained Class of appl.	Late after the flow end	Lightweight	Post-mortem analysis
	[71, 72]	Fine-grained Individual appl.	Early first 5 packets	Lightweight	On the fly
Behavioral Based	[13, 14]	Coarse-grained Class of appl.	Late after the flow end	Lightweight	Post-mortem analysis
	this work	Fine-grained P2P-TV appl.	Online 1-5 sec. window	Lightweight	Online Only P2P-TV

Table 4.1: Taxonomy of classification techniques (DPI: Deep Packet Inspection - SPI: Stochastic Packet Inspection)

4.2 Classification Framework

4.2.1 The Rationale

Our aim is to classify P2P-TV *end-points*, which can be identified by IP address and transport layer port pair $(IP, port)$. Typically, P2P-TV applications rely on UDP as the transport protocol. During installation, a single UDP port is selected at random, over which all the signaling and video traffic exchanged with other peers is multiplexed. Therefore, all the traffic going to/coming from a given $(IP, UDP - port)$ endpoint is actually destined to/sourced from the same P2P-TV application running on the host. This holds true for P2P-TV applications like PPLive[3], SopCast[4], TVAnts[5] and Joost[6][1], which we take as examples throughout this chapter.

As mentioned before, we design a P2P-TV classification methodology that relies only on the evaluation of the *amount of information*, such as packets and bytes, exchanged by peers during small time-windows. The rationale is that a raw count of exchanged data conveys useful information concerning several aspects of P2P-TV applications.

A human analogy may help in clarifying the intuition. Suppose peers in the network are people in a party room: people generally have different behavior, e.g., they will be more or less talkative. As such, somebody may prefer lengthy talks with a few other people, whereas somebody else may prefer very brief exchanges with a lot of people. This is similar to what happens with P2P applications: some applications continuously perform peer discovery by sending a single packet to a previously not-contacted peer; others tend to keep exchanging data with the same peers as long as possible.

Additionally, most P2P-TV applications have been designed around the concept

[1] Joost became a Web-based application in October 2008, but at the time when the experiments were performed it offered VoD and live-streaming by P2P.

of "video-chunks", i.e., small units of information whose size is a typical parameter of each application. P2P-TV video service has an almost *constant* downlink throughput, due to the nature of the video stream. By tracking its *breakdown* between the different contributors it is possible to highlight different policies that a particular application can adopt, namely, fetching chunks from many neighbors, or downloading from a restricted list of preferential peers. Yet, while any P2P-TV peer consumes equally, the amount of uploaded data can be significantly different from peer to peer, due to different configuration, such as upload capacity. For example, in [75], it is shown that uplink to downlink throughput ratio for PPLive, roughly varies in the $[0, 10]$ Mbps range. In reason of the above observation, we assume that the classifier is situated at the *edge* of the network (where all traffic exchanged by a given end-point transits), and consider only the *downlink* direction, i.e., traffic coming from the Internet and crossing the edge of the network into the end-point direction.

In the following, we restrict our attention to UDP traffic since, as we mentioned before, all the applications considered in this work largely prefer UDP as the transport layer protocol. However, endpoint identification can be extended to applications relying on TCP at the transport layer as well. Indeed, in case TCP is used, the client TCP port is ephemeral (i.e., randomly selected by the Operating System for each TCP connection active open), thus requiring more complex algorithms to aggregate traffic directed to a peer.

4.2.2 Behavioral P2P-TV Signatures

Let us consider the traffic received by an arbitrary end-point $\mathcal{P} = (IP, port)$ during an interval of duration ΔT. We simply evaluate the amount of information received by $\mathcal{P}$ as the number of received packets. In Section 4.4.3 we extend this concept to account also for the amount of bytes.

We partition the space $\mathbb{N}$ of the possible number of packets sent to $\mathcal{P}$ by another peer into $B_n + 1$ bins of exponential-size with base 2:

$$I_0 = (0, 1], \quad I_i = (2^{i-1}, 2^i] \; \forall i = 1, \ldots, B_{n-1} \quad \text{and} \quad I_{B_n} = (2^{B_n - 1}, \infty].$$

For each ΔT interval, we count the number N_i of peers that sent to $\mathcal{P}$ a number of packets $n \in I_i$; i.e., N_0 counts the number of peers that sent exactly 1 packet to $\mathcal{P}$ during ΔT; N_1 counts the number of peers that sent 2 packets; N_2 the number of peers that sent 3 or 4 packets and, finally, N_{B_n} is equal to the number of peers that sent at least $2^{B_n - 1} + 1$ packets to $\mathcal{P}$. Let also K denote the total number of peers that contacted $\mathcal{P}$ in the interval.

The behavioral signature is then defined as $\underline{n} = (n_0, \ldots, n_{B_n}) \in \mathbb{R}^{B_n + 1}$, where:

$$n_i = \frac{N_i}{\sum\limits_{j=0}^{B_n} N_j} = \frac{N_i}{K}$$

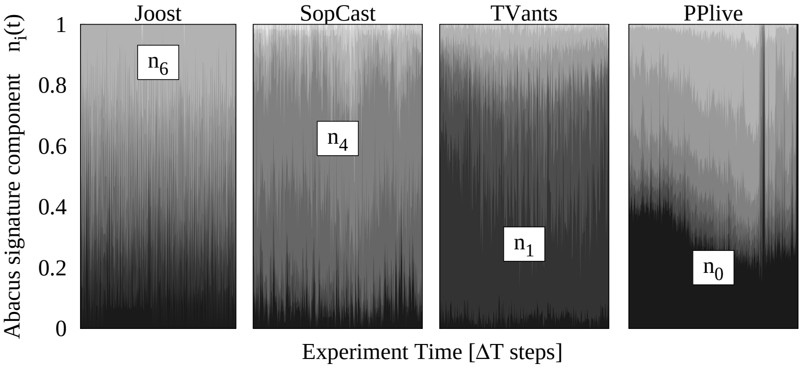


Figure 4.1: Temporal evolution of P2P-TV applications signature with a mark for the widest bin.

The signature $\underline{n}$ is the observed probability mass function (pmf) of the number of peers that sent a given number of packets to $\mathcal{P}$ in a time interval of duration ΔT; this function is discretized according to the exponential bins described above.

Since $\underline{n}$ has been derived from the pure count of exchanged packets, we name it "Abacus", which is also a shorthand for "Automated Behavioral Application Classification Using Signatures".

The choice of exponential width bins allows us to work with "short" signatures while keeping the most significant information that can be provided by the pmf. In this way, we can distinguish short flows and group together the long ones, since there is no need to have an extreme accuracy when considering long flows (e.g., distinguishing between 500 or 501 packet long flows), while it is more important to distinguish between short flows. We examine the impact of different binning strategies later on in Sec. 4.4.2.

Prior to describe the whole classification process, let us show the expressiveness of the Abacus signatures by presenting a few examples considering the four targeted P2P-TV applications. The temporal evolution of $\underline{n}$ is shown in Fig. 4.1, considering an arbitrary peer during 1-hour long experiment (from left to right, Joost, SopCast, TVAnts and PPLive are reported). Time of the experiment runs on the x-axis in multiples of $\Delta T = 5$s, whereas components n_i are reported on the y-axis using different levels of gray to highlight different bins (specifically, darker colors correspond to lower bins). Bins are ordered and staggered from bottom to top: thus, bin number 0 is the lowest one, and bin number B_n extends up to 1.

Each application has its own characteristic distribution, which is extremely different from the others. Moreover the largest (or most probable) bin, which is highlighted in the figure, is also characteristic of each application. Interestingly, the most probable bin remains the same during most of the application lifetime, despite its actual width varies over time. Notice that the breakdown is not stationary over time for all applications: this is for instance the case of PPLive, as it emerges from the rightmost plot of Fig. 4.1, which hints to transient or possibly "multi modal" behaviors. The dark vertical line

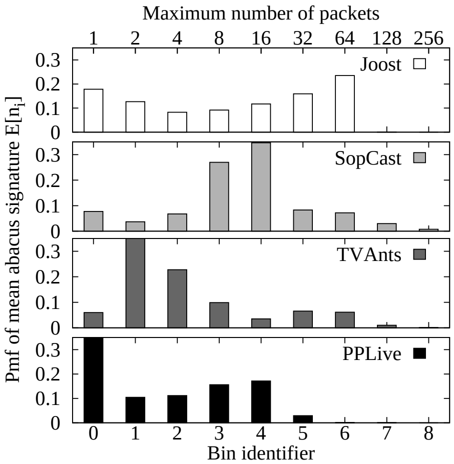


Figure 4.2: Mean value for each component of Abacus signatures.

toward the end of PPLive experiment corresponds to a sudden massive increase of n_0, due to a 10-seconds long blackout period, where the end-point under observation was essentially receiving single-packet probes, and likely no video chunks.

To better highlight how Abacus signatures capture the differences between applications, Fig. 4.2 reports the *time average* of the application signatures, averaged over all time intervals of Fig. 4.1. Bin identifier is reported on the x-axis, with top x-axis showing the maximum number of packets within the bin.

Interesting behaviors stand out from the picture. For instance, Joost peers preferentially receive either a single or several $(32, 64]$ packets from any given peer. SopCast instead prefers middle-sized burst of $(5, 16]$ packets, while TVAnts prefers lower order bins $(2, 7]$ packets. Finally, PPLive highly prefers single packet exchanges. This confirms that different P2P-TV applications have remarkably different behaviors, just like humans at a party: in the following we will exploit this evidence for classification purposes.

4.3 Methodology and Dataset

Our classification framework exploits the Abacus signatures described so far through SVM (extensively described in Chap. 2). We use `LibSVM` implementation [47] of SVM. In the context of SVM, entities to be classified are represented by means of some distinctive features, which in our case are the Abacus signatures. As early explained, SVM must first be trained with a supervised input (i.e.,features and class labels), called *training set*. The output of this phase is a *trained model*, which is then exploited to discriminate traffic during the validation phase of the classification process.

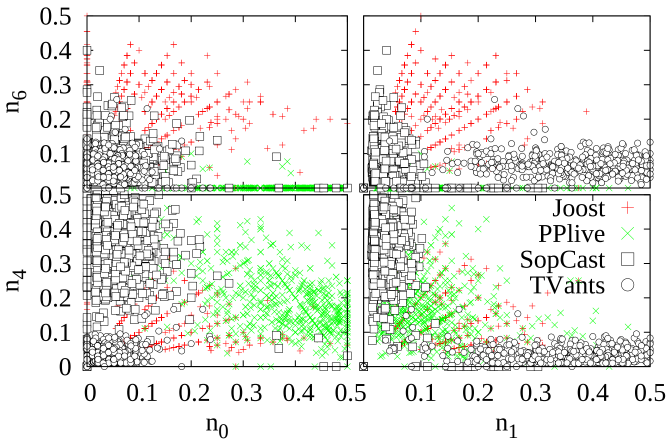


Figure 4.3: Some examples of Abacus space.

4.3.1 Model Generation

Our classification problem relies into the multi-class classification described in Sec. 2.2.4. As mentioned before LibSVM implements a "one-versus-one" strategy were $k(k-1)/2$ machines are trained, each one considering data corresponding to two different classes. In our case $k = 4$, corresponding to the fourth different applications considered so far.

The training phase correspond to the procedure of building the *separation surface* (see Sec. 2.2.2). In our case, Abacus signatures $\underline{n}$ are used to represent patterns belonging to different applications in a $(B_n + 1)$-dimensional space, which is then transformed into an infinite-dimensional space by means of a Gaussian kernel (Sec. 2.2.3).

For the sake of illustration, Fig. 4.3 reports a few "cuts" of the Abacus space before transformation, represented as bi-dimensional scatter plots of signature components pairs (n_j, n_k). In the plots, each point corresponds to a different time interval of those shown in Fig. 4.1. It can be seen that for instance, SopCast and TVAnts cannot be discriminated using the (n_0, n_6) components (top left plot), but that are instead easily separable (even in the original space) by means of a straight line in all other cases reported in Fig. 4.3.

To generate a classification model, SVM needs to be trained with supervised inputs, as shown in the top-portion of Fig. 4.4. Using testbed traces described later in this section, we generate signatures of known P2P-TV applications; specifically, for each trace, we build an Abacus signature $\underline{n}$ every interval of ΔT seconds. The signature is possibly chosen, at random, to be used for the *training set*. We will come back on the training set selection later on in Sec. 4.4.2. Once the SVM has been trained, the classification tool works as indicated in bottom part of Fig. 4.4. Abacus signatures of the traffic to be classified are computed and fed to the SVM model, who decides which application has generated it: finally, SVM classification is accepted provided it passes

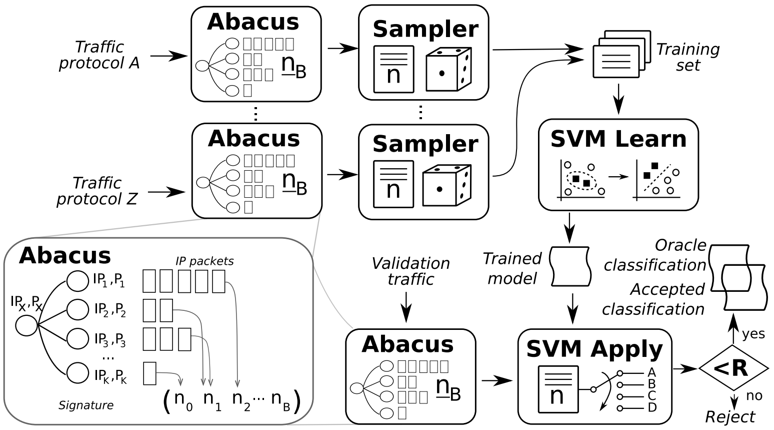


Figure 4.4: Classification framework.

a *rejection criterion*, to correctly discard non P2P-TV traffic.

4.3.2 Rejection Criterion

As previously mentioned, the multidimensional space is *partitioned* into regions and, during the classification process, each sample is *always* labeled as belonging to one region; i.e., the observed traffic is always classified as belonging to one of the classes the SVM has been trained with. However, since not all samples are necessarily generated by P2P-TV applications, we need to define a *rejection criterion* according to which we classify some traffic as not belonging to any of the considered classes.

Given that Abacus signatures are probability mass functions, we use an index suitable to quantify distribution similarity. Given two pmf's, there exist several indexes to evaluate their degree of similarity. The Bhattacharyya distance (BD) [76] is a measure of *divergence* of two probability density (or mass) functions. Given two pmf's p and q over n discrete values, the Bhattacharyya distance $BD(p,q)$ is defined by:

$$BD(p,q) = \sqrt{1-B} \quad \text{where} \quad B = \sum_{i=1}^{n} \sqrt{p_i q_i} \tag{4.1}$$

Bhattacharyya distance has several properties. First, it verifies the triangular inequality. Values of BD close to zero indicates strong similarity (if $p_i = q_i\ \forall i$, $B = 1$ and $BD = 0$) whereas values close to one indicates weak similarity. The Bhattacharyya coefficient $B \in [0,1]$, can be seen as the scalar product between the two vectors $p' = (\sqrt{p_1}, \ldots, \sqrt{p_n})$ and $q' = (\sqrt{q_1}, \ldots, \sqrt{q_n})$, which leads to a geometric interpretation of the coefficient B. In fact it can be seen as the cosine of the angle between p' and q'. The Bhattacharyya distance is a particular case of the Chernoff distance, and has been successfully applied in different contexts such as signal selection [77], or classification [78] as in our case.

In our context, we use BD to measure the *separability* of two traffic classes. In particular, we *reject* the SVM label C of a sample signature $\underline{n}$ whenever the distance

$$BD(\underline{n}, E[\underline{n}(C)]) > R \tag{4.2}$$

where $E[\underline{n}(C)]$ is the average signature computed on the training samples of application C. Notice that the average signature $E[\underline{n}(C)]$ identifies the center of the cluster formed by all training set signatures of application C. In other words, we accept SVM decision conditionally to the fact that the observed traffic signature $\underline{n}$ lies within a radius R from the center of the SVM training set for that class. For the time being, we select $R = 0.5$, and defer further discussion on the selection of R to Sec. 4.4.2.

There exist some cases where no false alarm can be raised (i.e., non P2P-TV traffic will be always classify as unknown), which makes Abacus robust by design. Let us consider the case when traffic is received from only one peer. Then, the Abacus signature $\underline{n}$ is a vector containing a single 1 at the coordinate (bin) i^*. In this case, the distance from the center $E[\underline{n}(C)]$ (C for short) of the cluster of an arbitrary application will be $BD(\underline{n}, C) = \sqrt{1 - \sqrt{C_{i^*}}}$. Then, the signature will be rejected if $BD(\underline{b}, C) > 0.5$ or equivalently

$$C_{i^*} < (1 - 0.5^2)^2 = 0.56 \tag{4.3}$$

which is actually always the case as we can see in figure Fig. 4.2, where all C_i are strictly lower than 0.4 for all applications. In the same way, if $K = 2$, any signature is a linear combination of two unit vectors and it can be proved that it will be rejected too. Therefore, if $K \leq 2$, the signature is always rejected (classified as "unknown") and therefore no false alarm is raised.

4.3.3 Model Validation

Assessing classification performance is known to be a non-trivial task due to the difficulty to know the "ground truth", i.e., what was the actual application that generated the traffic [67]. Testing the classification engine by means of artificial traffic (e.g., by generating traffic in a testbed) solves the problem of knowing the ground truth but synthetic traces are hardly representative of real world traffic.

Assessing the performance against traffic traces collected from operative networks is therefore mandatory. Moreover, even when considering real traffic traces, performance of the classifiers can be affected by the scenario (e.g, corporate and residential networks have very different traffic mixes). However, the major problem when dealing with real traffic traces is finding the ground truth.

In reason of the above trade-off, we adopt a mixed approach: we use both (i) traces actively gathered in a large scale testbed, and (ii) passive traces collected from different real operational network environments. Testbed traces contain P2P-TV traffic only and allow us to evaluate the engine capability to correctly discriminate P2P-TV applications. Conversely, real network traces do not contain P2P-TV traffic and allow us to verify that the engine correctly handles unknown applications.

Host	Site	CC	Access	Nat	FW
1-4	BME	HU	high-bw	-	-
5			DSL 6/0.512	-	-
1-9	PoliTO	IT	high-bw	-	-
10			DSL 4/0.384	-	-
11-12			DSL 8/0.384	Y	-
1-4	MT	HU	high-bw	-	-
1-3	FT	FR	high-bw	-	-
1-4	ENST	FR	high-bw	-	Y
5			DSL 22/1.8	Y	-
1-5	UniTN	IT	high-bw	-	-
6-7			high-bw	Y	-
8			DSL 2.5/0.384	Y	Y
1-8	WUT	PL	high-bw	-	-
9			CATV 6/0.512	-	-

Table 4.2: Summary of the hosts, sites, countries (CC) and access types of the peers involved in the testbed.

Application	Hr	Signatures	Packets	Bytes
SopCast	36	26k	17.2M	7.5G
TVAnts	36	26k	14.2M	7.1G
PPLive	26	19k	11.7M	5.1G
Joost	30	22k	6.1M	6.4G
Total	128	93k	48.2M	26.1G

Table 4.3: Details about the Testbed Traces.

Testbed Traces

To overcome the testbed representativeness limitation, we setup a large testbed involving multiple measurements points. The testbed was setup in the context of NAPA-WINE, a 7th Framework Programme project funded by the EU [9]. The testbed involved more than 30 controlled peers, hosted at 7 different Institutions, scattered over 4 European countries and connected to 9 different Autonomous Systems (see Tab. 4.2 for details). During the experiments, each PC ran the same application for one hour, during which all involved peers were forced to watch the same channel at the same time. SopCast, TVAnts, PPLive and Joost were run, and packet-level traces were collected during experiments[2].

In all (but Joost) cases, the nominal stream rate was 384 kbps (∼550 kbps) and Windows Media 9 Encoder was used. Further details concerning the experimental testbed traces are available in Tab. 4.3, which reports the overall duration, number of signature samples (when $\Delta T = 5s$), packets and bytes observed for any given application.

Overall, the testbed is representative of about 130 hours worth of video streaming, 93k signatures samples, 48M packets and 26 GBytes of data. Moreover, as different

[2] Traces differ because during the experiment some application could not successfully run, e.g., due to peer failure, or bad network condition.

Network	Traffic	Signatures	Packets	Bytes
Campus	UDP	1.9M	73.6M	10.6G
	Skype	0.5M	11.9M	2.2G
	DNS	0.2M	5.0M	0.7G
ISP	UDP	0.7M	28.5M	24.9G
	eDonkey	0.3M	9.8M	1.4G
	DNS	24.4k	0.6M	37.8M

Table 4.4: Details about Real Traces, only end-points for which $K > 2$ are considered.

network setups (e.g., access technologies, security policies, private/public addresses etc.) and peers configurations (hardware, OS, etc.) were part of the testbed, we are confident that the heterogeneity of the dataset is representative of a wide range of scenarios.

Real Traces

Real traffic traces were collected from two different networks in Italy. In both cases, traces were collected during May 2006, when P2P-TV applications were not popular in such networks:

- Campus (C) is a 5-days long trace, collected during one working week at the edge router of Politecnico di Torino LAN, which is representative of a typical data connection to the Internet [79]. The LAN contains about 7000 hosts, whose users can be administrative, faculty members and students. Most of the traffic is due to TCP data flows carrying web, email and bulk traffic, since a firewall blocks all P2P file sharing applications.

- ISP (I) is a 1-day long trace collected from FastWeb [80], which is the main broadband ISP in Italy, offering triple-play services over an all-IP architecture to more than 5 millions of users. FastWeb network is representative of a very heterogeneous and uncontrolled scenario, in which customers are free to use the network without restriction. Traffic is sniffed at a PoP level, to which about 500 users are connected, using more than 2000 different IP addresses considering VoIP phones, set-top-boxes, PCs.

The above traces contain no P2P-TV traffic. As such, they are instrumental to assess the amount of false classification, i.e., non-P2P-TV traffic classified as P2P-TV, produced by the classification engine.

As reported in Tab. 4.4, we consider both the total aggregated UDP traffic produced by all applications in the Campus and ISP traces, as well as relevant UDP traffic subsets, representative of both P2P and client-server applications. The rationale of this choice is that we want to test whether false-positive classification is more likely to arise

	P2P-TV	Non P2P-TV	Total
P2P-TV	TP	FN	TP+FN
Non P2P-TV	FP	TN	FP+TN

Table 4.5: Classification performance parameters.

when considering P2P applications or traditional client-server services. Specifically, we consider Skype and eDonkey traffic as examples of voice and file-sharing P2P applications, and DNS as an example of traditional client-server service. More precisely, Skype traffic is extracted from the CAMPUS dataset (since it is not popular in the ISP network due to flat rate telephony tariff), while eDonkey traffic is extracted from the ISP trace (since it is filtered by the CAMPUS firewall).

To reliably identify eDonkey, we develop and implement a DPI classifier, based on [81, 82], while we classify Skype resorting to [70], and rely on Tshark [83] protocol inspection capabilities to isolate DNS traffic.

Considering the Real Trace dataset, the condition $K \leq 2$ is verified for a large fraction of the signatures: 62% and 82% in CAMPUS and ISP respectively, to which a 72% and 85% of UDP volume corresponds. Thus, in reason of (4.3), for a very large portion of the UDP traffic misclassification is not possible. To gather conservative results, in the following we consider only those end-points for which $K > 2$.

4.4 Experimental Results

This section reports the results of our experimental campaign. We start by considering signatures that are defined on the number of packets exchanged, investigating the sensitivity of the classification engine to all the relevant parameters. Then, we show how classification performance can be improved by jointly considering the amount of packets and bytes exchanged in the signature definition. More precisely, classification performance are expressed in terms of (see also Tab. 4.5):

- the amount of True Positive (TP) classification, i.e. number of tests for which the classifier identifies the correct P2P-TV application given that the sample was actually of that P2P-TV class. TPs are counted considering the testbed traces.
- the amount of False Positive (FP) classification, i.e. number of tests for which the classifier identifies the sample as any of the P2P-TV application, despite the sample was not actually of any P2P-TV class. FPs are counted considering the real traces.
- the amount of True Negatives (TN), i.e. tests in which the classifier correctly rejects the sample (does not classify it as P2P-TV) which was indeed not generated by a P2P-TV application.

	PPLive	TVAnts	SopCast	Joost	Unk
PPLive	**81.66**	0.58	9.55	2.32	5.90
TVAnts	0.49	**98.51**	0.18	0.77	0.04
SopCast	3.76	0.11	**89.62**	0.32	6.19
Joost	2.84	0.55	0.28	**89.47**	6.86

	PPLive	TVAnts	SopCast	Joost	TNR
CAMPUS	2.42	2.23	0.01	0.02	**95.3**
ISP	0.66	0.13	0.43	0.10	**98.7**

Table 4.6: P2P-TV Classification Performance: Confusion Matrix of Testbed and Real Traces

- the amount of False Negatives (FN), i.e. tests for which a sample of a P2P-TV application was misclassified (rejected or classified as another P2P-TV class).

TP and FP results are usually normalized with respect to the total positive (and negative) samples. The TP-Rate (TPR) and the FP-Rate (FPR) are defined as follows:

$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{FP+TN}, \quad FNR = \frac{FN}{TP+FN}, \quad TNR = \frac{TN}{FP+TN}.$$

4.4.1 Classification results

In this first set of experiments, we report results considering the following parameters: for each application, the training set include samples extracted considering 2 peers at random from each group of $N = 7$ networks taking part to the experiment. From all signatures they generate, 4000 signatures are randomly extracted to define the training set, which corresponds to about 17% of all signatures. Experiments are then repeated 10 times, randomly changing the training set and so the validation set at each run, finally average classification results are computed. As other parameters, we consider signatures to be generated over $\Delta T = 5$s intervals. Classification are performed using SVM with a Gaussian kernel and exponential bins $B_n = 8$, with a rejection threshold $R = 0.5$. Parameters sensitivity and optimization is later discussed in the remaining part of this section. However, we shall note that the SVM parameters were optimized by means of a grid optimization as we did in Chap. 3; in particular see Sec. 3.2.2 for a complete discussion on the subject.

Top part of Tab. 4.6 reports the classification performance, adopting a "confusion matrix" representation. Each row considers traffic of a given application, and each column reports classification results. Values on the diagonal correspond to True Positives (highlighted in bold), whereas elements outside the diagonal correspond to False Negatives, which accounts for both misclassified samples, and rejected samples (labeled as Unknown). It can be seen that, in the worst case, about 81% of individual signatures are correctly classified. The most difficult application to identify appears to be

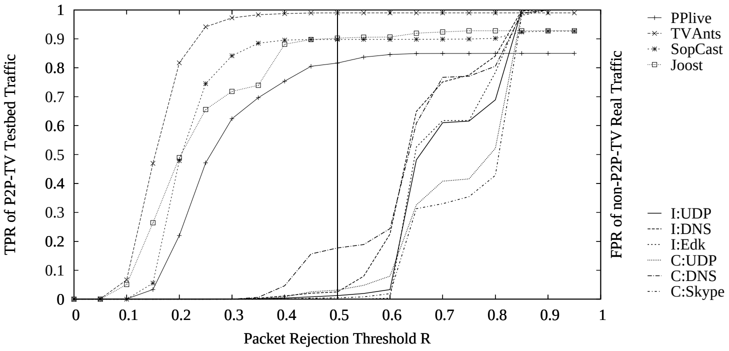


Figure 4.5: TPR and FPR as a function of the rejection threshold R evaluated on packet feature.

PPLive, which is confused with SopCast (9.55%) or Joost (2.32%). Other applications show higher TPR, with TVAnts showing almost perfect match. On average, about 4.5% of P2P-TV signatures are rejected, therefore being labeled as Unknown.

Bottom part of Tab. 4.6 reports results considering the real traces dataset. Since no P2P-TV traffic is present in this dataset, True Negative Ratio (TNR) is the main index to be considered (boldface, rightmost table column). Results show that the rejection criterion adopted is very robust, so that less than 5% of samples are misclassified in the worst case. Left part of the Table details the False Positives: PPLive and TVAnts are the cause of most misclassification, while Joost practically causes no False Positives.

4.4.2 Sensitivity Analysis

Impact of the Rejection Threshold R

The rejection threshold R is crucial to classification accuracy. Leaving the other parameters unchanged, we now assess its impact over the classification performance. The selection of the threshold R is guided by the following trade off: R should be large to maximize the TPR (i.e., avoid classifying P2P-TV as Unknown), while R should be small to minimize the FPR (i.e., avoid classifying unknown traffic as P2P-TV).

To investigate which rejection policy to adopt and which threshold value R to select, we evaluate the TPR (using Testbed traces) and FPR (using Real traces) as a function of R: these are depicted in Fig. 4.5, where a solid vertical line at $R = 0.5$ represents the threshold used so far. It can be seen that TPR of P2P-TV applications increases with R (for small values of R most of the classifications will be mistakenly rejected) and quickly saturates for $R \geq 0.5$. Conversely, the FPR of non-P2P-TV traffic increases only for large values of R, and for low values of $R \leq 0.5$ almost no false alarm is raised. This confirms $R = 0.5$ to be a good choice for the rejection threshold.

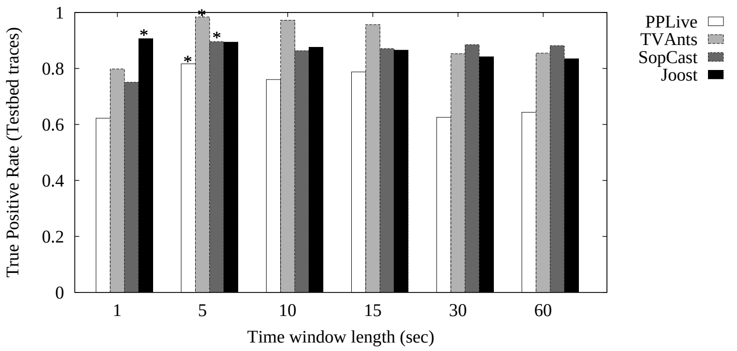


Figure 4.6: P2P-TV TPR for different values of the time interval ΔT. Best-case for each application is labeled with a star $*$ sign.

Impact of Time Interval ΔT

The choice of the value of the ΔT parameter is driven by the following trade off. On the one hand, timely detection of P2P-TV traffic needs ΔT to be small. On the other hand, sufficiently large time intervals must be considered to estimate the signature. Moreover, to limit computational complexity and the generated amount of information, network monitoring probes typically operate on large timescales. One would expect that in both extremal cases (too small o too large), the signatures tends to be more similar (concentrating in low or high bins respectively) and partly loosing their discriminative power.

Results are reported in Fig. 4.6, where the best results for each application is labeled with a star. As expected, medium-duration window (e.g., $\Delta T = 5\,\mathrm{s}$) yields higher TPR for most applications, while providing a more timely classification. Smaller values of ΔT limit the estimation of the bin distribution, impairing classification accuracy. Interestingly, for large windows (e.g., $\Delta T = 60\,\mathrm{s}$) the discriminative power of the Abacus signatures only mildly degrades for three out of four applications. Only for PPLive we observe a decrease of approximately 20% for the TPR, which is due manly to the rejection criterion being too aggressive and discarding correct classifications. We conclude that for longer time interval the rejection criterion should be more fine tuned.

Impact of Training Set Size and Diversity

We now assess the classification sensitivity to variations on the training set *size*, i.e., the impact of the number of samples that form the training set. Indeed, the training set should be large enough to be representative of the application behavior under a large range of conditions. On the other hand, the SVM training and classification computational costs benefit of a smaller set. Moreover, a too large training set could result in the well-known phenomenon of *over-fitting*, resulting in poor classification performance.

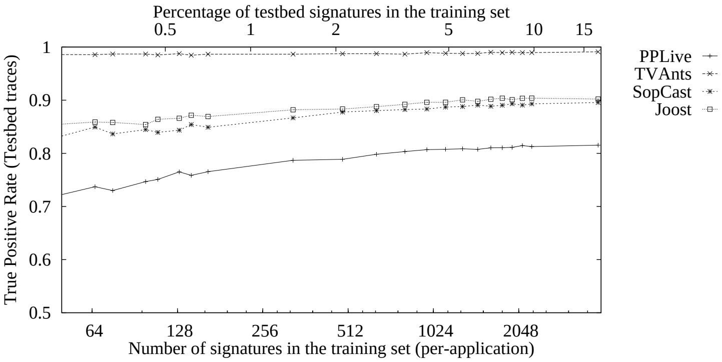


Figure 4.7: Impact of the training set size.

Fig. 4.7 reports the TPR for each application, as a function of the number of signatures used in the training phase per each application. For each value of the training set size, we run 10 independent experiments over which results are averaged. The bottom x-axis reports the number of signatures used for training on a logarithmic scale, while the upper x-axis reports the percentage of training samples versus the total testbed dataset. Training set size extends up to 4000 signatures per application, which corresponds to the 17% early used in Sec. 4.4.

Results show that no over-fitting phenomenon is experienced, since the TPR increases with the increase of the training set size. Best results are obtained considering 4000 signatures per application, which validates the choice made in previous section. Notice that even by drastically decreasing the training set size to about 300 signatures per application, the corresponding decrease in TPR is only modest, e.g., 8% in the worst case, while TVAnts shows excellent results even with an *extremely* reduced training set. This interesting performance is the result of both the discriminative power of SVM, and the descriptive expressiveness of Abacus signatures. Clearly, a better characterization of each application behavior is achieved including more signatures, as reflected by the improved performance.

We now fix the training set size and focus on the training set *diversity*, i.e., the number of different peers from which signatures are selected. Our aim is to roughly assess whether it is sufficient to observe a single peer in a given network to gather an adequate description of the application behavior in that network, or whether the observation of several peers is necessary. To answer this question, we fix the overall training set size to 4000 signatures per application and vary the number of peers selected as reference in each network (see Tab. 4.2 for details on the number of peers). Each experiment is repeated 10 times to collect average results. Fig. 4.8 shows the TPR obtained considering a reference set of *one*, *two* or *all* peers for each network in the testbed. Results show that the increase of the number of peers only provides a very limited gain on the classification performance. From a practical perspective, this is a very desirable property: even a single trace is sufficient to build expressive signatures.

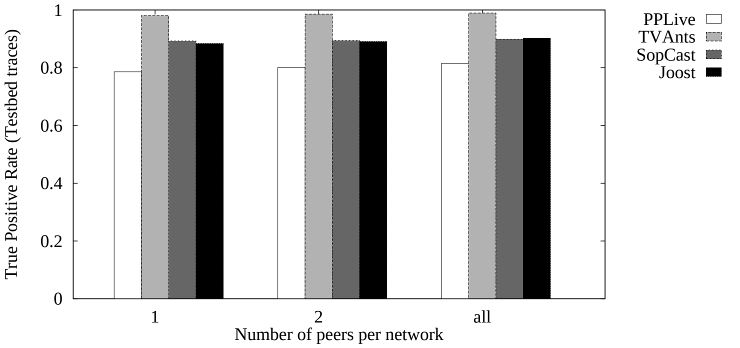


Figure 4.8: Impact of the training set diversity.

Impact of SVM Kernel and Binning Strategy

Since the core of our classification framework exploits SVM, all parameters that are susceptible of affecting its performance need to be investigated as well. Therefore, we focus on two main choices concerning the SVM design, that are: (i) the *kernel* function and (ii) the *binning* strategy.

The SVM literature is very rich of kernel functions, which are more or less indicated for different kinds of data. In our study we evaluate three well-known kernels: the general-purpose *Gaussian* kernel (K_G), the *Linear* kernel (K_L) and the Bhattacharyya kernel (K_B). The Gaussian and Linear kernels were already described and used in Chap. 3. The linear kernel (4.5) is simply the dot product of two feature vectors, while the Bhattacharyya kernel can be obtained by substituting each features with its square root [84]: due to the good performance exhibited early for the purpose of classification rejection, a natural question is whether the kernel function (4.6) can be helpful to better separate the different applications.

$$K_G(\underline{x}_i, \underline{x}_j) = e^{-\gamma||\underline{x}_i - \underline{x}_y||^2} \quad (4.4)$$

$$K_L(\underline{x}_i, \underline{x}_j) = \underline{x}_i \cdot \underline{x}_j \quad (4.5)$$

$$K_B(\underline{x}_i, \underline{x}_j) = \sqrt{\underline{x}_i \cdot \underline{x}_j} \quad (4.6)$$

As far as *bin distribution* is concerned, we use either $B_{exp} = B_n + 1 = 9$ exponential-width bins (base 2), or $B_{fix} = 255$ constant-width bins (1-packet steps); represented by Exp. and Const. in Tab. 4.7. Both spans over the $[0 : 255]$ packets range. Recall that the number of bins impacts both memory requirement and computational complexity, so that exponential binning should be preferred in case of comparable classification performance.

Results are shown in Tab. 4.7, which reports classification results in terms of the TPR of P2P-TV applications, and reports also the number of Support Vectors (SV) of the trained model. This latter number is a measure of the classification computational cost, since the number of operations that has to be performed to classify each signature grows linearly with the number of SVs. The cost of the training phase is not considered,

		True Positive Rate (TPR)				Support Vectors (SV)				
Bins	Kernel	PPLive	TVants	SopCast	Joost	PPLive	TVants	SopCast	Joost	Total
Exp.	K_G	**81.66**	98.51	**89.62**	**89.47**	**1015**	**106**	**845**	**415**	**2381**
	K_B	77.73	98.52	88.58	87.99	1759	110	1185	798	3852
	K_L	73.44	**98.54**	88.55	87.42	2062	219	1348	956	4585
Const.	K_G	**67.12**	**97.86**	**89.76**	**69.66**	**853**	**81**	**654**	**635**	**2223**
	K_B	65.27	97.14	89.58	68.27	1215	113	902	755	2985
	K_L	64.90	97.70	89.45	68.88	1382	316	911	1091	3700

Table 4.7: Classification performance and cost for different binning strategies, SVM kernels. In bold the best results.

since it is an offline operation rarely performed. The selected kernel has impact on the computational cost as well, with the linear kernel being light-weighted, and the Gaussian model being the most expensive. The Bhattacharyya kernel is in between the other two.

Tab. 4.7 collects results, bold fonts highlight the best choices. Given a binning strategy, the Gaussian kernel yields consistently better results for both TPR and number of SVs. It must be noted that the number of SV required for the Gaussian kernel are sensible less than the one required for the Linear kernel. We see a more important decrease in the performance of the constant binning, where the TPR for PPLive and Joost falls below the 70%. This is mostly due to the rejection criterion, which wrongly identify as unknown a conspicuous number of signatures. In fact the Bhattacharyya distance is less effective with this longer signatures which contains a lot of zero values which result in bigger distances to the class center. Results obtained with the Bhattacharyya kernel are almost equal to the linear kernel, with the advantage that the number of SV is smaller. Finally it must be noted that TVAnts requires a very small number of SV to obtain very good performance, irrespectively of the binning and kernel choice. In contrast PPLive which has shown to be difficult to classify, requires a number of SV that is ten times the same number for TVAnts for its best choice of binning and kernel.

With respect to the bin distribution choice, the use of exponential binning reduces the memory consumption and the number of operations to be performed by B_{fix}/B_{exp}, i.e., almost a factor of 30. For example, assuming 1 GBytes of RAM, $B_{exp} = 9$ exponential bins would allow to compute about $15M$ end-points considering 64bit floating point notation. With the same amount of memory, using B_{fix} linearly spaced bins allows to track roughly $0.5M$ end-points. Considering CPU time, a server equipped with an Intel Xeon E5345 clocked at 2.33GHz reaches 3000 classifications per second using exponentially distributed bins. Given that a signature is produced every $\Delta T = 5\,\mathrm{s}$, about 15000 end-points could be classified in real-time even by our non optimized code. Considering linearly distributed bins, only 126 classifications per second are computed, allowing to classify no more than 630 end-points.

The previous analysis reenforces the selection made until the moment, i.e. the exponential binning strategy and Gaussian kernel.

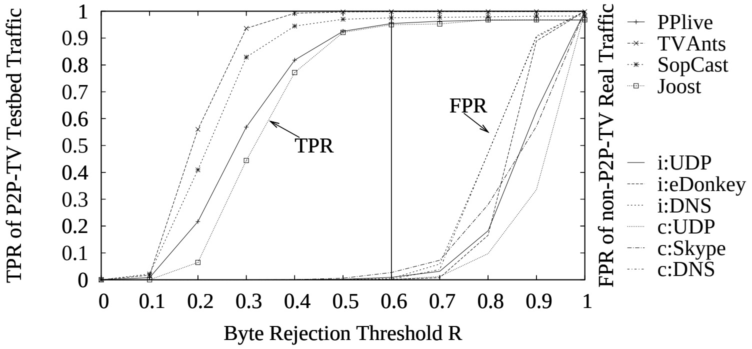


Figure 4.9: TPR and FPR as a function of the rejection threshold R evaluated on bytes feature.

4.4.3 Improving the Accuracy: Extending the Signature

In this section we augment the Abacus signature to include not only the number of packets received by each period $\underline{n}$, but also the number of received bytes. Following the same procedure as before with packets, we consider a ΔT time interval in which the endpoint $\mathcal{P}$ receives $b_1, \ldots, b_K$ bytes from K peers. $B_b + 1$ exponential-width classes are identified, according to the number of bytes received from $\mathcal{P}$, and counting the occurrences of each class in B_i. The byte-wise signature $\underline{b} = (b_0, \ldots, b_{B_b})$ is then obtained by normalizing the count B_i over the total number of contacted peers K:

$$b_i = \frac{B_i}{K}$$

The tuple $\underline{b}$ is a pmf, whose component b_i can be interpreted as the probability that an arbitrary peer sends between $(2^{i-1}, 2^i]$ bytes to $\mathcal{P}$. For byte-wise signatures, we set the number of bins to $B_b = 14$.

We define the application signature by concatenating the packet-wise $\underline{n}$ and byte-wise $\underline{b}$ signatures in a single vector $\underline{a} = (\underline{n}, \underline{b})$. Since the extended signature $\underline{a} = (\underline{n}, \underline{b})$ is composed of two parts, we can define two rejection thresholds, considering $\underline{n}$ or $\underline{b}$ only. We therefore report in Fig. 4.9 the TPR and FPR evaluated on as a function of the rejection threshold R applied to byte signatures. Contrasting Fig. 4.9 with the packet-signature results shown early in Fig. 4.5, we observe that the bytes signature is somehow more robust to the choice of R, suggesting to apply the rejection criterion to $\underline{b}$ only. In fact we can also safely adopt a bigger threshold $R = 0.6$, which further limits the percentage of false alarms raised.

We then perform the classification based on the extended signature $\underline{a}$ with a byte-wise rejection criterion and a rejection threshold $R = 0.6$. Results reported in Tab. 4.8 are gathered for $\Delta T = 5$s, with 4000 signatures extracted at random to build the training set. Compared to previous results of Tab. 4.6, the extended signature leads to significant performance improvement, so that TPR is now higher than 95%, and misclassification probability is reduced to few percentage points.

	Signatures: Confusion Matrix				
	PPLive	TVAnts	SopCast	Joost	Unk
PPLive	**95.42**	0.22	1.86	0.36	2.14
TVAnts	0.06	**99.84**	0.10	0.00	0.00
SopCast	0.98	0.15	**97.55**	0.03	1.29
Joost	0.21	0.01	0.01	**94.97**	4.80

Table 4.8: Extended Abacus signatures: Confusion Matrix of P2P-TV applications

	Signatures			Packets			Bytes			Peer	
	TP	Mis	Unk	TP	Mis	Unk	TP	Mis	Unk	TP	Unk (n)
PPLive	95.42	2.44	2.14	98.11	1.60	0.29	98.32	1.54	0.14	100.0	0.0 (0)
TVAnts	99.84	0.16	0.00	99.77	0.23	0.00	99.82	0.17	0.01	100.0	0.0 (0)
SopCast	97.55	1.17	1.29	99.18	0.78	0.04	98.96	0.98	0.06	97.06	2.94 (1)
Joost	94.97	0.23	4.80	99.50	0.25	0.25	99.62	0.23	0.15	93.33	6.67 (2)

Table 4.9: Extended Abacus signatures: per Signature, Packets, Bytes and End-Point classification results.

To better appreciate results, in Tab. 4.9 reports performance considering correctly classified *packets*, *bytes* and *peers*. Packet-wise and byte-wise performance can be directly gathered by taking into account the number of packets and bytes carried by each signature; the peer classification is instead evaluated considering a *majority* criterion, so that a peer is classified as running application X if the majority of time such peer samples have been classified as X.

Tab. 4.9 reports the percentage of correct classification (TP), of misclassification (Mis, corresponding to the sum by rows of non diagonal values in the confusion matrix) and rejection (Unk) for all the above metrics. Notice that FN=Mis+Unk. Interestingly, performance improves when the number of correctly classified *packets* and *bytes* is considered, suggesting that misclassification occurs when signatures carry few data, e.g., when the application is possibly malfunctioning. In case of *peer* classification, reliability of end-points identification increases as well. Only 3 hosts are classified as not running any P2P-TV application, and notably there is no misclassification. Investigating further, we found that rejected cases correspond to peers that received a small amount of traffic, and, thus, possibly were not playing any video.

So far, we have seen that extending the abacus signatures improves the classification performance of P2P-TV application: we now have to assess the impact of this extension on the effectiveness of the rejection criterion. We again consider real traffic collected from operational networks, considering only the traffic portion for which possible false positives may be triggered (i.e., the portion of the samples for which $K > 2$). Tab. 4.10 reports results referring to the extended signatures, showing the false positive rate (FPR) and the breakdown of false alarms between the different P2P-TV applications. First, notice that the number of false alarms is very limited, being only 2.7% considering end-points for which $K > 2$. If all UDP traffic is considered irrespectively of K, FPR drops to less than 0.1%. This negligible number of false detections confirms the

			FP Confusion Matrix			
Network	Traffic	FPR	PPLive	TVAnts	SopCast	Joost
CAMPUS	UDP	**2.70**	0.57	1.00	1.13	-
	Skype	0.04	-	0.03	0.01	-
	DNS	0.17	0.02	0.10	0.05	-
ISP	UDP	**0.90**	0.61	0.14	0.15	-
	eDonkey	0.09	0.02	0.04	0.03	-
	DNS	0.44	0.03	0.33	0.08	-

Table 4.10: Non P2P-TV Traffic in Campus and ISP traces: False Positive Ratio (FPR) and FP Confusion Matrix.

reliability of the classification engine. Moreover, false positive rate is low for individual applications too: indeed, it is very rare that eDonkey or Skype traffic is confused with any P2P-TV application (0.09% and 0.04% of false positives).

4.5 Signatures Portability

We now evaluate *network portability* of Abacus signatures. The objective is to answer the question: how generic is a training performed considering traces collected in a network? Our testbed dataset is different enough to see what happens when, for example, the classifier is trained considering a trace collected in a University Campus network, and then used in a totally different network, like a ADSL scenario. Moreover, both the access types and the channel popularity could impact the accuracy of the training set, which we deal with in the following. Finally, we also test how robust the classifier is in presence of high packet loss or limited bandwidth.

For the sake of simplicity, we consider only packet-wise Abacus signatures and testbed traces, and no longer apply the rejection criterion. Results are summarized in Tab. 4.11: the first column reports the experiment label, the second column (Train) states which training set was used, while the third column (Test) reports the dataset used for the validation of the classification process. TPR for each application are reported in the subsequent columns. To ease the comparison, the first row (labeled *Ref*) reports the baseline results: notice that TPR is slightly higher with respect to Tab. 4.6, since we do not apply the rejection criterion.

Portability across Different Network Sites (NS)

In the first scenario, we consider traffic captured from PCs running at different institutions, i.e., in different Countries, networks, etc. (see Tab. 4.2). We start by considering peers that are all placed in corporate or campus networks, with high-bandwidth connections to the Internet. There are 7 of such sites. For each application, we select 4 sites, and used traffic collected there for the training. Then, traces collected in the

	Train	Test	PPLive	TVAnts	SopCast	Joost
Ref	ALL	ALL	**84.84**	**98.51**	**92.63**	**91.50**
NS	4/7	3/7	78.90	97.61	90.30	88.61
AT	ADSL	ADSL	83.48	97.86	95.61	91.36
	ADSL	HB	79.63	93.73	87.30	90.61
	HB	ADSL	58.28	98.15	93.70	81.55
CP	POP	POP	95.88	-	-	-
	POP	UNP	48.59	-	-	-
	UNP	POP	94.79	-	-	-
TI	2008	2006	18.81	98.44	51.06	-
EI	HB	Bw	91.14	76.80	75.76	-
	HB	Delay	88.19	84.62	77.80	-
	HB	Loss	75.22	91.77	84.31	-

Table 4.11: Signature portability: TPR evaluation.

remaining 3 networks are classified to evaluate TPR. To gather robust results, we consider every possible combination $\binom{7}{4} = 35$ of training and validation subsets. For each combination, 3 tests are performed with different random training samples.

Results are reported in the raw labeled as *NS* in Tab. 4.11, which shows that signatures are network-portable under homogeneous settings: indeed, the largest performance drop is 5%, which corresponds to the PPLive case.

Portability across Different Access Technologies (AT)

We now test to what extent signatures are portable across different access technologies, e.g., ADSL versus High Bandwidth (HB) access. As noted in [75], nodes with high-bandwidth access can act as "amplifiers", providing content to possibly several peers; conversely, ADSL peers may only act as "forwarder" due to the limited uplink capacity. Despite we consider only the downlink traffic, different behaviors can impact the Abacus signatures, e.g., due to a different fraction of signaling packets a peer receives. For example, an amplifier peer can receive many small sized acknowledgments, while a low-capacity peer mainly receives large packets containing video data. We therefore split the testbed dataset into two parts: the first contains traces collected from all High Bandwidth PCs, while the second contains ADSL PCs. Three tests are performed: (i) classifying ADSL traces using ADSL training set, (ii) classifying ADSL traces using the HB training set and (iii) classifying HB traces using ADSL training set. Each test has been repeated 10 times, and average results are reported.

Results are reported in rows labeled *AT* in Tab. 4.11. Overall, Abacus signatures confirm their portability even across different access networks: for TVAnts, SopCast and Joost, results are modestly impacted by train/test combination (being 8% of reduced TPR the worst case). In case of PPLive, the TPR drops to 58% when HB training is used to classify ADSL traffic. This is likely due to the fact that PPLive is very aggressive in exploiting HB peers upload capacity, so that the number of peers sending acknowledgments shifts the signature toward low bins, i.e., few acknowledgment packets are received from a given peer. ADSL peers, on the contrary, contribute with little upload bandwidth, so that the incoming traffic is mainly due to video chunks received as trains of packets, i.e., groups of large data packets that are received from contributing peers.

Portability across Channel Popularity (CP)

We now consider what is the impact of channels with different popularity. Channel popularity indeed may significantly influence the P2P-TV application behavior. For instance, if popular channels are considered a large number of peers are available, while for unpopular channel few peers can be used to exchange the video content. We performed a second experiment considering a very popular (POP) channel using PPLive. We selected PPLive since it is the P2P-TV application for which Abacus showed the worst performance so far. The total number of peers observed during this experiment was larger than 200000, while in the previous dataset less than 56000 peers were observed. We refer to this dataset as a unpopular channel (UNP). As before, we evaluate the portability over all combination of train/test sets, repeating the experiments 10 times. Results are reported in the rows labeled *CP* in Tab. 4.11. Few considerations hold: first, PPLive classification performance improves when it comes to the classification of popular channels (i.e., TPR in POP/POP and UNP/POP cases is about 95% versus the about 85% of the UNP/UNP case used as reference). Nonetheless, we observe that the classification of UNP dataset when training has been done considering the POP dataset leads to poor performance (TPR drops to less than 50%). This partly limits the portability across channels. A simple solution consists in building a training set containing a mixture of signatures from both traces, which raises the TPR again to about 85%. This result suggests that channel popularity should be explicitly taken into account when building the training set, by including samples that are representative of different channel popularity.

Portability over Time (TI)

We now focus on the signatures portability over different periods of time. From a practical point of view, this allows to know how often classifier should be retrained. We resort to the traffic traces used in [85], which authors kindly made available to the scientific community. Traces of [85] were collected in July 2006 during the Fifa World Cup: the study focused on the same applications we examined in this article, with the exception of Joost which was not available at that time. Overall, the time-portability measurements account for 14 hours of video, $14M$ packets and 2.3 Gbytes of data. We classify this old dataset using the Abacus classifier trained with the dataset collected in 2008 (same training set of Sec. 4.4). Notice that the network environment was also different, so that we are jointly evaluating time and network portability.

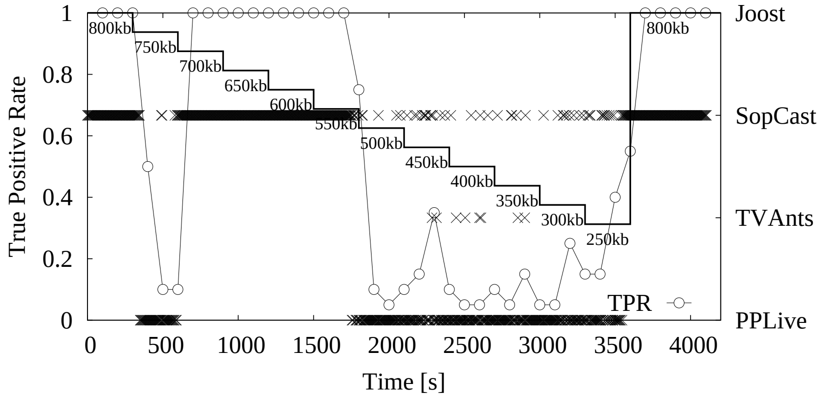


Figure 4.10: Portability over Emulated Impairment: example of temporal evolution of SopCast classification for decreasing bottleneck bandwidth.

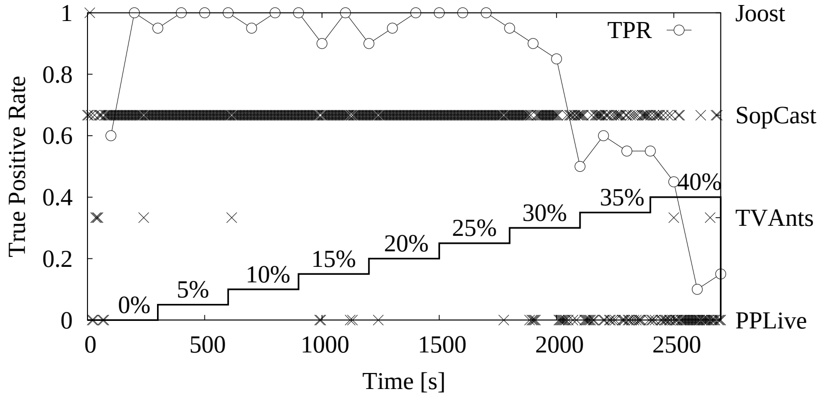


Figure 4.11: Portability over Emulated Impairment: example of temporal evolution of SopCast classification for increasing packet loss rate.

Results are reported in the row labeled *TI* of Tab. 4.11, which shows that TVAnts is correctly classified, SopCast has a TPR of 51%, while PPLive is almost completely misclassified. This suggests that applications changed drastically their behavior from July 2006 to March 2008. Notice that TVAnts was at version 1.0.58 in [85] and it is now at 1.0.59, which suggests that little changes have been implemented. On the other hand, PPLive moved from version 1.1.0 to version 1.9.15 and SopCast from 0.9 to 3.0.3, hinting to clearly more drastic changes. Thus, as long as applications do not change, the Abacus signatures are extremely portable across time –even across years– as we see in the case of TVAnts. On the other hand, if an application implements new algorithms which result in new behavior, then Abacus should undergo a new training phase.

Portability over Emulated Impairments (EI)

As a final case, we consider whether Abacus signatures are portable across different network conditions. We consider the traces gathered in an active testbed [86], where changing network conditions were artificially enforced. In particular, in these experiments, a Linux router was used to emulate some network conditions: bandwidth, delay and packet losses were imposed on the downlink path to the PC running the P2P-TV application.We refer to [86] for a complete description of the testbed: we only point out that impairments range from mild to very tough conditions (e.g., 200 Kbps of available downlink bandwidth, delay up to 2 s and packet losses up to 40%).

Traces gathered in this testbed are classified considering the HB training set, and results are reported in the last lines of Tab. 4.11 labeled *EI*. Even in these extreme conditions, Abacus still exhibits very high TPR, which can still exceed 90% for some applications, with a worst case of about 75%. Reported results are averaged over all the time varying conditions, including very distorted scenarios. Classification results are differently impaired by different network conditions. For example, PPLive is mostly affected by loss increase, while TVAnts classification results are more sensitive to bandwidth change. SopCast results are mostly affected by bandwidth and delay changes.

Interestingly better results are achieved considering PPLive classification in the case of bandwidth limitations. While this seems counter intuitive, it can be explained considering that most False Negatives obtained from other applications are actually misclassified as PPLive. This suggests that PPLive signatures are more variable and spread out, avoiding FN classification for PPLive but possibly causing more FP classification for other applications.

As an example, we report on the time evolution of two different experiments of SopCast classification, considering a scenario in which the available bandwidth is decreasing (Fig. 4.10), or the packet loss rate is increasing (Fig. 4.11). Every 5 minutes network conditions are artificially worsened by either reducing the available bandwidth by 50 Kbps, or by increasing the packet loss rate by 5%. The resulting impairment profile is reported in the picture.

Fig. 4.11 plots individual classification decisions, taken each $\Delta T = 5$s: these are represented with crosses, referring to the right y-axis, and allow to see when and how the application has been eventually misclassified. The picture also reports the True Positive Rate, evaluated over 20 consecutive signatures (i.e., 100 seconds), represented as a continuous dotted line referring to the left y-axis. Considering Fig. 4.10, which refers to bandwidth limited scenario, it can be seen that as soon as the bottleneck bandwidth kicks in, SopCast is misclassified as PPLive during a brief period, possibly hinting to a sudden reaction of the application to the anomalous conditions. Then, SopCast is correctly classified until the available bandwidth drops too low: afterward, SopCast TPR drops quickly, being most of the time misclassified as PPLive and seldom with TVAnts. At the end of the experiment, when the bottleneck bandwidth is removed, SopCast is again correctly classified. Similar considerations hold for the loss scenario depicted in Fig. 4.11, in which samples are misclassified only when loss rate exceeds 30%.

4.6 Conclusions

In this chapter we have applied SVM for multi-class classification purpose, in particular we consider the problem of classifying P2P-TV applications. We proposed Abacus, a novel behavioral approach for fine-grained classification of P2P-TV applications. Our methodology relies only on the simple count of packets and bytes exchanged amongst peers during small time-windows. The rationale is that these two counts are representative of the different operation of each application, such as signaling activities, chunk size, etc. We applied our framework to four well known applications: PPLive, TVAnts, Sopcast and Joost.

We have also addressed the problem of dealing with non P2P-TV traffic by means of defining a rejection threshold: in short, if the signature is far enough from the mean signature in the training set of the corresponding application, then the result is rejected and the signature is classified as “unknown”.

We evaluated our classification framework by means of a large experimental campaign that includes P2P-TV traces actively collected in a large scale testbed (in the context of the NAPA-WINE European project [9]) and real traces that do no contain P2P-TV traffic. This mix approach allowed us to evaluate the capacity of our classification engine to correctly identify the P2P-TV application that generates the traffic. Furthermore, it also allowed us to identify if the traffic does not correspond to any of the considered P2P-TV applications.

If packet based signatures are considered, we found out that in mean we can accurately identify signatures in almost 90% of the cases with a worst case of 82%. Furthermore, if bytes are included on the signatures, our classification engine correctly classifies about 95% of packets, bytes and peers in the worst case. Concerning the non P2P-TV traffic, our results indicate that the classification engine raise very few false alarms, well below 1% in the worst case. Such astonishing performance is the result, on the one hand, of the discriminative power of SVM, and, on the other hand, of the descriptive expressiveness of Abacus signatures.

A large set of experiments have been carried over to assess Abacus parameter sensitivity and portability. We have analyzed different time-windows over which we calculate the Abacus signature, finding that the accuracy on the classification performance is only mildly degraded if longer time scales are considered. With respect to the SVM parameters, the choice of using a Gaussian kernel with exponential binning has proved to be the best combination in terms of accuracy and complexity. Results over the training set and diversity have shown that a very large training set is not required and that information gathered from a single peer in each network is enough to build expressive signatures and so to obtain very good performance.

As far as Abacus signature portability is concerned, we have analyzed very different scenarios including network heterogeneity, access technology, channel popularity, time and even distortions in network bandwidth or losses. In most of the cases, Abacus seems to be very robust in face to these changes and classification performance is maintained or limitedly worsened. Probably the most important conclusion of this analysis is that channel popularity must be taken into account in the training phase of the model.

To sum up, training the Abacus classifier is simple, as signatures can be generated automatically using a very small number of traces. In terms of both memory requirements and computational complexity, Abacus is also very lightweight. Abacus is therefore apt to be deployed in real network environments. Finally, since the information required by Abacus simply relies on packet and byte counts, it is worth investigating how to integrate it in NetFlow monitors which are commonly deployed in operative networks. However, in this case, a more fine rejection criterion should be defined.

Part II

Multiple Access Mechanisms in Ad-hoc Networks

Chapter 5

Modeling CSMA/CA

There are no doubts about today's importance of wireless networks. They are present almost everywhere in everyday life: cellular networks, public Internet access (Wireless LAN have increasingly became the edge network of choice), residential WiFi networks and a large etcetera. In particular, Mobile Ad-hoc NETworks (MANETs) are infrastructureless wireless networks able to provide temporary and instantaneous solutions where other infrastructured wireless networks (e.g. cellular) are expensive or even infeasible to deploy (for instance rural or disaster areas).

More in detail, MANETs are very attractive since they are easy to deploy and offer great flexibility. Furthermore, due to their inherently distributed nature, they are more robust against network elements failure. In these networks, nodes act both as transmitter and receiver and they operate in a self-organized and adaptive manner. In addition, to further increase the coverage of the network, multihop routing is used; i.e. nodes between origin and destination may be used as relays. This means that each node can be at the same time a terminal (origin or destination of some traffic) and a router (relay for traffic between other origin/destination pair). Much research efforts (e.g. routing, security, quality of service, etc) have been dedicated to this area specially to enable the adoption of this kind of networks in wide-scale.

A key issue in such networks is the design of a Medium Access Control (MAC) mechanism, which is however not an easy task. When several neighboring transmissions access the medium simultaneously they might jam each other and a mechanism to decide which transmissions can access the medium at a given time is then required. Due to the multihop characteristic of ad-hoc networks, a packet needs to be retransmitted until reaching its final destination. These retransmissions enlarge the interference created by each packet and the loss probability, thus increasing the need of a MAC mechanism.

If too many nodes are allowed to transmit at the same time by the MAC mechanism, the interference created by them will prevent an intended receiver from correctly receiving the signal. If on the contrary too few nodes are allowed to transmit simultaneously, resources (such as bandwidth) are wasted and the overall performance may be degraded. The essential characteristics of a well designed access mechanism will be then a large number of simultaneous *successful* transmissions, and at the same time, equal access opportunity for all nodes. However, since the MAC mechanism is run locally by each node, which has only local information of the network state, it is difficult to globally optimize the number of successful simultaneous transmissions based on their individual decisions.

The medium can be shared according to very different strategies. For example, a controlled access strategy such as TDMA (Time Division Multiple Access) could be used. In this case, time is divided into slots and the sharing is organized in a predetermined basis, i.e. a collision free schedule is computed and informed to all nodes. This strategy may be quite inefficient since the schedule must be recomputed each time something changes. Alternative strategies are random access based mechanisms such as ALOHA or CSMA (Carrier Sense Multiple Acess) which are generally admitted to be more adequate given the mobile and open characteristics of ad-hoc networks. In ALOHA, at each time slot, each node access the channel with a fixed probability appropriately chosen to avoid a large number of collisions. The main feature of CSMA is to sense the medium before transmitting: a node intending to transmit first senses the medium; if the latter is idle, the node transmits; else it waits until its idle again. Both, ALOHA and CSMA have the useful property of being fully decentralized. There exists other strategies such as CDMA (Code Division Multiple Access) or (O)FDMA ((Orthogonal) Frequency Division Multiple Access); however we will not go into details on them since our main study objet will be CSMA. Good overviews of MAC mechanisms can be found in [87, 88].

CSMA/CA (CSMA with Collision Avoidance) is perhaps the most widely used MAC mechanism for wireless networks and its popularity is mainly due to its simplicity. The collision avoidance (CA) is required to avoid that once the medium is sensed idle, several nodes start transmitting at the same time, creating collisions. The CA consist in the introduction of a random time between the time that the medium is sensed idle and the beginning of the transmission. This random time varies in a interval that increases with the number of consecutive collisions. However, collisions may occur anyway. A Ready-To-Send/Clear-To-Send (RTS/CTS) handshake, also known as “virtual sensing”, can be used in addition to CSMA/CA to avoid the well known “hidden terminal” problem. This problem arises when two transmitters, that are out of range of each other, are allowed to transmit to the same receiver at the same time creating a collision. This collision can be avoided by using an RTS/CTS exchange previous to data transmission: the receiver will authorize the communication (if it is possible) and at the same time will inform its potential interferers of the ongoing communication. We will not analyze here some of the already known weaknesses of this protocol, such as the access unfairness (which is usually referred to as “starvation phenomenon”[89]) or the lack of performance guarantees for the accepted transmissions, since the goal of this chapter is to analyze how to define an accurate model for the CSMA/CA protocol as it is. The problem of defining a MAC mechanism with performance guarantees will be considered in the next chapter.

More precisely, the objective of this chapter is twofold. Firstly, we will try to identify and classify the different models used so far to analyze the performance of the CSMA/CA protocol, highlighting their benefits and drawbacks. To achieve this we will first describe, with more details, the main features of CSMA/CA which are essential to the model definition, leaving aside all implementation-specific details. We will see that most of the existing models do not include channel variations due for example to shadowing/fading effects. Moreover, analytical formulae for performance indicators, such as the spatial reuse or the throughput, can be obtained only in simple cases such as a regular line topology. Our second objective is then to present two different shadowing/fading aware CSMA models and to show that these models present significantly different performance results than the previous one, revealing the need of taking into account these effects. We will also discuss why previous model cannot be extended and the difficulties to obtain analytical results.

It must be noted that even in the simplest model where the signal power emitted by a node decays exponentially with the distance in an isotropic way, the space distribution of nodes plays a key role in the overall performance of the network since it determines the SINR (Signal to Interference Noise Ratio) at each receiver and hence its rate. Based on this observation, we will concentrate on models that take into account the geometry of the network. This assumption excludes for instance the Markovian models for IEEE 802.11 DCF presented in [90] (or the fixed point analysis of [91]) and for the slotted version of IEEE 802.15.4 presented in [92]. In these works a per user Markov model is used to capture the state of each user at each moment and different ways of coupling them are derived according to the specific characteristics of the protocol under analysis.

5.1 CSMA/CA Basics

In this section we will describe the CSMA/CA operation, emphasizing those aspects that are crucial at the time of defining a model to represent it. There exist different flavours of the CSMA/CA protocol depending, for instance, if the time is assumed to be divided in slots or not. For example, in IEEE 802.11 [93] a non-slotted CSMA/CA is used whereas a slotted one is used in beacon-enabled IEEE 802.15.4.

The behaviour of CSMA/CA in the non-slotted case can be schematically explained as follows. Each link (or transmission) has a backoff timer which is chosen according to a given backoff distribution. This timer decreases with time and when it reaches zero, the link becomes active provided that the channel is sensed idle (more details on this aspect will be discussed further on). In this case, the backoff timer of the neighboring connections is frozen, i.e. the backoff value decreases only when no activity is detected in the exclusion domain (see next section for a more precise definition of exclusion domain). When the transmission is finished a new backoff value is drawn from the backoff distribution. An example of this dynamic is shown in Fig. 5.1 for two neighboring nodes.

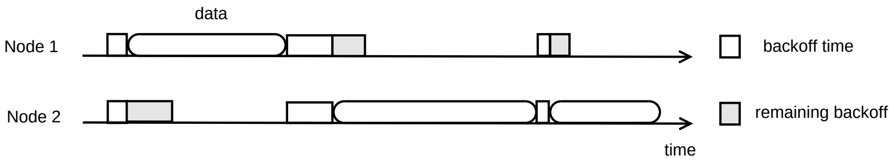


Figure 5.1: Illustration of the backoff dynamic: when the backoff timer of Node 1 reaches zero it transmits and Node 2 frozes its backoff timer; at the end of the transmission Node 1 gets a new backoff timer and Node 2 continues to decrease its remaining backoff timer until it reaches zero.

When time is divided in slots transmissions are synchronized. At the beginning of the time slot all nodes draw a new backoff timer and the decision of which transmission is activated is the same as in the non-slotted case. The difference lies in the fact that all transmissions finish at the same time.

In both cases, the channel is sensed in two phases (the first one being a physical sensing whereas the second one is logical):

1. Clear Channel Assessment (CCA): is the procedure performed by a node intending to access the channel to identify the channel state (idle/busy).
2. RTS/CTS Handshake: followed by the data transmission and finished by an ACK (acknowledge packet), this handshake is used to exchange information about the duration of the transmission avoiding the so-called “hidden terminal” problem.

More details on these two phases are describe in the following sections.

Clear Channel Assessment

In CSMA/CA networks a node intending to transmit first senses the medium; if it is idle, the node transmits; else it backs off and tries again after a random time. In order to sense the status of the channel the so-called Clear Channel Assessment (CCA) procedure is used by network nodes. CCA depends on the specific MAC protocol used and the terminal settings. For the most widely used CSMA/CA protocols (e.g. IEEE 802.11 DCF or IEEE 802.15.4) CCA is performed according to one of the following modes [93]:

- Mode 1 (Energy Detection): CCA reports a busy medium if the detected energy (measured power level) is above a user-selectable threshold.
- Mode 2 (Carrier Detection): CCA reports a busy medium if at least one signal is detected having the modulation characteristics of the considered wireless network.
- Mode 3 (Energy and Carrier Detection): CCA reports a busy medium if at least one signal is detected having the desired modulation characteristics and the total power is larger than the given threshold.

All modes intend to represent the fact that if there is some activity on the channel, the transmission must be postponed. The first mode seems more adequate for narrowband systems, whereas the second one is for wideband systems [94]. As we will see, and as far as modeling is concerned, an important difference between these modes is that Carrier Detection can be modeled as a pairwise relation between nodes, whereas the Energy Detection mode requires to take into account the additive interference created by all the ongoing transmissions. In the next chapter we will analyze these two modes in more detail.

RTS/CTS Handshake

As mentioned in the introduction of the chapter, an RTS/CTS handshake can be used in addition to the CCA to avoid the "hidden terminal" problem. When a node intends to transmit, it first sends an RTS message to its receiver which in turns replies with an CTS message if it is currently not busy, i.e. it can successfully receive the data. Additionally, the RTS and CTS messages carry the total duration of the tagged transmission in the Network Allocation Vector (NAV). In this way, all "overhearing" nodes (nodes in the transmission range) will be aware of the duration of the ongoing transmission. These nodes will be silenced during the tagged transmission, meaning that they refrain themselves from transmitting or receiving. In this way nodes that are close to each other do not transmit at the same time. The set of nodes silenced by an active transmission is referred to as the *exclusion domain* (ED).

This handshake solves the hidden terminal problem but other problematic situations can appear. For instance, a node that receives a RTS message is refrained from transmitting even if its receiver is outside the interference region of the ongoing transmission. This problem is known as the "exposed terminal". A different problem is generated when a potential receiver node decodes a CTS message and as a consequence cannot receive even if the activation of its transmitter does not interfere the ongoing transmission.

Other problematic situations can be identified. For example a node could be silenced by the RTS message of a transmission that will finally not take place (the intended receiver will not answer with a CTS because it is already silenced by another communication). In this case, the virtual sensing (expressed by the NAV value) does not reflect the real state of the network. Another situation arises when a node, which is jammed by an active data transmission, cannot decode an RTS message and so it cannot update its NAV. This means that this node is not considering that another communication is taking place and when the jamming transmission is finished, it could start a new transmission creating a collision. Clearly these situations are not desirable since transmissions that could take place at the same time are instead rejected or accepted ones are interrupted. However these drawbacks are a direct consequence of the simplicity of the mechanism which is in turn its major advantage. Different solutions are suggested in the standard or were proposed later to solve, or at least to alleviate, the negative effects of these situations. However, as we mentioned before, it is not our intention to solve the drawbacks of the CSMA/CA protocol, but to analyze the definition of an accurate model for it in order to better understand its behaviour.

5.2 Traditional Models for CSMA/CA

In this section we will describe those models that have been proposed to analyse the CSMA/CA protocol. We do not pretend that the list of works discussed here is exhaustive. Instead, we will highlight the models that, in our opinion, are the most representative of the different approaches that can be used in this context.

Before going into details, we will introduce the notation used in this and the following sections. Given two nodes, h and k, the power received from h by k is:

$$P(h,k) = P_h L(h,k), \tag{5.1}$$

where P_h is the transmission power of node h, and $L(h,k)$ is the path loss from h to k which depends on the distance between nodes. Different path loss functions can be considered. For instance, if r is the distance between h and k, a simple case is to consider $L(h,k) = l(d(h,k))$ where $l(r) = Ar^{-\alpha}$ where $\alpha > 2$. Since this simplified model makes no sense for r close to 0, we will use instead $l(r) = A\max(r_0, r)^{-\alpha}$ where $r_0 > 0$. Both functions give similar results when $r > r_0$ or Ar is large.

Let T_i and R_i denote the transmitter and receiver of link i respectively. We will assume that the communication between T_i and R_i will be successful if the two following conditions are verified:

$$P(T_i, R_i) \geq P_0 \quad \text{and} \quad \text{SINR}_i = \frac{P(T_i, R_i)}{W + \sum\limits_{j \neq i} P(T_j, R_i)} \geq \gamma,$$

where W is the thermal noise and P_0 and γ are selectable thresholds.

The RTS/CTS packets, as well as the ACKs, are usually sent at rates lower than that of data packets, and so we can safely ignore the condition over the SINR. In this case the, the exclusion domain (ED) of link $i = [T_i, R_i]$ (i.e. links silenced by the activation of link i) may be defined as follows:

$$\begin{aligned} C(i) &= \{j : P(T_j, R_j) > P_0 \text{ or } P(T_j, T_i) > P_0 \text{ or } P(R_j, T_i) > P_0 \text{ or } P(R_j, R_i) > P_0\} \\ &= \{j : P(h,k) > P_0 \text{ for } \ h \in \{T_i, R_i\}, k \in \{T_j, R_j\}\}. \end{aligned} \tag{5.2}$$

Note that if the transmission power is constant and equal for all nodes ($P_h = P \ \forall h$), then the ED is symmetric in the sense that if $j \in C(i)$, then $i \in C(j)$. This means that two contender links cannot access the channel at the same time.

Note that the condition that the reception power from h in k is larger than P_0 is equivalent to verifying that the distance between them is smaller than a corresponding threshold. More precisely, if $r = d(h,k)$ is larger than r_0 and $l(r) = A\max(r_0, r)^{-\alpha}$, then:

$$P(h,k) > P_0 \Leftrightarrow r < \left(\frac{P_0}{PA}\right)^{-1/\alpha}. \tag{5.3}$$

This assumption implies that the ED of a given link is constituted by those nodes which distance to the transmitter or receiver of the tagged link is smaller than a given threshold. As we will see, this equivalence largely simplifies the problem and it will be the key to allow the calculation of analytical formulae for the CSMA/CA performance. In what follows, we will present the three considered models: packing formalism, truncation of a reversible process, and Matérn like models.

5.2.1 Packing Formalism and Gibbs Distribution

This section is devoted to the presentation of the models introduced in [21]. Following the author's notation, the ED of a tagged link is defined as the union of two discs:

(i) the set of nodes within the RXRange (receiving range) of the sender or the receiver of the tagged link and,

(ii) the set of nodes within the CSRange (carrier sensing range) of the tagged sender.

This definition of the ED is meant to represent the CSMA/CA RTS/CTS handshake: all nodes, different from the tagged receiver, which can decode a RTS (i.e. a node within the RXRange of the tagged transmitter) are silenced. Similarly, all nodes that can decode a CTS (i.e. within the RXRange of the tagged receiver) are also silenced. In addition, it is assumed that CSMA/CA silences all nodes within the CSRange of the tagged sender (physical carrier sense). It must be noted that the CSRange aims to represent the CCA. More details on this assumption will be discussed in Sec. 5.2.4. The ED results then in the union of two (usually intersecting) discs, one centered at the transmitter and the other at the receiver of the tagged link. This means that the contenders of an active link (the nodes in its ED) are its nearest neighbors.

In what follows, we will usually refer to a *line topology* where nodes are equally spaced (by a distance d, assumed to be 1 space unit) along a straight line. Each node can transmit or receive and it communicates with its nearest neighbors. In this case, a minimum distance between two active transmissions is then required. Let l be 1 plus this minimum distance, then it can be an active link every l space units. An example of this situation is illustrated in Fig. 5.2.

To simplify the analysis but also to analyze the protocol under "extreme" conditions, saturated traffic is assumed, meaning that each link always has data to send. Under this assumption, a very important performance metric in the wireless ad-hoc setting is the number of simultaneous transmission that can be scheduled by the protocol. For this, the spatial reuse – defined as the mean proportion of links which are active at a typical time slot – is generally the performance indicator of reference.

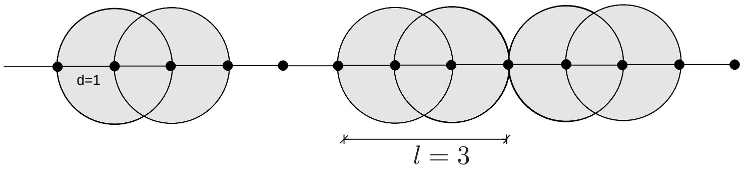


Figure 5.2: Illustration of the packing analogy.

Slotted Case: Packing Formalism

As we mentioned above, when the time is divided in slots, the decision of which links can be activated in each time slot is decided at the beginning of the slot. At its turn, each link will transmit if the channel is sensed idle. If the line topology is assumed, defining a transmission schedule (set of active links) is equivalent to finding a packing of the line with as many non-overlapping intervals of length l as possible. This means that the spatial reuse σ only depends on l (i.e. the "space" occupied by an active link).

More precisely, using this packing analogy, authors of [95] show that the spatial reuse achieved with L equidistant nodes when $L \to \infty$ is:

$$\sigma = \exp[-2F(1)] \int_0^1 \exp(2F(u))du \quad \text{where} \quad F(u) = \int_0^u \frac{1-y^{l-1}}{1-y} dy \tag{5.4}$$

In particular if $d = \text{RXRange} = \text{CSRange} = 1$, each active link silences its two left/right neighbors and the minimum distance between two active links is $l = 3$. See Fig. 5.2 for an illustration of this situation. In this case, $\sigma = 0.275$ whereas the maximal spatial reuse is 1/3 (one out of three links is active).

Non-Slotted Case: Gibbs distribution

In this case data transmissions can start and finish at any given time. Thus, the duration of the transmissions is supposed to be random. In particular, it is assumed that they are exponentially distributed. Concerning the backoff distribution, in the real protocol it is defined as uniform discrete in an interval depending on the value of the contention window. However, as in most works in the literature, the authors of [21] assume that the backoff timers are also exponentially distributed. A continuous backoff distribution implies that the probability of a collision occurrence is null, which constitutes an ideal case.

Under all the mentioned hypothesis (attenuation depending only on distance, exponential distribution of the backoff and transmission duration, and saturated traffic condition), a continuous time Markov chain can be used to model the dynamic of CSMA/CA. For the line topology, a transmission pattern is a vector $x \in E = \{0, 1\}^L$ that specifies which of the L links are active (if i is active, then $x_i = 1$). A transmission pattern must verify then the following condition:

$$x_i \sum_{j \in C(i)} x_j = 0, \tag{5.5}$$

where $C(i)$ is the ED of link i. If link i is active ($x_i = 1$), then the links in its exclusion domain must be inactive ($x_i = 0$). A Markov chain with E as state space is then defined.

Let $n(x)$ be the number of active links in the transmission pattern (state) x; it is said that the state x is at level i if $n(x) = i$. Transitions of the Markov chain can only occur between states separated by one level: from level i to level $i-1$ when a transmission is finished, and from level i to level $i+1$ when a backoff time finishes and a new connection arrives. Then the transition rates are μ and λ respectively, where λ and μ are the parameters of the exponential distribution of the backoff timer and the transmission duration respectively. The access intensity is defined as $\rho = \lambda/\mu$. It is easy to see that the Markov chain is irreducible and reversible since it is a loss network (see [96]). Then, its unique distribution has a product form:

$$\pi(x) = \frac{\rho^{n(x)}}{Z}, \tag{5.6}$$

where Z is a normalizing constant. The calculus or the estimation of the constant Z is not trivial which seriously limits its application for the obtention of closed formulae. The only exception as we will see below is the line topology.

From this equation it is clear that all transmission patterns with the same number of active links have the same stationary probability. For $\rho > 1$ (observe that there is no stability condition on ρ since it is a loss network), $\pi(x)$ increases with $n(x)$ and the patterns with a high number of active links have a higher probability of appearing. In the limit when $\rho \to \infty$, only the patterns with the highest number of active links have a non-zero probability. This effect translates into high spatial reuse but poor fairness since some links will rarely access the channel ("starvation phenomenon").

The spatial reuse (i.e. mean number of active links normalized by L) can be calculated as:

$$\sigma = \frac{1}{L} \sum_i i\pi_i,$$

where π_i is the probability to have i active links (i.e. the chain is at level i). This probability can be calculated as follows:

$$\pi_i = \frac{N(i)\rho^i}{\sum_k N(k)\rho^k}, \tag{5.7}$$

where $N(i)$ denotes the number of states x such that $n(x) = i$ (i.e. with i active links). Then, the spatial reuse is:

$$\sigma = \frac{1}{L} \sum_i i\pi_i = \frac{1}{L} \sum_i i \frac{N(i)\rho^i}{\sum_k N(k)\rho^k}.$$

In general, it is difficult to obtain a closed formula for $N(i)$. In fact it is possible only for the line topology, for which $N(i) = 2^i C_i^{i+v}$ where $v = L + (l-1) - li$ (number of free spaces once there are i intervals of length l). In this case, the spatial reuse when $L \to \infty$ results:

$$\sigma = \frac{2\rho {y_1}^{l-1}}{1 + 2l\rho {y_1}^{l-1}}, \quad \text{where } y_1 \text{ is the real positive root of } 1 - y - 2\rho y^l. \tag{5.8}$$

For a fixed value of L the spatial reuse increases with ρ, reaching the maximal spatial reuse 1/3 when ρ tends to infinity i.e when the backoff time is much lower than the average exchange time (there is always an attempting connection). On the contrary, when ρ is fixed, the spatial reuse decreases with L.

These results imply that this decentralized protocol can optimally organize the transmissions in space (when the access intensity is high the maximal spatial reuse is achieved). It must be noted that these results can be achieved with an ideal collision avoidance mechanism. One of the reasons for which the real protocol is unable to achieve this maximum spatial reuse, is the discrete distribution of the backoff timer which, as mentioned before, implies a non-null probability of collisions.

This analysis can be seen in a more general context. In fact, the stationary distribution (5.6) can be seen as a Gibbs distribution:

$$\pi(x) = \frac{e^{-H(x)}}{Z},$$

where the function H derives from a Gibbs potential V, i.e. $H(x) = \sum_i V(x_i) + \sum_{i \sim j} V(x_i, x_j)$ and

$$V(x_i) = -x_i \log(\rho) \quad \text{and} \quad V(x_i, x_j) = \begin{cases} +\infty & \text{if} \quad i \sim j \\ 0 & \text{in other case} \end{cases}$$

The normalizing constant Z is usually known as the "partition function" (in Sec. 5.2.2 we will describe a method to estimate it). See [97] for an excellent reference to Gibbs Fields.

Results discussed until now refer only to the line topology since, as mentioned before, closed formulae cannot be derived even for a 2-dimension regular lattice network (it is not possible to calculate $N(i)$). Actually, authors of [21] proved that in this case one or more Gibbs distribution may exist depending on the value of the access intensity ρ. This characteristic will imply that there is a phase transition: for values of ρ smaller than some constant ρ_{c_1}, the border effect (links at the border have less contenders and so higher probability of access the channel) does not propagate into the network and the protocol is long term fair (all links have the same access probability) as the network size goes to infinity. For values larger than ρ_{c_2} the border effect translates into the network independently of the size and the protocol is not fair.

The main conclusion of this work is that CSMA/CA can be spatially efficient, finding spontaneously the maximal spatial reuse states, but unfairness or even "starvation" is the price to pay for it (even in the ideal case without collisions). The trade-off between fairness and spatial reuse can be moderated by appropriately choosing the access intensity or by considering not so regular scenarios, for instance by considering asymmetric exclusion domains.

5.2.2 Truncation of a Reversible Process

The same results of the previous section can be obtained by the truncation of a "free" process. Results of this section correspond to the work introduced in [22] which relates the analyzed problem with statistical mechanics (i.e. the use of probability theory for the analysis of systems with large number of interacting particles). Consider a free process X for which a node accesses the channel independently of the other nodes. It is easy to find the stationary distribution of the process X; let $x \in E = \{0,1\}^L$ be a transmission pattern, then:

$$\pi(x) = p^{n(x)}(1-p)^{(L-n(x))},$$

where $n(x) = \sum_{i=1}^{L} x_i$ is the number of active nodes, and p is the probability that a node is active. With the same assumptions of the previous section, $p = \frac{\rho}{1+\rho}$ (remember that $\rho = \lambda/\mu$ is the access intensity) which results in:

$$\pi(x) = \frac{\rho^{n(x)}}{(1+\rho)^{L-1}}.$$

Clearly, X is a reversible process. Let X' be the restriction of the original process to $E' \subset E$. Assuming that X' is irreducible, it results that its stationary distribution is simply the truncation of the stationary distribution of X to E', i.e.

$$\pi'(x) = \frac{\pi(x)}{\sum\limits_{x \in E'} \pi(x)} = \frac{\rho^{n(x)}}{Z_L} \quad \forall x \in E'.$$

The main problem is to identify E' and to calculate the normalizing constant (partition function) Z_L. In our case, and as discussed in the previous section:

$$E' = \{x \in E : x_i \sum_{j \in C(i)} x_j = 0\}. \tag{5.9}$$

For the line topology, a possible way to approximate the partition function Z_L is based on the method of the subtracted singularities. Let $Z_G(t) = \sum_{L=0}^{\infty} Z_L t^L$ be the generating function (also referred to as grand partition function), then Z_L can be estimated in terms of the smallest pole t_0 of $Z_G(t)$:

$$Z_L \approx -\frac{Res[Z_G(t_0)]}{t_0^{L+1}}.$$

The approximation error decreases exponentially with L. The partition function can be written as:

$$Z_L = \sum_{x \in E'} \rho^{n(x)} = \sum_i \alpha_L^i \rho^i,$$

where α_L^i is the number of patterns with i active links (see (5.7) in previous section, where $\alpha_L^i = N(i)$). For the free process, $\alpha_L^i = C_i^N$ and for the case of CSMA/CA with CSRange = RXRange = 1 (i.e. $l = 3$) the following recurrence can be derived by adding one more link to the line and examining the number of configurations having i active links:

$$\begin{aligned} \alpha_{L+1}^i &= \alpha_L^i + \alpha_{L-2}^{i-1} \quad L \geq 2, & (5.10)\\ \alpha_0^0 &= \alpha_1^0 = \alpha_2^0 = 1. & (5.11) \end{aligned}$$

The first term corresponds to aggregating a non transmitting node and the second one to the case where the $(L+1)$-node is active. Then the following recurrence for Z_L is also valid:

$$Z_{L+1} = Z_L + \rho Z_{L-2} \quad L \geq 2,$$
$$Z_0 = 1,\ Z_1 = 1 + \rho,\ Z_2 = 1 + 2\rho.$$

Therefore the generating function is:

$$Z_G = \frac{1 + \rho t + \rho t^2}{1 - t - \rho t^3}$$

Let t_0 be the smallest (positive) pole of $Z_G(t)$, the approximation for Z_L results:

$$Z_L \approx -\frac{Res[Z_G(t_0)]}{t_0^{L+1}} = \frac{1 + \rho t_0 + \rho t_0^2}{(3\rho t_0^2 + 1) t_0^{L+1}} \tag{5.12}$$

It must be noted that the pole t_0 is the solution of $1 - t - \rho t^3 = 1 - t - \rho t^l = 0$, which is the same equation that appears in the Gibbs analysis of [21] (see (5.8), where the extra 2 factor appears from considering undirected links).

These independent and different works lead to the same results. In [21] authors analyze the spatial reuse and fairness of the protocol; in [22] authors analyze instead the link throughput (which is in fact closely related with the spatial reuse previously considered). Let S_{ij} be the throughput of link $[i, j]$, defined as the mean number of successful transmissions over link $[i, j]$ per unit time. Then,

$$S_{ij} = \rho_{ij} P([i, j] \text{ is active}) = \rho_{ij} P(C([i, j]) \text{ is idle}) = \rho_{ij} \frac{Z_{L \setminus C_{[i,j]}}}{Z_L}$$

where ρ_{ij} is the traffic intensity of packets from node i to node j and $C([i, j])$ is the exclusion domain of link $[i, j]$. An approximation for the link throughput is obtained from the estimation (5.12) of Z_L. In particular, for the line topology where each node communicates equally with its two neighbors, it is proved that the node throughput $S_i = \sum_j S_{ij}$ tends to 1/3 when ρ goes to infinity. As observed before, for very high access intensity, transmissions will be densely packed by the protocol (every third node would be active).

In [89], a similar Markovian model is considered but with different transition rates. The RTS/CTS exchange is taken into account by relating the access intensity ρ with the link capacity, the contention window parameters, the data frame length and the duration of the time slot. More precisely ρ is in fact ρ_{ij} for link (i, j) and so $\rho^{n(x)}$ in (5.6) translates into $\prod \rho_{ij}$. Authors derive an approximation formula for the throughput in this context and consider different fairness scheduling such as proportional and max-min fairness for which distributed algorithms based on local information are discussed.

5.2.3 Matérn like Process

In this section, a radically different approach is considered. The location of the nodes in the network is seen as the realization of some point process [98]. This means that the network can be considered as a snapshot of a stationary random model in the (Euclidean) space and that it is possible to analyze it in a probabilistic way. Stochastic geometry is a powerful tool (suitable to analyze networks on the plane or in higher dimension) that allows to define and compute macroscopic properties of the network by averaging over all potential geometrical patterns of the nodes. There exist several applications of this tool to networking problems being the analysis of MAC protocol, such as ALOHA or CSMA, a particular case. A good reference with many applications of stochastic geometry to wireless networks is [99]. An excellent reference including the mathematical tools required to understand the networking applications is the monograph [20].

Hard Core (HC) models represent a class of point process whose points are never closer to each other than some given distance $h > 0$. If we define $h = \text{RXRange} = \text{CSRange}$, the nodes allowed by CSMA to transmit at a given time are always at a distance larger than h, and so they can be seen as a HC model.

A particular HC model is known as Matérn hard core (MHC). Such a process is constructed from an underlying Poisson point process (p.p.) by removing certain points depending on its neighbors positions and additional marks associated with the points [20]. More precisely, consider that each point x_i has a mark U_i where $\{U_i\}_i$ are independent and identically distributed random variables with uniform distribution on $[0, 1]$. The MHC is defined by the points whose mark is the smallest in a ball centered at them with radius h, i.e. the point x_i is not removed if $U_i < U_j$ for all $j \in B_{x_i}(h)$. See Fig. 5.3 for an example of the Matérn selection process. The MHC is a process-dependent thinning of the original Poisson process but it is not itself a Poisson p.p. (there is no $h > 0$ such that the points of a Poisson p.p. are separated a distance h). The intensity of the MHC can be calculated from the intensity of the original Poisson p.p. This model was applied for instance in [100] to the analyze the performance and

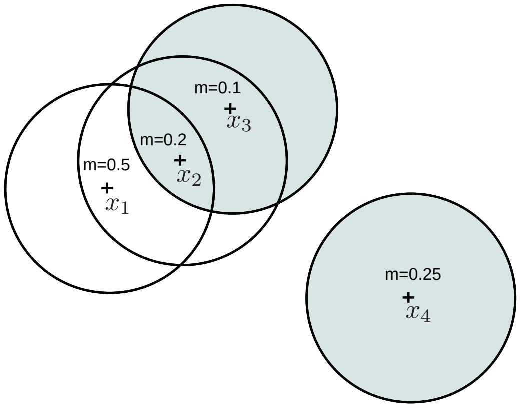


Figure 5.3: Example of the Matérn selection process: only x_3 and x_4 are retained.

planning of dense IEEE 802.11 networks.

It must be noted that the Poisson assumption for the underlying p.p. is not necessary to define a Matérn like process. If it assumed that each point has a mark, the selection of the points with the smallest mark in a given region (e.g. a ball with fixed radius) can be defined irrespectively of the underlying point process. However, the difficulty is to obtain analytical results for this general case; in fact the calculations made for the Poisson case cannot be extended to other distributions. The only exception is perhaps the simplest case where the distribution of the points is deterministic (e.g. a regular fixed topology).

A Matérn like model for CSMA/CA was defined in [20] as follows. By adding a mark to each node, which can be for instance its backoff timer, define a Matérn like process that keeps those nodes whose mark is the smallest in its exclusion domain. In this way, all remaining links verify the condition (5.5) i.e. are valid transmissions for the CSMA/CA mechanism. However, results obtained by this model may be too conservative: a node can be prevented from transmitting even if there is no activity in its exclusion domain. Consider for instance the situation illustrated in Fig. 5.3: node x_1 is not retained because it detects a contender x_2 (the mark of x_2 is smaller than the mark of x_1 and x_2 is in the exclusion domain of x_1) but node x_2 is also not retained because it detects another contender x_3 which is not a contender of x_1. The Matérn CSMA model will not retain neither x_1 nor x_2, but in fact x_1 and x_3 could access the channel simultaneously without creating collisions.

The conservative nature of this model can be easily shown in the case of the line topology. Consider as before RXRange=CSRange=1, then the exclusion domain of each node is constituted by its two left/right neighbors. The probability that a node i (not in the border) is retained by the Matérn CSMA model (i.e. the access probability) is:

$$p_i = P(U_i < min\{U_{i-2}, U_{i-1}, U_{i+1}, U_{i+2}\}) = \frac{1}{5}.$$

This value is directly bounded with the number of contenders of each link; the probability will be equal to the inverse of the number of contenders (exclusion domain size) plus 1. If $L \to \infty$, the border effects are negligible and the spatial reuse (SR) is:

$$\sigma = \frac{Lp}{L} = p = 0.2 \tag{5.13}$$

With this model, for each slot, the set of active links is a subset of the corresponding set previously defined in Sec. 5.2.1, which implies that a smaller SR will be obtained: in fact we obtain 0.2 versus 0.275 of the previous model. The main advantage of this model is that it can be extended to different topologies in the plane (regular or irregular) in which case analytical formulae or approximations can be derived. Remember that for the previous models analytical results can be obtained only for the line topology. For example, if the original p.p. (i.e. the nodes of the network) is assumed to be Poisson in $\mathbb{R}^2$ with intensity λ, the spatial density of active nodes (density of the MHC) can be calculated as $\lambda_{MHC} = p\lambda$, where p is the access probability (see [100]):

$$p = \frac{1 - e^{-\lambda \pi h^2}}{\pi h^2}.$$

It must be noted that this probability represents a spatial average and not a time frequency (i.e. this result does not mean that a node will access the channel with time frequency p).

5.2.4 Conclusions

Several conclusions may be drawn from the study we just presented, some of which we would like to highlight now. Different approaches were considered, ranging from the packing approach valid for a line topology to the analysis of a snapshot of the network by means of Matérn like process, and including classical Markov chains for the asynchronous case. However all these models have made simplifying assumption, some of which were not even explicit.

As already pointed out, the path loss function is assumed to be deterministic and decreasing with the distance (see (5.1)). Clearly, this is a simplifying assumption that ignores the random variations of the channel conditions due to, for instance, shadowing/fading effects. In the following section we analyze if the considered models can be extended to include this more general case. It is not difficult to see that some of the analyzed approaches are not applicable. For instance, the packing approach introduced in Sec. 5.2.1 cannot be used since the space occupied by an active transmission is no longer a deterministic set (i.e. an interval for the line topology), but a random one that could even be disconnected. More precisely, we will propose an extension that includes shadowing/fading effects and we will show that results obtained by this model may be very different of those of previously analyzed ones.

The attentive reader may have been realized that in the previously considered models almost nothing is said regarding the CCA. In fact, this aspect is not explicitly analyzed or described in any of the above considered models. However, we will show that if the CCA is in Carrier Detection (CD) mode (c.f. Sec. 5.1), then the analysis based on the definition of exclusion domains is enough to represent both the CCA and the RTS/CTS handshake. More precisely, in the Carrier Detection mode, the CCA will declare the channel as busy if the transmitter of an intending link detects at least one signal (pertaining to the same network). This is equivalent to receive such a transmission with a power larger than a certain threshold P_{CCA}. If we assume that $P_{\text{CCA}} > P_0$, then this condition is already included in the definition of the exclusion domain (c.f. 5.2). This means that at the moment of deciding to access the channel, if the transmitter does not belongs to the exclusion domain of any of the ongoing transmissions, then the CCA will declare the channel idle. This conclusion is based on a symmetry assumption: if node h can detect node k is also the case in the other sense, i.e. node k can detect node h.

It should be noted that if the CCA in Energy Detection mode, then the models based on exclusion domains (i.e. as a pairwise relation between nodes) are not longer valid. In this case, the interference created by all the ongoing transmission must be taken into account to decide if the channel is idle or not. That is to say, an intending transmitter can sense the channel as busy, even if it does not belong to the exclusion domain of any of the ongoing transmissions. In Chap. 6 we introduce a model for this mode and we compare it with the Carrier Detection mode. We will see that a shot noise process can be used to approximate the CCA in this mode.

From these observations, we may conclude that the previously introduced models are valid when the fading/shadowing effects are negligible and the CCA is performed in the Carrier Detection mode. All in all, it should be noted that some of these models provide analytical results only for the line topology which constitutes only a particular case study. The only model that provides analytical formulae is the Matérn like process with the limitation that its results are intrinsically conservative.

5.3 A Shadowing/Fading Aware Model for CSMA/CA

As we have already seen in the previous section, the assumption of an attenuation function depending only on the distance between nodes facilitates the analysis of the protocol. However it should be clear that this assumption limits severely its accuracy with respect to the real protocol. In this section, we analyze the inclusion of the fading/shadowing effects in the models and we will show that (as expected) this inclusion increases significantly their complexity. To the best of our knowledge there is no analytical model of CSMA/CA that includes this aspect.

In this case, given two nodes h and k, the power received from h by k is:

$$P(h,k) = P_h F_k^h L(h,k), \tag{5.14}$$

which depends now on the random variable F_k^h representing the fading/shadowing from node h to node k. In this work we will consider two different models for the channel variations: Rayleigh fading (suitable when many obstacles are present and there is no line of sight between transmitter and receiver), and Lognormal shadowing. To what modeling is concerned, the difference resides in the distribution of the variables F_k^h. More precisely, the random variables $\{F_k^h\}_{h,k}$ are assumed to be independent and exponentially distributed with parameter μ in the first case, and Lognormally distributed with standard deviation σ in the second one.

As before, the RTS/CTS handshake defines the set of nodes that must be silenced by a tagged link. Whereas in the deterministic case the resulting set (i.e. the exclusion domain) was an interval or a disc, in this case a random set is obtained. More precisely, the silenced nodes are those whose reception power from the tagged transmitter or from the tagged receiver, is larger than a certain threshold P_0. We shall call this set, the Random Exclusion Domain (RED). Thus, the RED of link $i = [T_i, R_i]$ may be defined as:

$$C(i) = \{j = [T_j, R_j] : P(h,k) > P_0 \text{ for } \ h \in \{T_i, R_i\}, k \in \{T_j, R_j\}\} \,. \tag{5.15}$$

Note that even if this definition coincides with 5.2, the random nature of the fading/shadowing implies that this RED is not necessarily made of the nearest neighbors of the link as in the previously analyzed models. We will discuss now if the models introduced in Sec. 5.2, when fading effects are not considered, can be extended to this case or not. We will see that analytical results can only be obtained by defining a Matérn like process, even in the simplest case of the line topology.

5.3.1 Gibbs Distribution

As we have already explained, the packing formalism cannot be applied in this random case. Thus, we focus ourselves on the dynamic case for which two approaches were used in the not random case: Markov chains or the truncation of a reversible process.

In particular, under the assumptions of Sec. 5.2.1, i.e. exponential distribution of the backoff timer and the transmission duration, the same Markov chain can be defined for this case. The state space is still $E = \{0,1\}^L$ and a transmission pattern $x \in E$ must verify the same condition as before $x_i \sum_{j \in C(i)} x_j$ (c.f. (5.9)). Then, we still have a loss network with the same transitions rates, thus the stationary distribution has also a product form:

$$\pi(x) = \frac{\rho^{n(x)}}{Z}.$$

As for the ED definition, this formula coincides with the one already obtained in the previous section (see (5.6)). However, they differ on the normalizing constant Z. This constant is the sum of the state probabilities over the set of all possible states (i.e. valid transmission patterns). While transitions rates are the same as before, the states are not. When channel conditions change, the RED also changes modifying the allowed transmission patterns (states of the chain). This characteristic highly increases the complexity of the analysis. Remember that even in the not random case, the stationary distribution was not evaluated directly, but by calculating $N(i)$ the number of states with i active links. Clearly, this quantity cannot be evaluated here, even for the simplest case of the line topology, and so we cannot calculate neither the spatial reuse nor the throughput.

In a similar way, it is quite easy to see that the analysis based on the truncation of a free process cannot be applied either. Even if the process can be seen as a truncation of the same free process, there is not a recursive relation for Z_L as before (see (5.11)). For the previous case, each active link occupied a fixed interval of length l of the line. In this case, as we have already explained, each link does not only occupy a random sized space (the size of its exclusion domain), but this space changes with time.

From these observations we can conclude that, even if the general framework of the previously analyzed modes can still be valid, its complexity does not allow to obtain analytical results. If we assume that there is some kind of "quasistationarity" in the sense that the shadowing/fading changes slowly enough to allow the chain to converge to its stationarity state, then the stationary distribution can be calculated as:

$$\pi(x) = \int \pi(x|F)dF \quad \text{ where } \quad \pi(x|F) = \frac{\rho^{n(x)}}{Z_F},$$

and $Z_F = \sum_{x \in E_F} \rho^{n(x)}$ being E_F the set of possible patterns assuming a constant fading F. Still, the solution is not trivial and it seem unlikely to find analytical results by this way. We think that techniques used by physicists known as "Spin Glasses" [101] may help to estimate Z_F or Z. However the high complexity of these techniques put this analysis out of scope of this thesis.

5.3.2 Matérn like Process

As mentioned before, when shadowing/fading effects are considered there is no more a minimum fixed distance between two active links. This means that a hard core model such as the MHC introduced in Sec. 5.2.3 cannot be used here. Instead a Matérn like process can be defined by replacing the ball $B_{x_i}(h)$ for the exclusion domain $C(i)$; i.e. a node will be retained by this process if its mark is the smallest in its RED. It is quite easy to see that the retained nodes represent valid transmission patterns for CSMA/CA and that once again this model is conservative (a link may be removed even if there is no active nodes in its exclusion domain). We will refer to this model as MRED as for Matérn RED.

Consider first the reference (deterministic) line topology and assume as before independent uniform marks, $m(i)$ representing the mark of link i. For this case, we can calculate the access probability as follows:

$$P(i \text{ access the channel}|m(i) = t) \quad = \quad \prod_{j \neq i} P(j \notin C(i)) + P(j \in C(i), m(j) > t)$$

$$
\begin{aligned}
&= \prod_{j\neq i} 1 - P(j \in C(i)) + P(j \in C(i))P(m(j) > t) \\
&= \prod_{j\neq i} 1 - P(j \in C(i)) + P(j \in C(i))(1-t) \\
&= \prod_{j\neq i} 1 - tP(j \in C(i)) = \prod_{j\neq i}(1 - tp_{ij})
\end{aligned}
$$

To calculate p_{ij}, i.e. the probability that j belong to the exclusion domain of i, consider the (independent) events $A_{hk} = \{P(h,k) > P_0\} = \{F_k^h > \frac{P_0}{PL(h,k)}\}$ with $h \in \{T_i, R_i\}$ and $k \in \{T_j, R_j\}$. Then p_{ij} can be calculated as follows:

$$p_{ij} = P\left(\bigcup_{h,k} A_{hk}\right) = 1 - P\left(\bigcap_{h,k} A_{hk}^c\right) = 1 - \prod_{h,k} P(A_{hk}^c), \tag{5.16}$$

where:

$$P(A_{hk}^c) = 1 - P(A_{hk}) = 1 - P\left(F_k^h > \frac{P_0}{PL(h,k)}\right),$$

which depends on the fading distribution. Then, the access probability can be written as:

$$P(i \text{ access the channel}) = \int_0^1 \prod_{j\neq i}(1 - tp_{ij})dt.$$

This formula can be easily and accurately calculated numerically, and the spatial reuse is simply the mean access probability:

$$\sigma = \frac{1}{L}\sum_{i=1}^{L} P(i \text{ access the channel}). \tag{5.17}$$

Other performance indicators such as the fairness or throughput, can also be evaluated from the calculation of the access probability. In Sec. 5.4 we will compare the results obtained by this model with the ones obtained by the RED model defined in the previous section.

The example presented here is very simple. However, more general cases can be analyzed. For instance, in [20, 100] results are obtained for a network where transmitters are assumed to be an independently marked Poisson point process on the plane. In these works, authors calculate the access probability of a typical node as well as the probability of joint medium access for several nodes. The coverage probability and the density of successful transmissions can also be evaluated. More precisely, the access probability is given by:

$$p = \frac{1 - e^{-\overline{\mathcal{N}}}}{\overline{\mathcal{N}}},$$

where $\overline{\mathcal{N}} = E(C(0))$ is the mean number of contenders of a typical node in the network. If the intensity of the underlying Poisson p.p. goes to infinity the access probability p goes to $1/\overline{\mathcal{N}}$. In particular if Rayleigh fading with parameter μ is assumed,

$$\overline{\mathcal{N}} = 2\pi\lambda \int_0^\infty e^{-P_0 \mu l(r)} r dr,$$

where $l(r)$ is the path-loss function (see next chapter for more details on this model).

The Matérn process suffers, as we mentioned before, of being too conservative. This means that less connections than in the real protocol will be accepted and so the estimated throughput for the accepted connection will be larger than for the real protocol. However, up to date and to the best of our knowledge, a quantitative analysis of the difference between this model and the real protocol has not been performed. In the following section, we will present such analysis.

5.4 Comparison Results

In this section, we will compare the results obtained by the different models introduced in the previous sections. The objective of this comparison is twofold. On the one hand, we want to evaluate the impact of including the shadowing/fading effects on the models. On the other hand, we want to quantify how conservative the Matérn like models are.

To achieve this we will assume a slotted division of time; i.e. all transmissions start and finish at the same time and the selection of the set of active transmissions is random and is the same, in law, at each time slot, but independent from time slot to time slot. As mentioned before, a very important performance indicator is the spatial reuse (SR). However, the accepted connections may obtain a very poor quality. We hence also measure the mean rate (MR) obtained by each of them, which will be evaluated as $\rho = \log_2(1 + \mathrm{SINR})$.

In this section, we will consider a line topology with $L = 100$ nodes and $d = 1$. Regarding the propagation model, we fix $A = -53$dB, $r_0 = 0.01$ and $\alpha \in \{2.5, 3, 4\}$, which correspond to different propagation scenarios (e.g. for a typical urban environment α is about 3). Finally, we fix $P = 2.3$dBm and $W = -96$dBm for all nodes. Concerning the fading model, we analyze two different models: Rayleigh fading with parameter $\mu = 1$ and Lognormal shadowing with standard deviation $\sigma = 4$dB. The results of this section are obtained by a mix of simulation and analytical results. In the simulations, each algorithm is performed $N = 1000$ times, each time representing a slot. At each time slot, a symmetric matrix of random numbers is constructed representing the symmetric random fading/shadowing. In each repetition (slot), the order at which the nodes intend to access the channel is random, selected according a Uniform distribution in the interval $[0, 1]$. The decision of accepting the transmission or not is taken according to the selected model.

The model introduced in Sec. 5.2.1 will be referred to as CED as for Constant Exclusion Domain. Its Matérn version (cf. Sec. 5.2.3) will be identified as MCED.

The shadowing/fading aware model defined in Sec. 5.3.1 will be referred to as RED as for Random Exclusion Domain, whereas MRED refers to its Matérn version defined in Sec. 5.3.2.

5.4.1 RED Model Analysis

We will analyze here the RED model and we will further compare it with CED. We consider first the mean number of contenders of a typical transmission. Note that this set has the same law for all links provided all transmitter-receiver segments have the same length and orientation. The law of the typical RED is then defined as the law of the RED of any such link. All performance metrics clearly depend on it: a large typical RED results in small access probability and so in poor spatial reuse. On the contrary a small RED results in high access probability and so in high spatial reuse.

Let N_i be the number of links in $C(i)$:

$$N_i = \sum_{j \neq i} \mathbf{1}_{\{j \in C(i)\}}.$$

The mean number of contenders of link i can be calculated as follows:

$$\mathbf{E}(N_i) = \sum_{j \neq i} P(j \in C(i)) = \sum_{j \neq i} p_{ij}, \tag{5.18}$$

where p_{ij} was already calculated in the previous section (see (5.16)) as:

$$p_{ij} = 1 - \prod_{h,k} P(A^c_{hk}) \tag{5.19}$$

where:

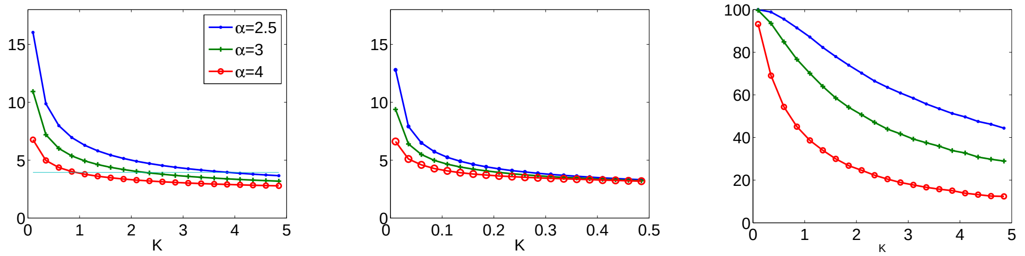
$$P(A^c_{hk}) = \begin{cases} e^{-\frac{P_0}{PL(h,k)}} & \text{for Rayleigh fading,} \\ \Phi(\frac{\log(P_0/PL(h,k))}{\sigma}) & \text{for Lognormal shadowing,} \end{cases}$$

and Φ is the Gaussian cumulative distribution function.

For the line topology and assuming $d(h,k) \neq 0$, we have $\frac{P_0}{PL(h,k)} = \frac{K}{|h-k|^{-\alpha}}$, where $K = P_0/PAd^{-\alpha}$. A similar expression will be valid for a regular topology in $\mathbb{R}^2$. However we cannot calculate explicitly the spatial reuse or the throughput obtained by this model in any case.

Figure 5.4(a) reports on the mean number of contenders for the line topology with Lognormal shadowing for different values of α as a function of the constant K. Results under Rayleigh fading assumption are reported in Fig. 5.4(b). The values of K were selected in order to obtain comparable results for both channel models. Note that when α increases, the mean number of contenders decreases: for high values of α it is less likely that distant links interfere.

We now show that RED differs from CED or equivalently that fading/shadowing plays a role. Firstly, note that as K increases, for all values of α, the mean number of



(a) Lognormal shadowing

(b) Rayleigh fading

(c) Percentage of distant competitors

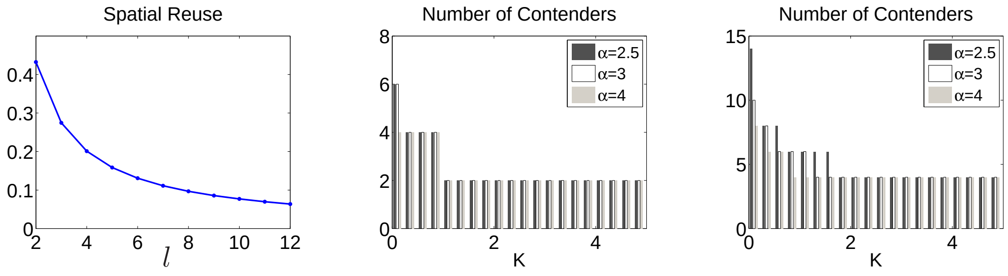
Figure 5.4: Mean number of contenders for RED assuming (a) Lognormal shadowing or (b) Rayleigh fading as a function of $K = P_0/PA$, and (c) percentage of distant competitors for the Lognormal case.

contenders converges to a value approximately equal to 4. This same value for CED would mean that the contenders of each link are its two left/right nearest neighbors. Although this mean number is the same, the actual contenders are not the same. In each sample, we counted the number of times there was at least one contender link outside this region. In Fig. 5.4(c), we show these results in percentage for the Lognormal case. For example, if $\alpha = 3$ and $K = 2.1$ (value of K for which the number of contenders is exactly 4), we obtain that in 51% of the cases, there is a contender which is not a two left/right neighbor. Note that for most values of K and α, in more than 20% of the cases there is a contender which is not a two left/right neighbor. In what follows we will compare RED and CED in terms of spatial reuse (SR) and mean rate (MR).

Comparison with CED

The difference on the number and the nature of the contenders also translates into different performance results. Remember that the CED model introduced in Sec. 5.2.1 was analyzed in a particular case where the RXRange was assumed to be equal to the CSRange and equal to the distance between two consecutive nodes. However, to perform a fair comparison with RED, a more realistic scenario will be assumed in which the carrier and sensing range depends on the variable P_0 (or K). The previous case simply corresponds to a particular value of P_0 that depends on other parameters such as the transmission power or the attenuation function: the relation between K and the RXRange is given by equation (5.3).

The SR for CED can be calculated as a function of the minimum distance between two active nodes using formula (5.4): results are reported in Fig.5.5(a). The most left value is $l = 2$ which corresponds to the case that almost one of two links is active and the SR is approximately 0.5. The most right value $l = 12$ is an extreme case and the corresponding SR is very poor. It must be noted that these results do not depend on the value of α nor on the fading/shadowing model.



(a) CED (b) Lognormal shadowing (c) Rayleigh fading

Figure 5.5: (a) Spatial reuse for CED as a function of l. Number of contenders (or ED size) for those values of K that correspond to the Lognormal shadowing (b) and Rayleigh fading (c) for the RED model.

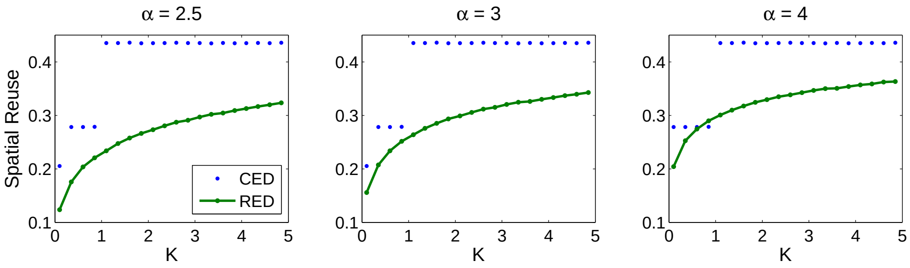


Figure 5.6: Comparison of the Spatial Reuse for CED and RED assuming Lognormal shadowing.

The relation between K and l can be seen through the number of contenders (NC) of an active link (nodes in its exclusion domain), since for the line topology NC is equal to $2 \times (l-1)$. In Fig.5.5(b), results on the NC are shown for different values of α, for the values of K that correspond to those analyzed for RED, under the Lognormal assumption. Three different values are obtained which correspond to $l = 2, 3, 4$. In this case, there is no difference for the different values of α probably due to the chosen granularity of K. If Rayleigh fading is considered, results are reported in Fig. 5.5(c) and once again, the values of K correspond to those analyzed for the RED model. In this case the NC shows more variations with respect to K and α.

Comparing these results with those obtained by RED (see Fig. 5.4), we find that for the same value of K, the number of contenders of RED is larger than for CED. If Lognormal shadowing is considered the differences can be very important. This means that the inclusion of the shadowing/fading effects results in considering more nodes as contenders as for the deterministic case, including nodes that are not the nearest neighbors.

It should be clear now that all performance metrics for CED are step functions of K. In Fig. 5.6, the SR for CED and RED are reported assuming Lognormal shadowing. It can be seen that, for both CED and RED, the SR increases with α and with K;

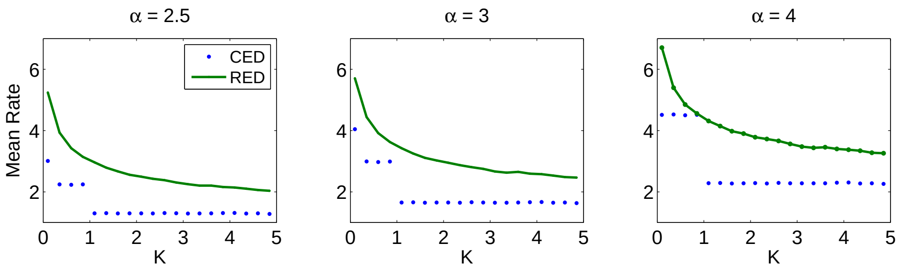
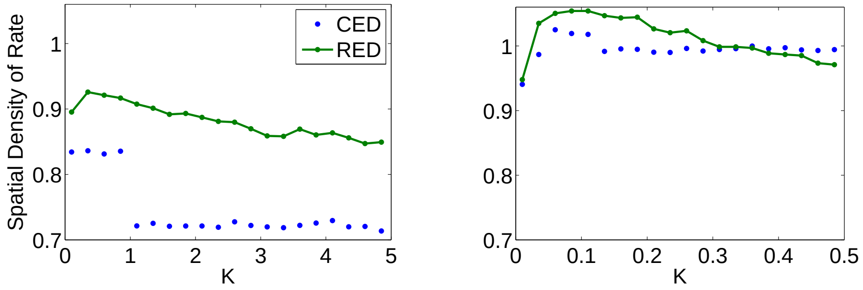


Figure 5.7: Comparison of the Mean Rate for CED and RED assuming Lognormal shadowing.



(a) Lognormal shadowing

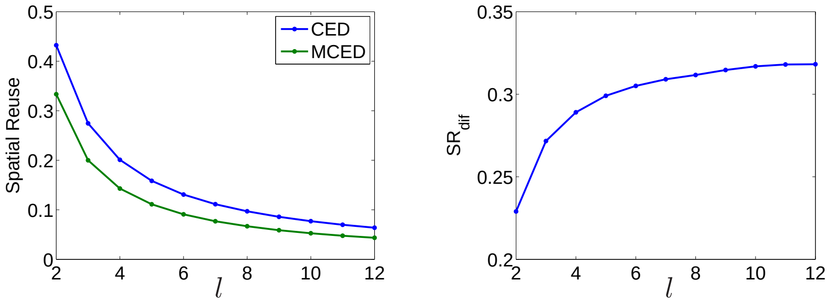
(b) Rayleigh fading

Figure 5.8: Spatial Density of Rate for CED and RED assuming Lognormal shadowing (a) or Rayleigh fading (b) for $\alpha = 3$.

in each case there are less contenders (see Fig. 5.4(a) and Fig. 5.5(b)). Similarities between both models are limited to this observation. Note that for each value of K, the SR for RED is smaller than the corresponding value for CED. This is due to the fact that, as already discussed, the number of contenders of RED is larger than that of CED. However, this difference decreases with α. As an example of the magnitude of the difference on the SR between the two models, if we fix $\alpha = 3$ we obtain that the relative difference is about 25% in mean and can go up to 40%.

We now focus on the comparison in terms of the mean rate (MR). Results are reported in Fig. 5.7. For both models, the mean rate decreases with K: when K increases the number of active connections increases and so does the interference. For CED the impact of α on the SR was not significant, whereas for the MR it is more significative since the SINR depends on it. In particular, for larger values of α, each interferer have less impact on the SINR. In any case, for each value of K and α, RED obtains strictly larger values than CED (exactly the contrary as for SR). The relative difference on the MR are increased with respect to the ones obtained with the SR, being in general larger than 50% and going up to 100%.

The previous observations illustrates on the tradeoff between the spatial reuse and the rate: if the SR increases the MR decreases. To evaluate more accurately this



(a) Spatial Reuse (b) SR_{dif}

Figure 5.9: Spatial Reuse for CED and MCED as a function of l (a) and its relative difference (b).

tradeoff we consider the Spatial Density of Rate (SDR), i.e. the average of the product of the access probability (p_i) and the mean rate (ρ_i) of each connection:

$$SDR = \frac{1}{L}\sum_{i=1}^{L} p_i \rho_i.$$

Results for the SDR are reported in Fig. 5.8 for both attenuation models with $\alpha = 3$. It must be noted that results depend on the channel model. When RED is considered, SDR is relatively constant with K for both channel models. The difference between Lognormal and Rayleigh is that for the former we obtain smaller values. On the other hand, the difference between both propagation models is much more significant for CED. Not only is there a difference in the magnitude, but the dependence on K is accentuated for the Lognormal case.

More than the quantitative results, the goal of this section was to reveal that qualitatively different results are obtained with the two considered models. As such, the shadowing/effects must be taken into account.

5.4.2 Matérn like Models Analysis

In this section we will compare the performance of the Matérn like models MCED and MRED with the one obtained by CED and RED. We have already seen in Sec. 5.2.1, that Matérn like models are conservative, since they underestimate the number of simultaneous connections. Here we want to quantify this difference. We first present the results for the constant exclusion domain, where the shadowing/fading effects are not considered, and we defer to the last part of the section the comparison for the shadowing/fading aware models.

The SR and MR for the CED model were already shown in the previous section. Concerning MCED, we have seen in Sec. 5.2.3 that the SR can be calculated in terms of the exclusion domain size which determines the value of l. Hence, we report results

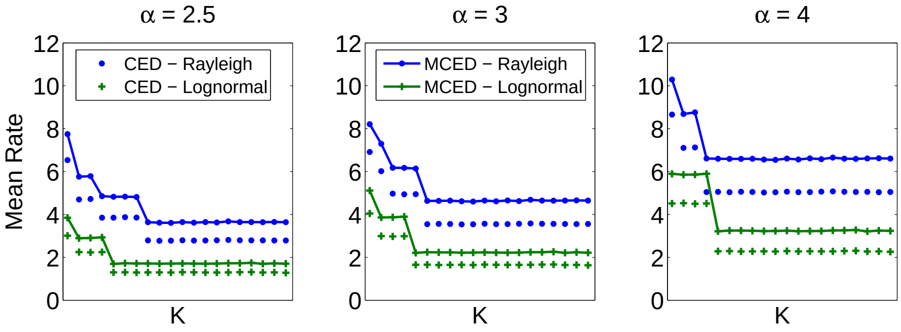
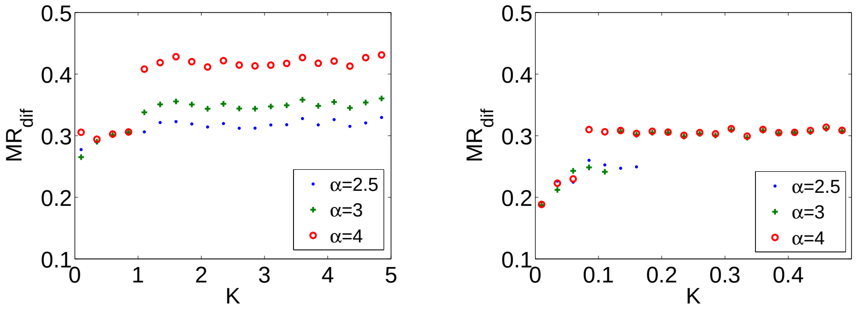


Figure 5.10: Mean Rate for CED and MCED as a function of K for Lognormal shadowing and Rayleigh fading.



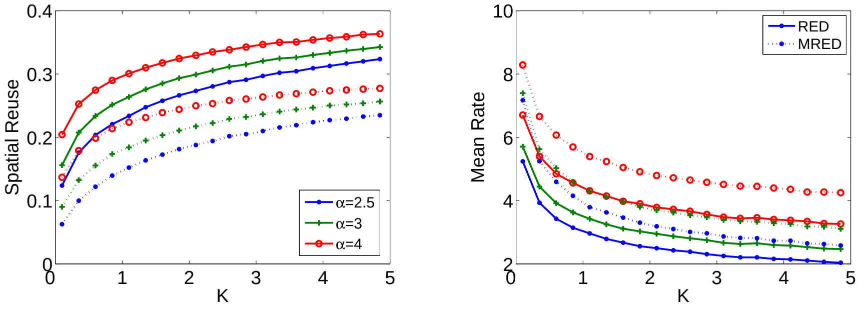
(a) Lognormal shadowing (b) Rayleigh fading

Figure 5.11: Relative difference between CED and MCED for the Mean Rate assuming Lognormal shadowing (a) or Rayleigh fading (b).

of the SR for MCED as a function of l in Fig.5.9(a). As expected, the SR obtained by MCED is always smaller than the one of CED. In Fig. 5.9(b), we report the relative difference ($\text{SR}_{dif} = (\text{SR}_{mced} - \text{SR}_{ced})/\text{SR}_{ced}$) between both models which increases with l and varies between 20% and 30%, corresponding the larger differences to extreme values of l.

We report in Fig. 5.10 the MR for CED and MCED. As expected the MR obtained by MCED is always larger than the one obtained by CED, as it is the contrary for the SR, and for both models the MR increases with α. To ease the comparison we report also the relative difference in Fig. 5.11, from which it can be see that the channel model has different impact. For instance, if we concentrate on the Rayleigh case, the value of α limitedly affects the relative difference whereas for the Lognormal case the differences increase with it, specially for large values of K where the difference can be 40%.

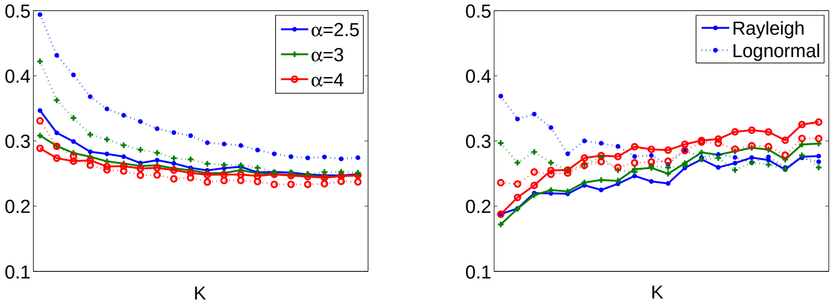
We analyze now the difference between RED and MRED. We report in Fig. 5.12 the spatial reuse and the mean rate for Lognormal shadowing for all values of α. The spatial reuse for the MRED was obtained from the formula (5.17) derived in Sec. 5.3.2. We have already analyzed the RED model in Sec. 5.4.1, so we concentrate on its difference



(a) Spatial Reuse

(b) Mean Rate

Figure 5.12: Spatial Reuse (a) and Mean Rate (b) for RED (solid line) and MRED (dotted line) assuming Lognormal shadowing.



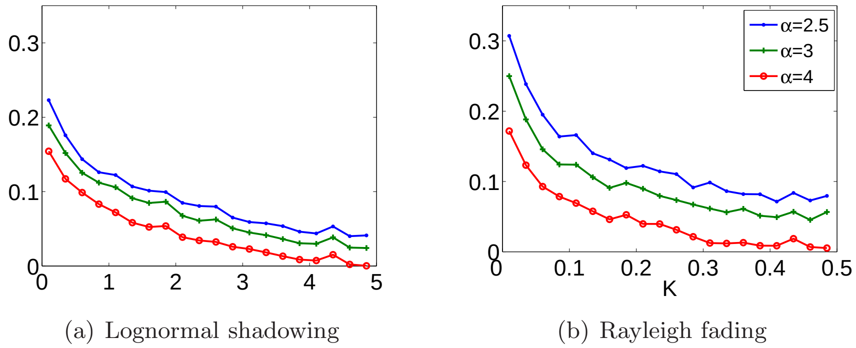
(a) SR_{dif}

(b) MR_{dif}

Figure 5.13: Relative difference between RED and MRED for the Spatial Reuse (a) and the Mean Rate (b).

with MRED. Moreover, the general conclusions of this comparison are the same as the ones already obtained from CED and MCED. That is, the SR (MR) for MRED is always smaller (larger) than the one obtained by RED. Hence we concentrate in the relative difference for the SR and the MR. Results are shown in Fig. 5.13, for both channel models and all values of α. It can be seen that in general, the Lognormal case presents larger differences than that of Rayleigh. For both models, the differences in the SR decrease with α and with K. Regarding the MR, the differences clearly decrease with α for the Rayleigh case, which is not the case under the Lognormal assumption. All in all, it can be concluded that the relative difference is larger than 20% being about 30% in mean for the spatial reuse and mean rate.

As in the previous section, we will also consider the SDR which in certain sense accounts both the SR and the MR at the same time. We report in Fig. 5.14 the relative difference for this metric, for both channel models. It can be seen that the difference can still be important although this time it can be negligible for certain values of K and α. These values correspond to large number of contenders and so low values of



(a) Lognormal shadowing

(b) Rayleigh fading

Figure 5.14: Relative difference between RED and MRED for the Spatial Density of Rate assuming Lognormal shadowing (a) or Rayleigh fading (b).

spatial reuse. The differences for both cases decrease with K and α, with a maximum of 30% (20%) for the Rayleigh (Lognormal) case.

5.5 Conclusions

In this chapter we have considered different models for the analysis of the classic CSMA/CA protocol. In order to correlate the assumptions of these models with the real features of the protocol, we described first those aspects that are essential for modeling purposes. From this analysis two main observations are deduced. On the one hand, these models are based on simplifying assumptions that, in certain cases, may influence the results obtained from them. On the other hand, assumptions regarding some important features of the protocol are not explicitly stated.

In particular, all of the analyzed models are based on the hypothesis that the attenuation function is monotonic on the distance between nodes, and do not include shadowing/fading effects. We then concentrated on the possibility of extending such models so that they include the random variations of the channel conditions. In particular, we proposed an extension based on the definition of random exclusion domains (RED). We showed that very different results are obtained by this model when compared to the previous ones. This comparison shows that the shadowing/fading effect plays a major role in the achieved performance of the protocol and that it must be taken into account.

However, some of the techniques used by the previously analyzed proposals, that allow to obtain analytical results, cannot be extended to this more general case (e.g. the packing formalism or the truncation of a free process). These analytical results (even with the assumption of a deterministic attenuation function) were obtained only for the line topology. The only model capable to provide analytical results for more general topologies and for random channels is a Matérn like model (at least to the best of our knowledge). Unfortunately, this kind of models are known to be intrinsically conservative. We have quantified how conservative this model is, finding that the relative difference is about 30% in terms of spatial reuse and mean rate. If, on the other hand, the spatial density of rate is considered, the differences are reduced, being less than 10% in many cases (specially under the Lognormal shadowing model). Despite these quantitative differences, results obtained are qualitatively similar. For instance, the dependence of the performance indicators (e.g. the spatial reuse and the mean rate) on K is almost the same under the two models. The difference between the obtained results is roughly constant for all values of K.

Finally, in addition to not considering shadowing/fading effects, the consulted references do not explicitly state the considered CCA mode (i.e. the physical sensing of the channel). Indeed, we have shown that if the CCA is assumed to be in Carrier Detection mode, then models based on the definition of an exclusion domain for each active connection are still accurate. However, this is not the case if CCA is in Energy Detection mode. In this case, the interference created by all the ongoing transmission should be taken into account, and so it cannot be considered as a pairwise relationship. Models for this case will be analyzed in the next chapter.

Chapter 6

Modeling and Comparison of the CCA modes

In the previous chapter we analyzed different models that can be used to represent the CSMA/CA protocol. In particular we have identified that an underlying, but essential, assumption in all of them is that the Clear Channel Assessment (CCA) is performed in Carrier Detection (CD) mode. This means that an intending transmitter will consider the channel as busy if it detects at least one signal. However, we have already discussed that if the Energy Detection (ED) mode is used, then the previously analyzed models are no longer valid. In this mode the medium is detected as busy if the total received power is larger than a given threshold. In particular, the models based on the definition of an exclusion domain in a pairwise manner are not adequate. However, it is important to highlight that the importance of the CCA in the performance of the CSMA/CA protocol is crucial, specially when the RTS/CTS is not activated, since in such case it is the mechanism that decides which nodes will be authorized to transmit.

In this chapter we analyze in more detail the differences between these two modes (ED vs CD). To perform a more complete comparison we will focus on modeling techniques that provide closed form formulae for the access probability in both cases. The models that we have studied in the previous chapter correspond to the CD mode and most of them provide analytical results only in particular cases (for instance a line topology). On the other hand, models based on stochastic geometry appear to lead to obtain explicit formulae in many cases. As we shall see, this kind of models are also adequate for deriving results in the ED mode. We are aware that these models represent an intrinsically conservative approximation of the real protocol, but the emphasis of this chapter is put on the comparison between the models of the CCA modes more than in a comparison between the models and the real performance.

The main objective of this chapter is to obtain a representation of the set of transmitters that can access the medium simultaneously according to both considered CCA modes. It should be clear that even if the set of transmitters is assumed to be a Poisson point process, the set of nodes that are allowed to transmit cannot be obtained as an independent thinning of the original process in any case. The analytical framework that we propose in this chapter allows us to calculate the access probability of a typical node and so the density of nodes that can access the channel at the same time. More precisely, we will introduce the additive shot noise (ASN) and the extremal shot noise (ESN) associated with the point process of transmitters and we will show that both CCA modes can be described in terms of them. This property does not only allow us to obtain closed form formulae but also provides a quantification of the impact of considering the total interference received instead of the maximum received power.

6.1 CCA model

We present here the analytical framework that will allow us to calculate the access probability for the two considered CCA modes.

Let $\widetilde{\Phi} = \{(X_i, m_i, F_i)\}$ be an independent marked (i.m.) Poisson point process (p.p.) with intensity λ in $\mathbb{R}^2$ such that:

- $\Phi = \{X_i\}$ denotes the location of the potential transmitters;
- $\{m_i\}$ are iid marks with uniform distribution in the interval $[0, 1]$;
- $\{F_i = (F^i_j : j)\}_i$ represents the fading model. F^i_j represents the fading from node X_i to node X_j and the random variables $\{F^i_j\}_{ij}$ are assumed iid with distribution G. In particular if Rayleigh fading is assumed, F^i_j is exponentially distributed with parameter μ.

In this context we will define two different processes that mean to represent the two different modes of the CCA: the Carrier Detection (CD) mode and the Energy Detection mode (ED). As we will see later, the first mode can be represented as a extremal shot noise whereas the second one can be represented by an additive one.

It should be noted that a model for the CD mode was already presented in Sec. 5.3.2. This model corresponds to a Matérn like process. We briefly recall that in this case, a node x detects a node y (or equivalently that they are contenders) if the power received on x from y is larger than a given threshold P_0. The exclusion set of a node is then defined as the set its contenders. Assuming that the marks $m(.)$ correspond to the time at which each node intends to access the channel, a node will be retained by the Matérn process (i.e. the CCA in CD will consider the channel as idle) if it has the smallest mark in its exclusion domain. Let $\mathcal{N}(X_i)$ be the set of contenders of node X_i, that is:

$$\mathcal{N}(X_i) = \left\{(X_j, F_j) \in \widetilde{\Phi} : P_j F^j_i L(X_j, X_i) \geq P_0, j \neq i\right\}, \tag{6.1}$$

where P_j is the transmission power (that will be assumed constant and equal for all nodes) and $L(X_j, X_i) = l(|X_j - X_i|)$ is the path-loss function. Let e_i be the medium access indicator, that is $e_i = 1$ if node X_i passes the corresponding CCA. Then, the condition to access the medium is:

$$e_i = \mathbf{1}_{\{\forall X_j \in \mathcal{N}(X_i), m_i < m_j\}}. \tag{6.2}$$

This means that a node will be allowed to transmit if it does not detect any of the nodes that intended to access the channel before it. This model captures the key feature that two nodes that detect each other (they belong to their respective exclusion domains) cannot access the channel simultaneously. As mentioned earlier this model is a conservative approximation to the real protocol (cf. Fig. 5.2.3).

It is easy to see that the access condition (6.2) is equivalent to impose that the maximum received power from those nodes whose mark is smaller than m_i, is smaller than P_0. Then, the access indicator e_i can be rewritten as follows:

$$e_i = \mathbf{1}_{\left\{\max\limits_{(X_j,(m_j,F_j))\in\widetilde{\Phi}} P_j F_i^j L(X_j,X_i)\mathbf{1}_{\{m_j<m_i\}}<P_0\right\}}. \tag{6.3}$$

We define then the following process, that represents the set of transmitters that are allowed to transmit by the CCA procedure in the CD mode

$$\Phi_{ext} := \{X_i \in \Phi : e_i = 1\}.$$

We will see that it corresponds to an extremal shot noise, which is why we note it Φ_{ext}.

Following the same ideas as for the CD mode, we can define a way to select those points that pass the CCA in ED mode, i.e. that verify that the interference over them at the moment of accessing the medium is smaller than a given threshold I_0. Once again we base our decision on the marks and define a conservative approximation: we consider the interference created by all previous (i.e. that intend to access the channel before the tagged node) nodes, even if they are not transmitting. Then, the access indicator for the CCA in Energy Detection mode can be defined as:

$$e_i' = \mathbf{1}_{\left\{\sum\limits_{(X_j,(m_j,F_j))\in\widetilde{\Phi}} P_j F_i^j L(X_j,X_i)\mathbf{1}_{\{m_j<m_i\}}<I_0\right\}}. \tag{6.4}$$

The process that represents the set of allowed transmitters in the ED mode is then:

$$\Phi_{add} := \{X_i \in \Phi : e_i' = 1\}. \tag{6.5}$$

In this case, a node will be retained if the interference created by the previous intending nodes over it is smaller than a given threshold.

Both process Φ_{ext} and Φ_{add} are non independent thinnings of the original Poisson process since the decision of retaining a point depends on each particular point. This means that these processes are not Poisson. The intensity of these processes is λp where p is the corresponding access probability of a typical node and will be calculated in what follows. Before proceeding to this calculation we will introduce the shot noise, that will be a key element in the modeling.

6.2 Shot Noise

In this section we will define the additive and the extremal shot noise, and summarize their main properties that will be used later for the analysis of the previously defined processes Φ_{add} and Φ_{ext}. We will concentrate on the definition of a scalar shot noise, since it will be our case, but it can be easily extended to the vectorial case.

While the characterization of the interference in a wireless network as a shot noise is relatively new, the shot noise itself has been the object of studies from a long time now (at least since the works of Campbell in 1909 [102]). This section is based on [20] but some of the results on the closed formula of the shot noise distribution can be found in [103].

Definition 6.2.1 (Additive Shot Noise)**.** Let $\widetilde{\Phi} = \sum_i \varepsilon_{(x_i,m_i)}$ be a marked process on $\mathbb{R}^d$ with marks in $\mathbb{R}^l$ and $L : \mathbb{R}^{d'} \times \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^+$ a non-negative response function. The additive shot noise (ASN) associated with $\widetilde{\Phi}$ and L is defined by:

$$I_{\widetilde{\Phi}}(y) = \int_{\mathbb{R}^d}\int_{\mathbb{R}^l} L(y,x,m)\widetilde{\Phi}(d(x,m)) = \sum_{(x_i,m_i)\in\widetilde{\Phi}} L(y,x_i,m_i), \quad \textit{for} \quad y \in \mathbb{R}^{d'}.$$

The most common case is where the ASN lives in the same space as the point process, i.e. $d' = d$. The term shot noise comes from the interpretation that if we consider that each point x_i defines a "shot" (via the response function L) then $I_{\widetilde{\Phi}}(y)$ is the sum of the effects of the "shots" at point y. Since the response function L is positive, the ASN is well defined but it can be infinite.

The expectation of $I_{\widetilde{\Phi}}(y)$ can be evaluated from Campbell's formula [20] (where $\Lambda(.)$ is the intensity measure of Φ):

$$E(I_{\widetilde{\Phi}}(y)) = \int_{\mathbb{R}^d\times\mathbb{R}^l} L(y,x,m)F_x(dm)\Lambda(dx). \tag{6.6}$$

Please note that what we have defined here as additive shot noise is commonly referred to simply as shot noise. However we want to emphasize the difference with the extremal shot noise.

Instead of the sum of the shots, i.e. the sum of the impacts of all points, we can consider only the strongest impact. This case, which defines an extremal shot noise, is as follows:

Definition 6.2.2 (Extremal Shot Noise)**.** Let $\widetilde{\Phi} = \sum_i \varepsilon_{x_i,m_i}$ be a marked process on $\mathbb{R}^d$, with marks in $\mathbb{R}^l$ and $L : \mathbb{R}^{d'} \times \mathbb{R}^d \times \mathbb{R}^l \to \mathbb{R}^+$ a non-negative response function. The extremal shot noise (ESN) associated with $\widetilde{\Phi}$ and L is defined by:

$$M_{\widetilde{\Phi}}(y) = \sup_{(x_i,m_i)\in\widetilde{\Phi}} L(y,x_i,m_i), \quad \text{for} \quad y \in \mathbb{R}^{d'}.$$

Once again, the ESN is well defined but can be infinite. A very interesting result is that the distribution of $M_{\widetilde{\Phi}}(y)$ can be expressed in terms of the Laplace transform of an associated (additive) shot noise. More precisely, we have that

$$P(M_{\widetilde{\Phi}}(y) \leq t) = E\left(\exp\left\{\sum_{(x_i,m_i)\in\widetilde{\Phi}} \log \mathbf{1}_{\{L(y,x_i,m_i)\leq t\}}\right\}\right). \tag{6.7}$$

This result will allow us to express the distribution of the ESN explicitly for the particular case of a Poisson point process.

6.2.1 Poisson point process

Assume now that $\widetilde{\Phi}$ is an independent marked Poisson point process with intensity measure Λ and mark distribution $F_x(dm)$. In this particular case, we can obtain a closed formula for the Laplace transform of the ASN as well as the distribution of the ESN.

First of all, let us remember that the Laplace transform of $\widetilde{\Phi}$ is given by the following expression:

$$\mathcal{L}_{\widetilde{\Phi}}(f) = E\left(\exp\left\{-\sum_{x_i} f(x_i,m_i)\right\}\right) = \exp\left\{\int_{\mathbb{R}^d}\left(1-\int_{\mathbb{R}^l} e^{-f(x,m)}F_x(dm)\right)\Lambda(dx)\right\}. \tag{6.8}$$

Proposition 6.2.3. The Laplace transform of the ASN $I_{\widetilde{\Phi}}(y)$ associated to $\widetilde{\Phi}$ is in this case given by:

$$\mathcal{L}_{I(y)}(s) = \exp\left\{-\int_{\mathbb{R}^d}\int_{\mathbb{R}^l}\left(1-e^{-sL(y,x,m)}\right)F_x(dm)\Lambda(dx)\right\}. \tag{6.9}$$

This formula also allows one to evaluate higher moments of $I_{\widetilde{\Phi}}(y)$ by differentiating at zero. Prop. 6.2.4 gives a condition that assures that the ASN is absolutely continuous with respect to the Lebesgue measure, i.e. it has a density. Under this condition, the Plancherel-Parseval theorem [97] can be used to derive the distribution of $I_{\widetilde{\Phi}}(y)$ from its Fourier transform.

Proposition 6.2.4. If $\Lambda(\mathbb{R}^d) = \infty$ and if, for each $A \subset \mathbb{R}^+$ of Lebesgue measure 0,

$$\int_{\mathbb{R}^d}\int_{\mathbb{R}^l} \mathbf{1}_{\{L(y,x,m)\in A\}} F_x(dm)\Lambda(dx) = 0,$$

then, for all $y \in \mathbb{R}^{d'}$, the random variable $I_{\widetilde{\Phi}}(y)$ has a density.

Let us now concentrate on the extremal shot noise. The following proposition gives its distribution:

Proposition 6.2.5. The distribution of the extremal shot noise ESN associated to $\widetilde{\Phi}$ is in this case given by:

$$P(M_{\widetilde{\Phi}}(y) \leq t) = \exp\left\{ - \int_{\mathbb{R}^d} \int_{\mathbb{R}^l} \mathbf{1}_{\{L(y,x,m)>t\}} F_x(dm) \Lambda(dx) \right\}. \tag{6.10}$$

This result follows from (6.7) and the Laplace transform of a Poisson p.p., given by (6.8).

The previous results show us that under the Poisson assumption, the distribution of the ESN can be explicitly calculated. For the ASN we may only evaluate its Laplace transform. Its distribution is more difficult to calculate (it can nevertheless be obtained via the inverse fourier transform).

6.3 CCA & Shot Noise

After these general results on the shot noise, let us now come back to the setting of Sec. 6.1, i.e. the process Φ represents the location of the potential transmitters and the marks represents the time at which each node intends to access the channel and the fading. In this section, we will show that when conditioning on the value of the mark $m(.)$ of the considered node, the access conditions (6.3) and (6.4) can be expressed in terms of a shot noise (extremal and additive respectively). The access probability may be obtained by deconditioning on the mark distribution. This will be presented in the next section.

Let us assume that the intending node has a mark $m(\dot{)}$ equal to t. The set of nodes with mark smaller than t, which will be referred to as Φ^t, corresponds to an independent thinning of Φ. Then, the process Φ^t is also a Poisson process with intensity λt. We denote $\widetilde{\Phi}^t = \{(X_i, (m_i, F_i))\}$ with $X_i \in \Phi^t$ the associated marked process.

Let $I^t(y)$ be the ASN associated to $\widetilde{\Phi}^t$, that is:

$$I^t(y) = \sum_{(X_j,F_j)\in\widetilde{\Phi}^t} L(y, X_j, F_j) \quad \text{where} \quad L(y,x,f) = Pfl(|x-y|). \tag{6.11}$$

$I^t(y)$ represents the interference created over y by the nodes that intended to access the channel before it (i.e with smaller marks). Then the medium access indicator for the ED mode (6.4) can be expressed in terms of the ASN $I^t(y)$ as follows:

$$e'_i = \mathbf{1}_{\{I^t(X_i)<I_0\}}. \tag{6.12}$$

In the same way, let M^t be an ESN associated to $\widetilde{\Phi}^t$, defined as:

$$M^t(y) = \max_{(X_j,F_j)\in\widetilde{\Phi}^t} L(y, X_j, F_j) \quad \text{where} \quad L(y,x,f) = Pfl(|x-y|). \tag{6.13}$$

M^t represents the maximum shot of the previous intending nodes over y (i.e. the strongest signal power received at y). Then, the condition to access the channel in CD mode (6.3) can be expressed in terms of the ESN $M^t(y)$:

$$e_i = \mathbf{1}_{\{M^t(X_i)<P_0\}}. \tag{6.14}$$

Under the Poisson assumption of Φ and then of Φ^t, the results for the ASN and the ESN derived in the previous section can be used to calculate the access probability conditioned on the value of the mark. In what follows we will calculate the Laplace transform of the ASN (and the distribution when possible), and the distribution of the ESN for this setting assuming Rayleigh fading and a given path-loss function. The following two subsections discuss the case of ASN and ESN respectively. Since we will concentrate on a typical node 0, we suppress the reference to the point in the calculations (that is $I^t(0) = I^t$ and $M^t(0) = I^t$).

6.3.1 Additive Shot Noise

For the setting of Sec. 6.1, the Laplace transform of I^t is:

$$\begin{aligned}
\mathcal{L}_{I^t}(s) &= \exp\left\{-\int_{\mathbb{R}^2}\int_0^\infty 1 - e^{-sPfl(|x|)}G(df)\lambda t dx\right\} \\
&= \exp\left\{-2\pi\lambda t\int_0^\infty \left(1 - \int_0^\infty e^{-sPfl(r)}G(df)\right) rdr\right\} \\
&= \exp\left\{-2\pi\lambda t\int_0^\infty \left(1 - \mathcal{L}_F(Psl(r))\right) rdr\right\} = \exp\left\{-t\mathcal{K}_F(s)\right\},
\end{aligned} \tag{6.15}$$

where $\mathcal{K}_F(s) = 2\pi\lambda\int_0^\infty (1 - \mathcal{L}_F(Psl(r)))\, rdr$ and $\mathcal{L}_F$ is the Laplace transform associated to the fading distribution. This expression follows directly from (6.9).

In particular, if Rayleigh fading is assumed (with parameter μ), $\mathcal{L}_F(s) = \frac{\mu}{\mu+s}$ and

$$\mathcal{L}_{I^t}(s) = \exp\left\{-2\pi\lambda t\int_0^\infty \left(\frac{r}{1+\frac{\mu}{sPl(r)}}\right) dr\right\}. \tag{6.16}$$

Moreover, if the path-loss function is assumed to be $l(r) = (Ar)^{-\beta}$ with $\beta > 2$, the Laplace transform is:

$$\mathcal{L}_{I^t}(s) = \exp\left\{-\lambda t\left(\frac{sP}{\mu}\right)^{2/\beta}\frac{k(\beta)}{A^2}\right\}, \quad \text{where} \quad k(\beta) = \frac{2\pi^2}{\beta\sin\pi/\beta}. \tag{6.17}$$

If we consider the particular case of $\beta = 4$, equation (6.17) gives:

$$\mathcal{L}_{I^t}(s) = \exp\left\{-\sqrt{2c}s^{1/2}\right\} \quad \text{with} \quad \sqrt{2c} = \frac{\lambda t k(4)}{A^2}\left(\frac{P}{\mu}\right)^{1/2} = \frac{\lambda t\pi^2}{2A^2}\left(\frac{P}{\mu}\right)^{1/2}.$$

Then I^t follows a Lévy distribution with parameter c, which has the following closed formula:

$$P(I^t \leq s) = 1 - \mathrm{erf}\left(\sqrt{\frac{c}{2s}}\right) = 1 - \mathrm{erf}\left(\frac{\lambda t \pi^2}{4A^2}\sqrt{\frac{P}{s\mu}}\right),$$

where erf is the error function, i.e. $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-s^2} ds$. For other values of β, there is no closed formula for the distribution of the additive shot noise.

For $\beta \neq 4$, Prop. 6.2.4 gives a condition that assures that ASN has a density. In the particular case of a Poisson process, sufficient conditions are at least one of the following: the distribution function G of the marks admits a density which is not null (i.e. $G(0) < 1$) or the path-loss function $l(r)$ is strictly decreasing [20]. When ASN has a density, we may use the inverse Plancherel-Parseval theorem [97], to obtain its distribution (provided that $\mathcal{L}_I(s)$ is square integrable):

$$P(a \leq I(y) \leq b) = \int_{-\infty}^{+\infty} \mathcal{L}_I(2i\pi s)\frac{e^{-2i\pi bs} - e^{-2i\pi as}}{-2i\pi s} ds,$$

where $a \leq b$. Clearly, this expression is not tractable and of limited practical use.

6.3.2 Extremal Shot Noise

Concerning the extremal shot noise, we can obtain a closed formula for its distribution directly from (6.10):

$$\begin{aligned} P(M^t \leq s) &= \exp\left\{-\int_{\mathbb{R}^2}\int_{\mathbb{R}} \mathbf{1}_{\{L(0,x,f)\geq s\}} F_x(df)\lambda t dx\right\} \\ &= \exp\left\{-\int_{\mathbb{R}^2} P(F \geq s/Pl(|x|))\lambda t dx\right\} \\ &= \exp\left\{-2\pi\lambda t\int_0^\infty 1 - G(s/Pl(r))rdr\right\} \\ &= \exp\left\{-t\overline{\mathcal{N}}(s)\right\}, \end{aligned} \tag{6.18}$$

where $\overline{\mathcal{N}}(s) = 2\pi\lambda\int_0^\infty 1 - G(s/Pl(r))rdr$ may be interpreted as the mean number of contenders of node 0 since (cf. 6.1):

$$\overline{\mathcal{N}}(s) = E^0\left(\sum_{(X_j,F_j)\in\widetilde{\Phi}} \mathbf{1}_{\{PF_jl(|X_j|\geq s)\}}\right) = \lambda\int_{\mathbb{R}^2} P(F \geq sPl(|x|))dx. \tag{6.19}$$

In particular, if Rayleigh fading with parameter μ is assumed, the result is :

$$P(M^t \leq s) = \exp\left\{-2\pi\lambda t\int_0^\infty e^{\frac{\mu s}{Pl(r)}} rdr\right\}. \tag{6.20}$$

Moreover, if the path-loss function is assumed to be $l(r) = (Ar)^{-\beta}$ with $\beta > 2$ the final result is:

$$P(M^t \leq s) = \exp\left\{-t\overline{\mathcal{N}}(s)\right\} \quad \text{where} \quad \overline{\mathcal{N}}(s) = \frac{2\pi\lambda\Gamma(2/\beta)}{\beta A^2 (s\mu/P)^{2/\beta}} \tag{6.21}$$

and $\Gamma(a) = \int_0^\infty e^{-t} t^{a-1} dt$ is the Gamma function.

6.4 Access Probability

In this section we will calculate the access probability of a typical node, or equivalently the intensity of the processes Φ_{add} and Φ_{ext}. We will assume as default scenario Rayleigh fading and path-loss function $l(r) = (Ar)^{-\beta}$ with $\beta > 2$. We will also consider two different cases: deterministic and random thresholds P_0 and I_0. While it is probably more appropriate to consider the deterministic case, the lack of a closed formula for the distribution of the ASN (except for the case of $\beta = 4$) is a drawback that can be avoided if a random threshold is considered. Moreover, interesting conclusions will be drawn from the comparison of the random and deterministic case. Finally, we will see that the difference between the intensity of both processes can be quantified when the intensity of the original p.p. is large enough.

6.4.1 Random detection threshold

Assume first that both detection threshold P_0 and I_0 are random variables, more precisely, exponentially distributed with parameter γ.

Additive Shot Noise (ED)

Under the Palm probability, the access probability can be calculated as:

$$p_I^{rand} = E^0(e_i') = E^0(e_0') = \int_0^1 P^0(e_0' = 1 | m_0 = t) dt.$$

Recalling the access condition (6.12), this conditional probability can be calculated as follows:

$$\begin{aligned} P^0(e_0' = 1 | m_0 = t) &= P(I^t(0) < I_0) = P(I^t < I_0) \\ &= \int_0^\infty P(s < I_0) P_{I^t}(ds) = \int_0^\infty e^{-\gamma s} P_{I^t}(ds) = \mathcal{L}_{I^t}(\gamma). \end{aligned}$$

By deconditioning with respect to t, we obtain that in this case the access probability is:

$$p_I^{rand} = \int_0^1 P^0(e_0' = 1 | m_0 = t) dt = \int_0^1 \mathcal{L}_{I^t}(\gamma) dt = \int_0^1 e^{-t\mathcal{K}_F(\gamma)} dt = \frac{1 - e^{-\mathcal{K}_{\mathcal{F}}(\gamma)}}{\mathcal{K}_F(\gamma)}, \tag{6.22}$$

where $\mathcal{K}_F(s) = \int_0^\infty (1 - \mathcal{L}_F(Psl(r)))\, rdr$ was introduced before and depends on the fading distribution and the path-loss function. In particular, for our default scenario we obtain that:

$$\mathcal{K}_F(\gamma) = \frac{\lambda k(\beta)}{A^2}\left(\frac{\gamma P}{\mu}\right)^{2/\beta} = \frac{2\pi\lambda\Gamma(2/\beta)\Gamma(1-2/\beta)}{\beta A^2}\left(\frac{\gamma P}{\mu}\right)^{2/\beta}. \tag{6.23}$$

Remark 6.4.1. Observe that when λ goes to infinity, p_I^{rand} is asymptotically equivalent to $1/\mathcal{K}_F(\gamma)$.

In particular this means that the density of accepted transmitters when the intensity of the transmitters is large enough is the constant $\lambda/\mathcal{K}_F(s)$. For our default setting this is:

$$\lim_{\lambda\to\infty} \lambda p_I^{rand} = \frac{\beta A^2}{2\pi\Gamma(2/\beta)\Gamma(1-2/\beta)}\left(\frac{\mu}{\gamma P}\right)^{2/\beta} = \left(\frac{\mu}{\gamma P}\right)^{2/\beta}\frac{A^2}{\pi}\frac{1}{\Gamma(1+2/\beta)\Gamma(1-2/\beta)}. \tag{6.24}$$

Extremal Shot Noise (CD)

As for the additive case, the probability of being retained by Φ_{ext} for a typical node, given that its mark is equal to t, is given by:

$$P^0(e_0 = 1 | m_0 = t) = P(M^t(0) < P_0) = P(M^t < P_0) = \mathcal{L}_{M^t}(\gamma). \tag{6.25}$$

Unfortunately, in this case the Laplace transform does not have a closed formula. Still, we can calculate this probability using (6.18) as:

$$P(M^t < P_0) = \int_0^\infty P(M^t < s)\gamma e^{-\gamma s} ds = \int_0^\infty e^{-t\overline{\mathcal{N}}(s)}\gamma e^{-\gamma s} ds, \tag{6.26}$$

where $\overline{\mathcal{N}}$ was defined in (6.19). By deconditioning with respect to t and by Fubini's exchanging lemma, we obtain that:

$$\begin{aligned} p_M^{rand} &= \int_0^1 P^0(e_0 = 1 | m_0 = t) dt = \int_0^1 \int_0^\infty e^{-t\overline{\mathcal{N}}(s)}\gamma e^{-\gamma s} ds dt \\ &= \int_0^\infty \left(\int_0^1 e^{-t\overline{\mathcal{N}}(s)} dt\right)\gamma e^{-\gamma s} ds = \int_0^\infty \frac{1 - e^{-\overline{\mathcal{N}}(s)}}{\overline{\mathcal{N}}(s)}\gamma e^{-\gamma s} ds. \end{aligned}$$

Remark 6.4.2. When λ goes to infinity, $\frac{1-e^{-\overline{\mathcal{N}}(s)}}{\overline{\mathcal{N}}(s)}$ is asymptotically equivalent to $1/\overline{\mathcal{N}}(s)$.

Then, using (6.21), we obtain that the intensity of Φ_{ext} converges to the following constant:

$$\begin{aligned}\lim_{\lambda\to\infty}\lambda p_M^{rand} &= \frac{\beta A^2}{2\pi\Gamma(2/\beta)}\left(\frac{\mu}{P}\right)^{2/\beta}\int_0^\infty \gamma e^{-\gamma s}s^{2/\beta}ds \\ &= \frac{\beta A^2}{2\pi\Gamma(2/\beta)}\left(\frac{\mu}{P}\right)^{2/\beta}\left(\frac{1}{\gamma}\right)^{2/\beta}\Gamma(1+2/\beta) \\ &= \left(\frac{\mu}{\gamma P}\right)^{2/\beta}\frac{A^2}{\pi}. \end{aligned}\tag{6.27}$$

The last equality follows from the property of the Gamma function: $\Gamma(1+z) = z\Gamma(z)$.

Remark 6.4.3.

1. Since the sum is always larger than the maximum it should be clear that the intensity of Φ_{add} will be always smaller than the one of Φ_{ext}. Moreover, from equations (6.24) and (6.27) it can be observed that the difference between both intensities (for λ large enough) is simply a constant factor:

$$\lim_{\lambda\to\infty}\frac{\lambda p_I^{rand}}{\lambda p_M^{rand}} = \frac{1}{\Gamma(1+2/\beta)\Gamma(1-2/\beta)} < 1. \tag{6.28}$$

   This factor which depends only on the path-loss exponent β, is always less than 1 and can be seen as the loss on the density of active transmitters for considering the total interference received by the intending node, instead of the maximal reception power, to decide if the node can be activated or not (i.e. for considering the CCA in energy detection mode instead of carrier detection one).

2. When $\beta \to \infty$ both functions $\Gamma(1+2/\beta)$ and $\Gamma(1-2/\beta)$ converge to 1. This means that the intensities of Φ_{ext} and Φ_{add} will be equal. Moreover, the function $1/\Gamma(1+2/\beta)\Gamma(1-2/\beta)$ is monotone increasing, which means that the difference on the intensities decreases with β. These conclusions are not surprising: for large values of β the impact of each shot is less significant and so the difference between the maximum and the sum is less important. However, (6.28) gives a quantification of this difference when the intensity of the original process is large enough. Further details on this function will be discussed in Sec. 6.5.

6.4.2 Deterministic Threshold

In this section we will assume that both thresholds P_0 and I_0 are positive constants.

Additive Shot Noise (ED)

The access probability given that the mark of the node 0 is t, is given directly by the distribution of $I^t(0)$. This distribution has a closed formula only for $\beta = 4$ as we mentioned in the previous section. For this case, the access probability is:

$$p_I^{det} = \int_0^1 P(I^t(0) \leq I_0)dt = \int_0^1 1 - \text{erf}(at)dt \quad \text{where} \quad a = \frac{\lambda\pi^2}{4A^2}\sqrt{\frac{P}{I_0\mu}}.$$

Then,

$$\begin{aligned} p_I^{det} &= \int_0^1 \left(1 - \frac{2}{\sqrt{\pi}}\int_0^a e^{-u^2t^2}tdu\right)dt = 1 - \frac{2}{\sqrt{\pi}}\int_0^a \left(\int_0^1 e^{-u^2t^2}tdt\right)du \\ &= 1 - \frac{2}{\sqrt{\pi}}\int_0^a \frac{1-e^{-u^2}}{2u^2}du \\ &= 1 - \frac{1}{\sqrt{\pi}}\left(\frac{-1+e^{-a^2}}{a} + \sqrt{\pi}\text{erf}(a)\right). \end{aligned}$$

In particular if we define b such that $a = \lambda b$, the intensity of Φ_{add} can be written as:

$$\lambda p_I^{det} = \frac{1}{\sqrt{\pi}b}(1 - e^{-a^2}) + \lambda(1 - \text{erf}(a)). \tag{6.29}$$

Remark 6.4.4. When λ goes to infinity, we obtain the following result for the intensity of Φ_{add}:

$$\lim_{\lambda\to\infty} \lambda p_I^{det} = \frac{1}{\sqrt{\pi}b} = \frac{4A^2}{\sqrt{\pi}\pi^2}\sqrt{\frac{I_0\mu}{P}}. \tag{6.30}$$

Some interesting remarks can be deduced about the relation between the random and deterministic threshold assumptions:

Remark 6.4.5. Note that the quotient between the random and the deterministic case is also a constant factor when λ goes to infinity. Using (6.24) with $I_0 = 1/\gamma$ and $\beta = 4$ we obtain that $\lambda p_I^{rand} = \left(\frac{\mu I_0}{P}\right)^{1/2}\frac{2A^2}{\pi^2}$ when λ goes to infinity, and therefore:

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_I^{det}} = \frac{\sqrt{\pi}}{2} = \Gamma(1 + 1/2). \tag{6.31}$$

Remark 6.4.6. Note that in the previous remark $\Gamma(1 + 1/2) = \Gamma(1 + 2/\beta)$. We may then conjecture that for all $\beta > 2$:

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_I^{det}} = \Gamma(1 + 2/\beta). \tag{6.32}$$

If this conjecture is valid, the intensity of Φ_{add} when λ goes to infinity, assuming that $I_0 = 1/\gamma$, can be deduced from the random case:

$$\begin{aligned} \lim_{\lambda\to\infty} \lambda p_I^{det} &= \frac{1}{\Gamma(1+2/\beta)}\lim_{\lambda\to\infty} \lambda p_I^{rand} \\ &= \frac{1}{\Gamma(1+2/\beta)}\left(\frac{\mu I_0}{P}\right)^{2/\beta}\frac{A^2}{\pi}\frac{1}{\Gamma(1+2/\beta)\Gamma(1-2/\beta)}. \end{aligned} \tag{6.33}$$

Note that for $\beta = 4$ we find again the result obtained in (6.30). This conjecture will be validated by simulations in the next section, and enforced by the observations we now present for the ESN case.

Extremal Shot Noise (CD)

The results for this case are already known and the resulting access probability is (see [20, 100] or (6.27)):

$$p_M^{det} = P(M^t(0) \leq I_0) = \frac{1 - e^{-\overline{\mathcal{N}}(P_0)}}{\overline{\mathcal{N}}(P_0)}.$$

In particular, for our default setting (see (6.21)):

$$\overline{\mathcal{N}}(P_0) = \frac{2\pi\lambda\Gamma(2/\beta)}{\beta(P_0\mu)^{2/\beta}A^2}.$$

Remark 6.4.7. The access probability is asymptotically equivalent to $1/\overline{\mathcal{N}}(P_0)$, which means that:

$$\lim_{\lambda\to\infty} \lambda p_M^{det} = \frac{1}{\overline{\mathcal{N}}(P_0)} = \frac{\beta A^2}{2\pi\Gamma(2/\beta)}\left(\frac{P_0\mu}{P}\right)^{2/\beta} = \left(\frac{P_0\mu}{P}\right)^{2/\beta}\frac{A^2}{\pi}\frac{1}{\Gamma(1+2/\beta)}. \tag{6.34}$$

In this case, we can also deduce interesting relations between the different cases we analyzed:

Remark 6.4.8.

1. *In this case the following relation holds:*

$$\lim_{\lambda\to\infty} \frac{\lambda p_M^{rand}}{\lambda p_M^{det}} = \Gamma(1+2/\beta) < 1. \tag{6.35}$$

   Observe that this ratio is the same as the one we discussed for the ASN (see (6.32) in Rmk. 6.4.6), which further supports the validity of our conjecture.

2. If we compare the result for the additive and the extremal case for $P_0 = I_0$ and assuming that our conjecture is valid for $\beta \neq 4$ (see equation (6.33) and (6.34)):

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{det}}{\lambda p_M^{det}} = \frac{1}{\Gamma(1-2/\beta)\Gamma(1+2/\beta)} < 1. \tag{6.36}$$

   Interestingly enough, this ratio is the same as in the case when thresholds were assumed to be random variables (see (6.28)). We have already observed that the intensity of Φ_{add} will always be smaller than that of Φ_{ext}. In this case, we may go further on this: if the fading conditions and the order in which the nodes intend to access the channel is the same, a node that is retained by Φ_{add} will always be retained by Φ_{ext} (recall that we assumed $P_0 = I_0$). This means that the set of nodes allowed to transmit by Φ_{add} is a subset of those accepted by Φ_{ext}. In Fig. 6.1 we can see an example for $\lambda = 0.5$, $\beta = 3.5$ and $P_0 = I_0 = 5e - 005$.

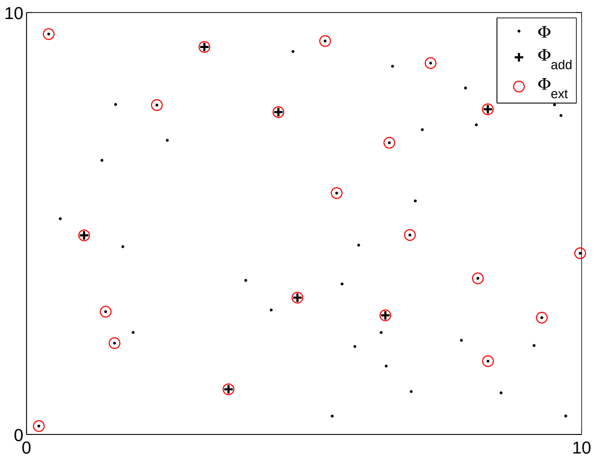


Figure 6.1: The set of points retained by Φ_{add} is a subset of those retained by Φ_{ext}.

3. By combining the results given by (6.36) and (6.31) we obtain that the relation between the additive process with random threshold and the extremal one with deterministic threshold $P_0 = 1/\gamma$ is:

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_M^{det}} = \frac{1}{\Gamma(1-2/\beta)} < 1. \tag{6.37}$$

This same result can be deduced from the equality (see (6.21) and (6.23)):

$$\mathcal{K}_F(\gamma) = \overline{\mathcal{N}}(1/\gamma)\Gamma(1-2/\beta).$$

This means that there is a close relation between the additive and the extremal shot noise even if we considered a random threshold for the first case and a deterministic for the other one.

6.4.3 Additive and Extremal Shot Noise

As we mentioned in the previous chapter there is a third mode of the CCA which corresponds to a combination of the two previously analyzed modes. In this case, the medium is reported busy if at least one signal is detected and the total received power is larger than a given threshold. In order to obtain closed form formulae, we will assume a random detection threshold for the interference and a deterministic one for the maximal received power. The access probability in this mode can be deduced from the following result [104]:

Proposition 6.4.9. Assume the setting of Sec. 6.1, let I and M be the corresponding additive and extremal shot noise. Assume also Rayleigh fading and path-loss function

$l(r) = (Ar)^{-\beta}$ with $\beta > 2$. Let us define:

$$\mathcal{L}_I^{(u)}(s) = E(e^{-sI}\mathbf{1}_{\{M \le u\}}).$$

Then,

$$\mathcal{L}_I^{(u)}(s) = \exp\left\{-2\pi\lambda\left(\frac{sP}{\mu A^\beta}\right)^{2/\beta}\Gamma(1+2/\beta)\left(\Gamma(1-2/\beta)+\frac{2}{\beta}\Gamma(-2/\beta,u\mu)\right)\right\},$$

where $\Gamma(a,x) = \int_x^\infty e^{-t}t^{a-1}dt$ is the incomplete Gamma function.

Proof.

$$\begin{aligned}
\mathcal{L}_I^{(u)}(s) &= E(e^{-sI}1_{\{M\le u\}}) = E\left(e^{-s\sum\limits_{(x_i,F_i)\in\widetilde{\Phi}} PF_il(|x_i|)}\prod_{(x_i,F_i)\in\widetilde{\Phi}}\mathbf{1}_{\{PF_il(|x_i|)\le u\}}\right)\\
&= E\left(e^{-\sum\limits_{(x_i,F_i)\in\widetilde{\Phi}} sPF_il(|x_i|)-\log\mathbf{1}_{\{PF_il(|x_i|)\le u\}}}\right)\\
&= \exp\left\{-\lambda\int_{\mathbb{R}^2}1-e^{-sPFl(|x|)}\mathbf{1}_{\{PFl(|x|)\le u\}}G(df)dx\right\}\\
&= \exp\left\{-2\pi\lambda\int_{\mathbb{R}^2}1-E\left(e^{-sPFl(r)}\mathbf{1}_{\{PFl(r)\le u\}}\right)rdr\right\}.
\end{aligned}$$

Under the assumption of Rayleigh fading, the distribution of the marks F is exponential of parameter μ and:

$$\begin{aligned}
E\left(e^{-sPFl(r)}\mathbf{1}_{\{PFl(r)\le u\}}\right) &= \int_0^\infty e^{-sPfl(r)}\mathbf{1}_{\{Pfl(r)\le u\}}\mu e^{-\mu f}df\\
&= \mu\int_0^{u/Pl(r)} e^{-(sPl(r)+\mu)f}df\\
&= \mu\left(\frac{1-e^{-(s-\mu/Pl(r))u}}{sPl(r)+\mu}\right). \qquad (6.38)
\end{aligned}$$

Then,

$$\mathcal{L}_I^{(u)}(s) = \exp\left\{-2\pi\lambda\int_{\mathbb{R}^2}\left(\frac{sPl(r)+\mu e^{-(s-\mu/Pl(r))u}}{sPl(r)+\mu}\right)rdr\right\}.$$

For the path-loss function $l(r) = (Ar)^{-\beta}$ we obtain that (similar calculus to obtain (6.17)):

$$\int_0^\infty \frac{r}{1+\mu/sPl(r)}dr = \left(\frac{sP}{\mu}\right)^{2/\beta}\frac{\Gamma(2/\beta)\Gamma(1-2/\beta)}{\beta A^2}, \qquad (6.39)$$

and by simply change of variables we can calculate:

$$\begin{aligned}\int_0^\infty \frac{\mu e^{-(s-\mu/Pl(r))u}}{sPl(r)+\mu} r dr &= \left(\frac{P}{\mu}\right)^{2/\beta} \frac{e^{-su}}{\beta A^2} \int_0^\infty \frac{t^{2/beta} e^{-tu}}{s+t} dt \\ &= \left(\frac{P}{\mu}\right)^{2/\beta} \frac{e^{-su}}{\beta A^2} \left(s^{2/\beta}\Gamma(1+2/\beta)e^{us}\Gamma(-2/\beta, us)\right) \\ &= \left(\frac{sP}{\mu}\right)^{2/\beta} \frac{\Gamma(1+2/\beta)\Gamma(-2/\beta, us)}{\beta A^2}, \end{aligned} \tag{6.40}$$

where the following equality for $x, y > 0$ has been used:

$$\int_0^\infty e^{-xt} \frac{t^w}{y+t} = y^w \Gamma(1+w) e^{xy} \Gamma(-x, xy).$$

The proof concludes by replacing (6.39) and (6.40) in (6.38). □

From this proposition we can directly obtain the access probability for the third mode of the CCA. In this case the medium access indicator is

$$e_0 = \mathbf{1}_{\{I^t \leq I_0, M^t \leq P_0\}}.$$

If we assume that I_0 is exponential with parameter γ, and P_0 is a positive constant, then the access probability is given by:

$$\begin{aligned} p_{I,M}(\gamma, P_0) &= \int_0^1 P(e_0 = 1 | m_0 = t) dt = \int_0^1 \mathcal{L}_{I^t}^{\gamma}(P_0) dt \\ &= \int_0^1 e^{-C(\gamma, P_0)t} dt = \frac{1 - e^{-C(\gamma, P_0)}}{C(\gamma, P_0)}, \end{aligned} \tag{6.41}$$

where $C(\gamma, P_0)$ is given by:

$$C(\gamma, P_0) = \pi\lambda \left(\frac{\gamma P}{\mu}\right)^{2/\beta} \frac{\Gamma(1+2/\beta)}{A^2} \left(\Gamma(1-2/\beta) + \frac{2}{\beta}\Gamma(-2/\beta, P_0\mu)\right).$$

Remark 6.4.10.

1. It is easy to see that $C(\gamma, P_0) = \mathcal{K}_F(\gamma) + N(\gamma, P_0)$ where the factor $N(\gamma, P_0)$ goes to $\overline{\mathcal{N}}(P_0)$ when $\gamma \to 0$ and that

$$\lim_{\lambda \to \infty} \lambda p_{I,M}(\gamma, P_0) = \frac{1}{C(\gamma, P_0)}.$$

2. Is quite simple to derive the results presented earlier for the extremal and additive shot noise. For instance,

$$\lim_{P_0 \to \infty} C(\gamma, P_0) = \pi\lambda \left(\frac{\gamma P}{\mu}\right)^{2/\beta} \frac{\Gamma(1+2/\beta)}{A^2} \Gamma(1-2/\beta) = \mathcal{K}_F(\gamma).$$

This follows directly from the limit to zero of the incomplete Gamma function and correspond to the case where no restriction is imposed over the maximum, i.e. only the condition over the interference is considered.

Using that $\gamma^{2/\beta}\Gamma(-2/\beta, \gamma P_0) \to \frac{P_0^{2/\beta}}{2/\beta}$ when γ goes to zero, we obtain that:

$$\lim_{\gamma \to 0} C(\gamma, P_0) = \frac{\pi\lambda\Gamma(1+2/\beta)}{A^2}\left(\frac{P}{P_0\mu}\right)^{2/\beta} = \overline{\mathcal{N}}(P_0).$$

This case correspond to imposing no restriction over the interference so that only the condition over the maximum is considered.

6.5 Simulations

In this section we will present some simulations related to the results showed in the previous sections. In particular we will pay special attention to the validation of the conjecture presented in the remark 6.4.6.

As in our analytical framework we consider a network topology where the set of transmitters are distributed according a Poisson p.p. in $\mathbb{R}^2$ with intensity $\lambda \in \mathbb{R}^+$. We generate a Poisson process in a circle of radius $R = 40$ and consider only those points that fall in an observation window equal to $[-5, 5]^2$. The points outside the observation window have an influence on the points inside it, but we do not consider them to minimize border effects. Theoretical results on this matter can be found in [105]. Other network parameters considered in this section are: $P = 10^{2.3}$, $A = 10^{5.3/3}$ and $\mu = 1$. In addition, otherwise state, default values are $\gamma = 20000$ $(1/\gamma = 5e - 5)$ and $P_0 = 1/\gamma$. All simulations were run a time long enough to assure that the variance is relatively small.

Before going into details on the simulations we will give some numerical results of the calculus for the intensity of Φ_{add}, Φ_{ext} and $\Phi_{I,M}$. More precisely, we have deduced the following main relations:

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_I^{det}} = \lim_{\lambda\to\infty} \frac{\lambda p_M^{rand}}{\lambda p_M^{det}} = \Gamma(1+2/\beta), \tag{6.42}$$

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_M^{det}} = \frac{1}{\Gamma(1-2/\beta)}, \tag{6.43}$$

$$\lim_{\lambda\to\infty} \frac{\lambda p_I^{rand}}{\lambda p_M^{rand}} = \lim_{\lambda\to\infty} \frac{\lambda p_I^{det}}{\lambda p_M^{det}} = \frac{1}{\Gamma(1-2/\beta)\Gamma(1+2/\beta)}. \tag{6.44}$$

In Fig. 6.2, we report on the functions defined on the right hand side of these equations. First, if we concentrate on the left most figure, which gives the relation between the random and the deterministic case for both processes, it can be seen that the minimum is attained at $\beta = 4$ for which the difference is of 12%. Moreover, for

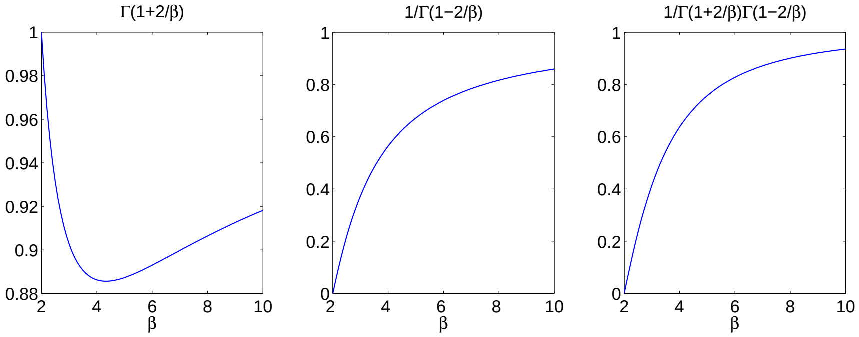


Figure 6.2: Functions defining the relations given in equations (6.42)-(6.44)

values smaller than $\beta = 3$, the function is larger than 0.9, which implies a difference of less than 10%. This means that the difference between a random and a deterministic threshold is only marginal.

As we mentioned earlier it can be observed that:

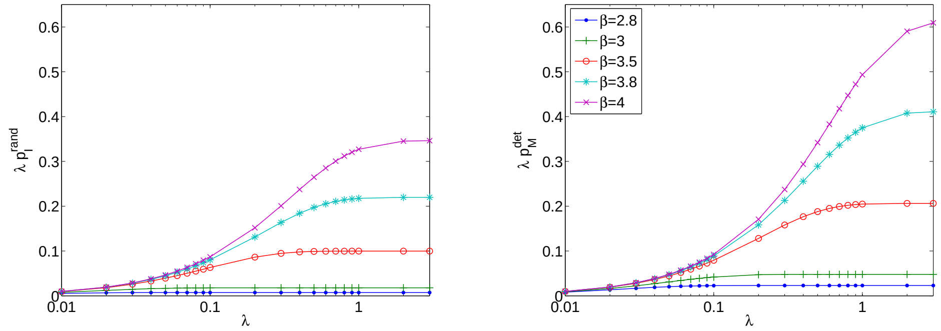
$$\lim_{\beta\to\infty} \frac{1}{\Gamma(1-2/\beta)} = \lim_{\beta\to\infty} \frac{1}{\Gamma(1-2/\beta)\Gamma(1+2/\beta)} = 1. \tag{6.45}$$

On the one hand, this means that for both random and deterministic cases, the intensities of Φ_{ext} and Φ_{add} tend to be equal as β increases (see (6.44)). On the other hand, the same conclusion is valid when the intensity of Φ_{add} for the random case is compared with the one of Φ_{ext} for the deterministic one (6.43).

However, as we can observe in Fig. 6.2, the limit is approached for very large values of β which is not a reasonable scenario in our context. Moreover the values obtained for β close to 2 are very small which implies that the corresponding differences on the intensities are very large. For instance, if we concentrate on the right most figure we have that for $\beta = 2.1$ and $\beta = 4$ the obtained values are 0.05 and 0.64 respectively. This means that the asymptotic intensity obtained for the additive shot noise will be 5% and 64% of that of the extremal shot noise respectively. The same conclusions can be obtained for the relation defined in (6.43).

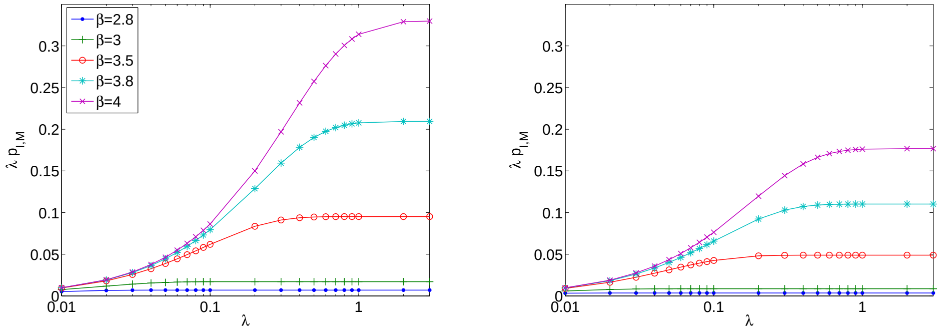
The previous relations are valid when $\lambda \to \infty$. However, it would be interesting to assess which is the minimum value (λ^*) of λ such that the previous conclusions are valid. We report in Fig. 6.3 the results for the intensities λp_I^{rand} and λp_I^{det} given by the formulas (6.24) and (6.34) respectively. We show the results as a function of λ for different values of β.

A first obvious observation is that the intensity of Φ_{ext} is larger that the one of Φ_{add} for each value of λ and β (cf. Rmk 6.4.8). It can be observed also that the value of λ^* for Φ_{ext} is always larger than the corresponding value for Φ_{add}. For instance, if $\beta = 3$, we obtain that $\lambda^*_{add} \approx 0.07$ whereas $\lambda^*_{ext} \approx 0.2$ (i.e. a multiplicative factor of three). This difference decreases with β. Moreover, λ^* increases with β: while small values are obtained for $\beta = 2.8, 3$, highest values of λ are required for larger values of β. This is



(a) Intensity of Φ_{add} with random threshold (b) Intensity of Φ_{ext} with deterministic threshold

Figure 6.3: Density of accepted transmission for Φ_{add} with random threshold and Φ_{ext} with deterministic threshold as a function of λ for several values of β.



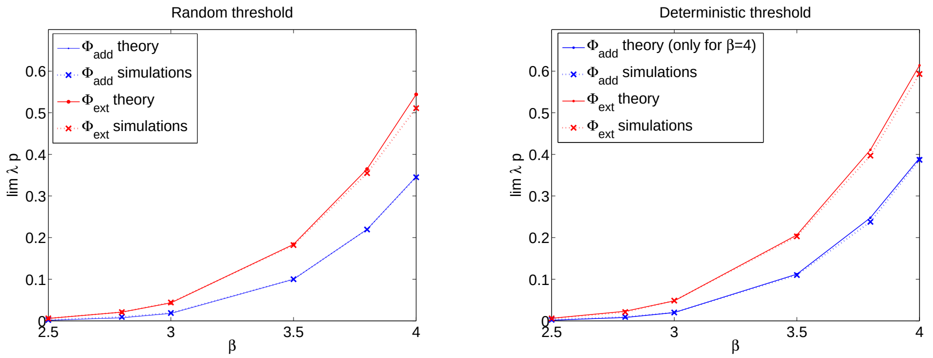
(a) Intensity of $\Phi_{I,M}$ with $\gamma = 2e4$ and $Po = 1/\gamma$ (b) Intensity of $\Phi_{I,M}$ with $\gamma = 2e4$ and $Po = 5e - 6 < 1/\gamma$

Figure 6.4: Density of accepted transmission for $\Phi_{I,M}$ with $\gamma = 2e4$ with (a) $Po = 1/\gamma$ and (b) $Po < 1/\gamma$ as a function of λ for several values of β.

natural since a larger β limits the interactions between nodes, requiring more of them in order to converge.

As we analyzed in Sec. 6.4.3 we can impose a restriction over the sum and the maximum at the same time. Results on the density of active transmissions for this case (see (6.41)) are reported in Fig. 6.4. In Fig. 6.4(a) we chose $P_0 = 1/\gamma$ and the obtained results are very similar to those obtained for Φ_{add} (see Fig. 6.3(a)). This is reasonable since the restriction over the sum is clearly more restrictive. However, if we chose P_0 smaller than $1/\gamma$, in this case the results are very different: about twice time smaller when P_0 is 10% smaller than $1/\gamma$ (see Fig. 6.4(b)).

Up to now we have analyzed numerical results given by the formulas obtained in the previous sections. We will compare now these results with the ones obtained by simulations. More precisely, we show in Fig. 6.5 the asymptotic intensities of Φ_{add} and



(a) Limit of the intensity of Φ_{add} and Φ_{ext} with random threshold

(b) Limit of the intensity ofΦ_{add} and Φ_{ext} with deterministic threshold

Figure 6.5: Validation by simulations of the limits for the density of accepted transmissions for the ASN (I) and the ESN (M) for both random and deterministic threshold. In particular validation of the conjecture made in 6.4.6.

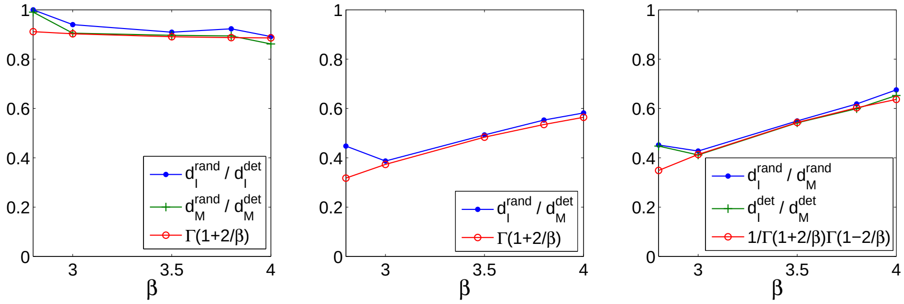
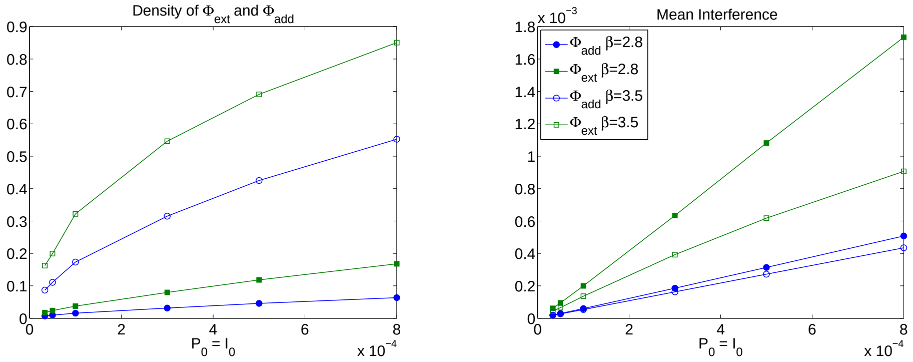


Figure 6.6: Simulations results for the relation given by (6.44)-(6.43).

Φ_{ext} when λ goes to infinity (i.e. is larger than λ^*) for the random and the deterministic threshold case. The first observation is that the match between simulations and theoretical results are almost perfect. Special attention should be paid to the results of Φ_{add} in the deterministic case (Fig. 6.5(b)) for which theoretical results were obtained only for $\beta = 4$. As we can appreciate our conjecture (cf. Rmk 6.4.6) is validated for all values of β.

In Fig. 6.6 we report on the results of the quotients defined in equation (6.44) to (6.43) obtained this time by simulations. We also show the theoretical limit. Once again the value of λ is chosen larger than the corresponding λ^*. As we can see the results are very accurate. The largest differences correspond to $\beta = 2.8$ which are due to the fact that each factor on the quotient is very small (e.g. for Φ_{add} the intensities are 0.0073 and 0.008 for the random and deterministic case).

Finally we analyze for the deterministic threshold the impact of the parameters P_0 and I_0 over the asymptotic intensity of Φ_{add} and Φ_{ext} for $\beta = 2.8$ and $\beta = 3.5$. Results are shown in Fig. 6.7(a). As expected the obtained results for Φ_{ext} are larger than



(a) Comparison of the asymptotic densities of Φ_{add} and Φ_{ext} as a function of $P_0 = I_0$

(b) Mean interference obtained by Φ_{add} and Φ_{ext} as a function of $P_0 = I_0$

Figure 6.7: Comparison of the asymptotic intensities and mean interference for Φ_{add} and Φ_{ext} as a function of $P_0 = I_0$ with $\beta = 2.8$ and $\beta = 3.5$.

those of Φ_{add} for all values of P_0. Moreover, when β increases the maximum and the sum of the response function L (6.11) decreases, which means that more connections are accepted, i.e. larger values of intensities are obtained. However it seems that the impact of the value of β is stronger for the ESN, resulting in a difference between the maximum and the sum that decreases with β: whereas the intensity of Φ_{add} is about 40% of the intensity of Φ_{ext}, this factor is about 60% when $\beta = 3.5$.

Finally, we report Fig. 6.7(b) the mean interference for each of the analyzed cases. For each node the interference created by all the active nodes over it is calculated at several epochs during the simulations, and the mean over all nodes on the observation windows is what we refer here. We analyze the interference since less interference will translate into higher probability of successful transmission.

Clearly the interference experienced by those nodes retained by Φ_{ext} will be larger than those retained by Φ_{add}: in the second case less connections are accepted and so less interference is created. Another natural observation is that the interference will be lower for larger values of β.

To see more in detail the differences between the two modes we plot in Fig. 6.8 the quotient of the mean interference and P_0 (resp. I_0). As we can see, the interference for the ASN is about half of the threshold I_0. This observation was not necessarily true a priori. Even if we impose that at the moment of connecting each node has an interference smaller than I_0, as other nodes became active the interference on the already connected nodes could exceed the selected threshold. However, our simulations suggest that the condition is restrictive to the level of obtaining a mean interference that is always smaller than the decision threshold. Moreover, this quotient does not change significantly with β, which is not the case for the ESN (we have already observed this stronger effect of β over the ESN for the intensity). It does not change with P_0 either. If we concentrate now on the results obtained by the ESN we can observe that the mean interference is always larger than P_0 (e.g. it is more than twice for $\beta = 2.8$).

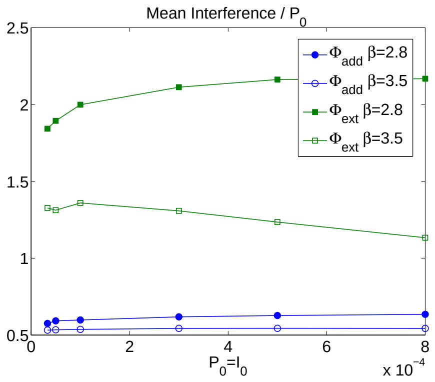


Figure 6.8: Mean interference divided by the threshold $P_0 = I_0$.

This shows that the interference for the ASN and the ESN are very different for small values of β despite this difference decreases with it.

6.6 Conclusions

In this chapter we proposed an analytical framework to analyze two modes of the CCA used in CSMA: Carrier Detection (CD) and Energy Detection (ED). Assuming that the set of network nodes is a realization of a Poisson point process in $\mathbb{R}^2$, we have shown that the set of active transmitters under these two modes (CD and ED) can be approximated by an extremal and an additive shot noise respectively (ESN and ASN) associated to the original point process. This framework allowed us to calculate the access probability for both cases and to quantify the differences between them.

It is easy to see that under the same network conditions the set of nodes allowed to transmit under the ED mode (i.e. retained by the ASN) is a subset of the ones allowed by the CD (i.e. retained by the ESN). We deduced here a quantification of this difference that depends only on the path-loss function exponent when the density of nodes is large enough (and Rayleigh fading is assumed). More in detail, we analyzed two different scenarios: random and deterministic threshold. We found that the intensity of the process ASN and ESN for both cases is related, and that their ratio when the density of nodes increases is the same and again depends only on the path-loss exponent. We want to highlight that for the ASN with deterministic threshold analytical results could only be obtained for $\beta = 4$. However, we conjectured that the same conclusions are valid for other values of β, a claim that was validated by simulations. Also by simulations we analyzed which is the mean interference experienced by an active transmitter for both processes. We found that under the ASN the mean interference is about half of the threshold used to decide its transmission, and that the impact of β is not significant. However, for the ESN the interference can be much larger than the corresponding threshold and the impact of β is stronger than for the ASN. Clearly, as β increases the differences between the ASN and the ESN decreases. Finally, results of considering a condition over the sum and the maximum at the same time were also deduced and compared with the previous modes. An important observation is that the condition over the sum is more restrictive, at least when comparable thresholds are considered.

This chapter is a first approximation to considering an additive version of the interference to decide which nodes can transmit instead of a Matérn like process. In this sense several aspects still require to be analyzed. For instance, a more formal comparison of the trajectories of both process (ASN and ESN) can be done (e.g. following the ideas of [106]). In addition, other fading models as well as other path-loss functions can be considered. Also, a different model can be assumed to the network nodes, for instance a Poisson Cluster can be used for which analytical results on the shot noise can be derived. Up to now, we have concentrated on the location of the transmitter but the location of the receptor is crucial to define a successful transmission. For instance, the receptor can be assumed to be at a constant distant of the transmitter, it can be the nearest point of the process or even a multicast scenario can be considered. With respect to the approximation of the CCA modes by the ASN and ESN, a careful evaluation of how conservative they are should be performed. It would be interesting to verify if both cases are equally conservative or not, and if there exist scenarios (e.g. low or high fading) in which the approximations are more accurate.

Chapter 7

Multiple Access Mechanisms with Performance Guarantees

In the previous chapter we introduced the basic operation of the classic CSMA/CA protocol, and we focused on the comparison of different models designed to analyse its performance. The interest on such a mechanism is due to the ever increasing use of wireless technologies to access the Internet. Moreover, the proliferation of new services with high requirements in terms of quality of service (e.g. high quality video), increases the need of mechanisms with very high levels of performance, or ideally with performance guarantees (which is neither the case for CSMA/CA). This chapter bears then on the design and the quantitative evaluation of MAC mechanisms for wireless ad-hoc networks with performance guarantees. By this, we mean mechanisms where each accepted connection obtains a minimum rate or equivalently a minimum SINR level and which are adapted to the wireless ad-hoc network framework, namely are fully decentralized, power efficient and provide a good spatial reuse.

As we mentioned before, CSMA/CA suffers of many well known weaknesses. For instance, the already explained problem of the "exposed terminal" is not solved, and this unnecessarily reduces the number of simultaneous transmissions [107]. Also, the use of a fixed transmission power, independent of the distance between the transmitter and the receiver, prevents certain transmissions that could be accommodated with power adaptation. Moreover, there are no guarantees in terms of transmission success nor in terms of performance (e.g. rate); this makes CSMA/CA inappropriate for real time traffic. This lack of guarantees is mainly due to the fact that the *additive* character of the interference is not taken into account in the protocol. Indeed, we show in Sec. 7.2 that when shadowing/fading effects are taken into account, one may have a large collection of transmitters such that (i) each transmitter is outside the set of contenders of some tagged transmitter; (ii) none of these transmitters contend with each other and hence all are allowed to transmit simultaneously; (iii) the interference level at the tagged node tends to infinity with the size of the collection.

Much effort has been put on the improvement of the CSMA/CA performance. Most of the papers on the matter are devoted to modifying particular parameters defined in the original CSMA/CA protocol (see for example [108, 109, 110]). We assumed instead a *clean slate* approach, aiming at revisiting or defining multiple access mechanisms which (i) are decentralized and hence adapted to the MANET context (ii) guarantee a certain level of performance for all accepted transmissions in the presence of variable channel conditions due to fading/shadowing effects.

In order to achieve these QoS guarantees, we argue that there is a need for mechanisms which explicitly take the SINR at the receivers into account for deciding which transmissions to accept/schedule from a given set of candidates. Two such mechanisms are proposed and analyzed in this thesis.

The first one, which will be referred to SBAC (SINR Based Access Control), roughly consists in admitting a new connection if its own SINR as well as that of each already active transmission are all larger than the required minimum (when taking the interference created by the new connection into account). The scheduling problem, which consists in determining a *maximal* set of simultaneous transmissions in such a way that the SINR at the receiver of each transmission is above a given threshold, is NP-complete for networks consisting of a set of transmitter-receiver pairs arbitrarily distributed in the Euclidean space [111]. SBAC can be seen as a greedy heuristic for solving this problem, where transmissions are scanned in a random order and accepted as long as the above described condition is satisfied. Other solutions are discussed in Sec. 7.1.

The second, which will be referred to as PCBA (Power Control Based Access) is based on power control: given a set of transmissions intending to access the channel, a subset is selected for which there exist feasible transmission powers such that the SINR for all of them is larger than a given threshold (for instance, all connections obtain the same SINR). It should be noted that power control is used as a way to increase the number of simultaneous transmissions and not as a mechanism for energy saving (as we shall see, PCBA is not always the best in terms of power efficiency). Power control is classical in cellular networks [112, 113]. It is also used for scheduling in such networks (see e.g. [114, 115, 116, 117, 118]). We will discuss these approaches in Sec. 7.1.

Both SBAC and PCBA guarantee a minimum prescribed rate but they differ in several aspects: SBAC assumes constant transmission power and actually provides rates larger than the required minimum, whereas PCBA adjusts power transmissions to provide exactly the prescribed rate for all the active transmissions.

The aim of this chapter is twofold. Firstly, to discuss the usefulness and the implementability of these two mechanisms and secondly, to compare the performance of these mechanisms in the context of wireless ad-hoc networks. In order to compare SBAC, PCBA and CSMA/CA within this context, we consider different topologies: a line and a two-dimensional grid and different traffic scenarios: elastic (e.g. data) and non elastic (e.g. voice) traffic. For each case, several metrics, typical of mobile ad-hoc networks, are studied; these metrics leverage rate, fairness, spatial reuse and power efficiency. The use of these metrics of course depends on the traffic scenario. For example, for data traffic, high rates are valuable, whereas for voice traffic a minimum rate level is required and anything larger is not really useful. We show that no mechanism outperforms the others in all cases. We also determine which one is best depending on the considered traffic type, the propagation model, the required SINR level, and of course the performance metric.

7.1 Related Work

In this section we present some works that we believe are relevant on the context of the proposed mechanisms. More precisely, we first present some heuristics that has been proposed as the solution of the problem of achieving the throughput capacity (i.e. the maximum rate at which data can be sent) in a wireless network which is closely related to the problem of ensuring a minimum rate for all accepted connections. Then, we present some works devoted to power control mechanisms, whose particular objective may not coincide, but they all share the common principle that more connections can be scheduled if the transmission powers are appropriately chosen.

There is a large number of works devoted to the capacity of wireless networks, but we will mention here only some of the most recent ones based on the SINR model for the interference. We claim that the network geometry plays a key role on the determination of the SINR at each receiver and so in this rate. This means that we explicitly do not include graph based models since its inefficiency has been already shown [119]. In [120] a greedy scheduling algorithm is proposed for the particular case of uniform distributed nodes in a square of unit area; an approximation ratio is calculated depending on the number of nodes and the path-loss function exponent. In [111], an algorithm valid for more general topologies is presented with an approximation ratio that depends on the network diversity (which reflects the length variation of the links to be scheduled); however this diversity can be as large as the number of nodes. The approximation ratio found in [121] is logarithmical on the ratio between the maximum and the minimum distance between two nodes on the network; as for the network diversity this quantity can be as large as the network size.

Finally, the work that seems to be the closest to SBA is [122], in which an algorithm that ensures a minimum SINR for all active transmissions is proposed whose approximation guarantee is independent of the topology of the network. This approximation factor depends on the path-loss exponent, the network space dimension and a constant appropriately chosen that accounts for the distance at which transmitters can impair an active receiver. However, this factor is not evaluated and some simple examples show that it could be large. Moreover, any of these proposals has no known decentralized incarnations and they do not take shadowing/fading into account, two aspects that we want to include here. In fact, we do not look for optimal solutions but solutions that ensure that a minimum rate is guaranteed for all the accepted connections.

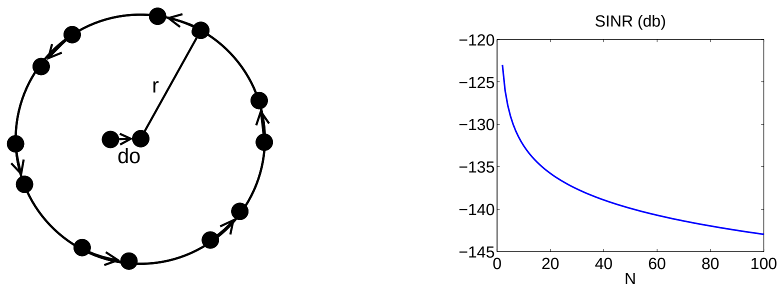
In [115] the problem of determining a minimal length schedule to satisfy links demands (i.e. SINR larger than a given threshold and no node can receive and transmit at the same time) when transmitters use optimally chosen transmission powers is addressed. Authors show that this problem is hard to solve and they find conditions under which the problem turns to be tractable. However, this condition is rarely satisfied in real networks. Moreover, the proposed solution is not decentralized. A different approach bounding routing, links scheduling and power control is assumed in [116]. The objective is to find transmission patterns such that the total average transmission power is minimized whereas each link has an average data rate equal or larger than a given threshold whereas each node has a peak transmission power bound. The problem is solved with a duality approach and from it an optimal routing is deduced. Main differences with our work are that the guarantee is over the mean rate and not each time a connection is accepted and that the fading/shadowing is not considered. Furthermore, the optimization is clearly performed off line (there is no decentralized algorithm that guides the evolution of the power transmission to the solution).

Probably the works that are the closest to PCBA are [117] and [118]. In [117], a two step scheduling mechanism is proposed. The first step consists in finding a set of "valid" simultaneous transmissions. This step is targeted to eliminate strong interferences before applying a power control mechanism and it is assumed that a central controller is responsible of its execution. The second step determines the powers needed to satisfy the SINR constraints. The main difference with our work is that we do not consider the first centralized step and we directly apply a distributed power control algorithm in order to determine the set of transmissions to be scheduled. Moreover, the fading is not considered, i.e. the attenuation is a deterministic function of the distance between nodes. In [118], the authors propose PCMA (Power Controlled Multiple Access), a wireless MAC protocol where each receiver sends busy-tone pulses to communicate its interference margin. The main idea is to preserve the paradigm of the RTS/CTS handshake by proposing an exchange of the same type refer to as RTPS/APTS: Request-Power-To-Send and Acceptable-Power-To-Send. The signal strength of the pulses is used to bound the transmission power of the interfering nodes. It is not clear however how to determine the interference margin in this context. In addition, an independent channel is required for the transmission of the busy-tones and contention between them is not addressed. However, authors show that PCMA allows for a greater number of simultaneous transmission than CSMA/CA, approximately twice.

7.2 Motivating Example

In this section we want to show, by means of a simple yet illustrative example, that accepted connections under the CSMA/CA protocol may obtain a very poor performance. As we will see, the main reason behind this situation is the fact that the channel condition is verified only on the intending transmitters, and the total interference experienced by already active receivers is neglected.

Consider a network with several short links over a circle as in Fig. 7.1(a) and a propagation environment such that these links do not detect (contend with) each other



(a)

(b)

Figure 7.1: (a) Example topology and (b) Mean SINR for the receiver located at the center of the circle.

(e.g. there are obstacles between them and the shadowing isolates them). Suppose that there is a link whose receiver is located at the center of the circle and that this link does not contend with the other links (for instance, because of distance). Finally, let us assume that the timers are such that the central link has the smallest one, then it senses the channel clear. All links are hence allowed to transmit simultaneously. We show below that the total interference created by the links located on the circle at the central receiver is possibly high.

Assume, as in the previous chapter that given two nodes, h and k, the power receiver from h by k is:

$$P(h,k) = P_h F_k^h L(h,k), \tag{7.1}$$

where P_h is the transmission power of node h, F_k^h represents the fading/shadowing from node h to node k and $L(h,k) = A\max\{r_o, d(h,k)\}^{-\alpha}$ is the path-loss function.

Let N be the number of links, P be the transmission power of each transmitter, and r be the radius of the circle. Assuming Rayleigh fading, i.e. F_i^j is exponentially distributed with parameter 1. Then, the interference I seen by the central receiver is:

$$I = \sum_{i=1}^{N} PF_i Ar^{-\alpha}.$$

F_i represents the fading between the transmitter of node i in the circle and the receiver located at the center. Under the Rayleigh fading assumption, the interference I is the sum of N independent exponential random variables, i.e. has a Gamma distribution with parameters N and $\lambda = 1/PAr^{-\alpha}$. The mean interference is then $N\lambda$, which grows linearly with N. Analogously, if we neglect the thermal noise (i.e. take $W = 0$ in (7.2)), the mean SINR at the central receiver is:

$$\overline{\text{SINR}} = \mathbf{E}\left(\frac{PAd_0^{-\alpha}}{\sum_{i=1}^{N} PF_i Ar^{-\alpha}}\right) = \frac{PAd_0^{-\alpha}}{PAr^{-\alpha}}\mathbf{E}\left(\frac{1}{\sum_{i=1}^{N} F_i}\right) = \frac{1}{N-1}\left(\frac{d_0}{r}\right)^{-\alpha},$$

where d_0 is the distance between the central transmitter and receiver. The results follows from the observation that $1/\sum_{i=1}^{N} F_i$ has a inverse gamma distribution with parameters N and 1, thus its expected value is simply $1/(N-1)$.

For the parameter setting of Sec. 7.4 with $r = 100$ and $d_0 = 1$, Fig. 7.1(b) shows $\overline{\mathrm{SINR}}$ as a function of N. As we can observe, a few links on the circle are enough to lead to a very poor SINR for the central link. This toy example shows clearly why CSMA/CA, even when augmented by the RTS/CTS handshake, cannot guarantee any performance because it is based on pairwise exclusions only. Hence the need for protocols that take the interference created by all nodes in the network into account to decide which transmission can access the channel.

7.3 Proposed Mechanisms

Motivated by the example in the previous section, we now consider admission mechanisms based on the total interference resulting from the active transmissions. More precisely, we define two mechanisms guaranteeing a *minimum rate* (equivalently SINR) for all accepted connections. Before describing the proposed mechanisms, let us reintroduce some notation. Let P_i be the power transmission of transmitter T_i to its receiver R_i. The SINR of an active link i is then:

$$\mathrm{SINR}_i = \frac{P_i F_{R_i}^{T_i} L(T_i, R_i)}{W + \sum\limits_{j \neq i} P_j F_{R_i}^{T_j} L(T_j, R_i)}, \tag{7.2}$$

where W is the thermal noise, which is considered constant and equal for all nodes. Let $A \in \mathcal{M}_{L\times L}$ be the *gain matrix*:

$$A_{ij} = \begin{cases} 0 & \text{if } i = j \\ \frac{\tau a_{ij}}{a_{ii}} & \text{if } i \neq j \end{cases}, \tag{7.3}$$

where τ is the target SINR and $a_{ij} = F_{R_i}^{T_j} L(T_j, R_i)$.

We will consider the slotted version of CSMA/CA (with the CCA in carrier sensing mode), in which all transmissions start and finish at the same time. The selection of the set of active transmissions is random and is the same, in law, at each time slot, but independent from time slot to time slot. At the beginning of the slot, the order at which each node tries to access the channel is decided randomly (for example, by using a timer). At its due turn, each transmitter/receiver pair decides whether or not to become active based on the corresponding access mechanism. It must be noted that the assumed slotted division of time prevents all algorithms from creating unfairness (i.e. the well known "starvation phenomenon" [21]).

Finally, let us note that the mechanisms that we introduce in what follows are still valid if specific performance levels are required for each transmissions; it is enough to replace τ by τ_i. This can be useful, since different values of τ_i can be associated with different service levels.

7.3.1 SINR Based Access Control (SBAC)

In SBAC, power is constant and equal to P at each transmitter. A new connection is accepted if and only if the SINR it obtains (which depends on the connections already accepted) is larger than the target threshold and, at the same time, the new SINR that the already active transmissions obtain, when taking this new connection into account, is also larger than the threshold. More precisely we want $\mathrm{SINR}_i \geq \tau$ for all active connections, i.e.

$$\mathrm{SINR}_i = \frac{PF_{R_i}^{T_i} L(T_i, R_i)}{W + \sum\limits_{j \neq i} PF_{R_i}^{T_j} L(T_j, R_i)} \geq \tau.$$

In terms of the gain matrix, the condition is:

$$\frac{\tau W}{Pa_{ii}} + \tau \sum_{j \neq i} \frac{a_{ij}}{a_{ii}} \leq 1. \tag{7.4}$$

If $W = 0$ or negligible with respect to Pa_{ii}, then the previous condition is that the sum of all rows of matrix A are less than 1, i.e. the matrix A is sub-stochastic. If $W \neq 0$, the exact condition is that the sum of all rows must be less than 1 minus a term that depends on each link:

$$\tau \sum_{j \neq i} \frac{a_{ij}}{a_{ii}} \leq 1 - \frac{\tau W}{Pa_{ii}}.$$

The dynamic of the algorithm is as follows. The first connection i attempting to access the channel is accepted if $\tau W / Pa_{ii} \leq 1$ (the second term of the left part of (7.4) is zero). For the second one, if the matrix $A \in \mathbb{R}^2$ associated with the pair is sub-stochastic in the sense given above, then the connection is accepted, otherwise it is rejected. For each new connection attempting to access the channel, its admission depends on the sum of the rows of the matrix A associated with the already accepted connections and this new one. Possible ways to implement this in a decentralized way are discussed in Sec. 7.5.

Clearly the set of accepted connections depends on the order in which the nodes attempt to access the channel. This order is assumed random. It must be noted that it is not the objective of this algorithm to maximize the number of accepted connections, since to do that, global knowledge is required. Instead, we preferred to keep our algorithm as simple as possible. As we shall see, even with this first version, substantial improvement over classical CSMA/CA is obtained.

We are aware that distributed implementations of this algorithm are not trivial due to the necessary exchange of information (possible ways to implement it in the slotted case are discussed in Sec. 7.5). However, we believe that the comparison with this algorithm is still relevant since it can be considered as the “best” solution in the analyzed context.

7.3.2 Power Control Based Access (PCBA)

Instead of the simple verification that the SINR is acceptable, we will suppose now that transmission power can be adjusted. In our second proposal, connections will be accepted as long as there exists a feasible power vector (i.e. the power of the already active connections plus the new one) guaranteeing that the SINR obtained for all accepted transmission is larger than the minimum. This condition, $\text{SINR}_i \geq \tau \ \forall i$, can be written as:

$$P_i \geq \frac{\tau W}{a_{ii}} + \tau \sum_{j \neq i} P_j \frac{a_{ij}}{a_{ii}} = \frac{\tau W}{a_{ii}} + \sum_{j \neq i} P_j A_{ij},$$

which can be expressed in a vectorial form by defining the vectors of transmission powers P and the vector $\eta \in \mathbb{R}^L$ with entries $\eta_i = \tau W / a_{ii}$, as follows:

$$P \geq AP + \eta \tag{7.5}$$

It can be proved that this inequality has a positive and finite solution if the spectral radius (maximal eigenvalue) of A is smaller than 1 (see for instance [123]). We will see how this solution can be obtained by means of a distributed algorithm. In particular, our second mechanism will be based on the distributed power control introduced by Foschini *et al.* in [112]. In fact, Foschini's algorithm ensures that the SINR obtained by all accepted connections is equal to the target SINR.

Foschini's algorithm

In [112], authors propose a simple class of power control algorithms that seek to determine as quick as possible whether or not a set of links can achieve a common SINR level τ. Ideally, a continuous dynamic for the power evolution will be given by a differential equation of type:

$$\dot{S}_i(t) = -\beta(S_i(t) - \tau), \tag{7.6}$$

where $S_i(t)$ is the SINR at the receiver of link i at time t and β is a positive proportionality constant. This dynamic cannot stop unless $S_i(t) = \tau$ for all i. However, it cannot be implemented in a distributed manner since one link (or user) has no direct control over the transmission power of the other users. Then, a dynamic that depends only on local measurements is required. To achieve this, the authors of [112] propose a simplification in which each user adjust its transmission power as if its interference does not change. Then, the derivative at the right part of (7.6) can be calculated and the following equation is obtained:

$$\dot{S}_i(t) = \frac{a_{ii}\dot{P}_i(t)}{I_i(t)} = -\beta(S_i(t) - \tau),$$

where $I_i(t)$ is the interference seen by user i at time t. This equation is equivalent to:

$$\dot{P}_i(t) = -\beta B\left(P_i(t) - \tau\frac{I_i(t)}{a_{ii}}\right).$$

Which can be rewritten in a vectorial form by defining the matrix $B = I - A$ i.e. $B_{ij} = -A_{ij} = -\frac{\tau a_{ij}}{a_{ii}}$ if $i \neq j$ and $B_{ii} = 1$:

$$\dot{P}(t) = -\beta P(t) + \beta\eta. \tag{7.7}$$

In a discrete time setting, let $P(k)$ be the power of vectors at time $k = 0, 1, 2, \dots,$ where the time coordinate is defined such that the unit time is the time between consecutive iterations of the algorithm. Then, a discrete version of (7.7) can be defined as.

$$P(k+1) - P(k) = -\beta BP(k) + \beta\eta,$$

which can be rewritten in turn as:

$$P_i(k+1) = (1-\beta)P_i(k)\left[1 + \left(\frac{\beta}{1-\beta}\right)\left(\frac{\tau}{S_i(k)}\right)\right], \tag{7.8}$$

This is the basis of a very efficient distributed scheme. In [112], it is proved that, if the spectral radius of A is less than 1 and $\beta \leq 1$, then $P(k)$ converges to P^* starting from any initial vector $P(0)$, where P^* is the smallest solution of (7.5) (i.e. if P is a solution then $P \geq P^*$).

This algorithm can be used to decide whether a new link can access the channel. For each new connection attempting to access the channel, admission depends on the spectral radius of the gain matrix A associated with the already accepted connections and the new one. If this spectral radius is less than 1, then the new connection can be accepted. If it is larger than 1, then it should be rejected. A natural incarnation of the algorithm is that where a set of active connections is first built (using a random scanning of the connections and admitting/rejecting them based on this spectral radius criterion) and (7.8) is then performed to obtain the power vector P. A more efficient incarnation is discussed in Sec. 7.5. For this power vector, the SINR obtained by all actives nodes is τ.

7.4 Comparison Results

We have proposed two mechanisms that guarantee a minimum SINR for all connections. A natural question is how they perform in terms of other metrics, such as the number of accepted connections. It could be the case that their strict requirement on the SINR translates into high rejection of the intending connections. For instance, in the example of Sec. 7.2, SBAC will admit only the central transmission. In this section, we will compare the proposed mechanisms, SBAC and PCBA, together with classic CSMA/CA, considering different network scenarios and performance metrics.

As in the previous chapter we will assume saturated traffic conditions (each node always has data to send). Under this condition, a very important performance metric in the wireless ad-hoc setting is the number of simultaneous transmission that can be scheduled by the protocol. For this, we evaluate spatial reuse, which we remember that it is defined as the mean proportion of links which are active at a typical time slot. However, as already explained, the accepted connections may obtain a very poor quality. We hence also measure the rate obtained by each of them. Since there is a clear tradeoff between spatial reuse and rate, we also define several different utilities to assess the overall performance of each algorithm. We also compare how fair is the rate distribution. A special emphasis is put on the comparison of the two mechanisms with CSMA/CA. More precisely, we will use the RED model introduced and analyzed in the previous chapter (c.f. Sec. 5.3) for comparison purposes; so that when we refer to CSMA results they actually correspond to the RED model.

Two different topologies are considered: regular lattices in $\mathbb{R}$ (line) and in $\mathbb{R}^2$ (grid), with a distance d between two neighbor nodes. It is also assumed that each node can transmit or receive and that it communicates with its nearest neighbors. We fix $L = 100$ nodes for both topologies, distributed in a lattice of 10×10 for the grid. As in the previous chapter, we use $A = -53$dB, $r_0 = 0.01$ and $\alpha \in \{2.5, 3, 4\}$. Finally we fix $P = 2.3$dBm and $W = -96$dBm for all nodes. Concerning the channel model, as before we analyze two different models: Rayleigh fading and Lognormal shadowing. The random variables $\{F_i^j\}_{i,j}$ are independent and exponentially distributed with parameter $\mu = 1$ in the first case, and Lognormally distributed with standard deviation $\sigma = 4$dB in the second one.

The results of this section are obtained mainly by simulations where each algorithm is performed $N = 1000$ times, each time representing a slot. At each time slot, a symmetric matrix of random numbers is constructed representing the symmetric random fading/shadowing. In each repetition (slot), the order at which the nodes intends to access the channel is random, selected according a Uniform distribution in the interval $[0, 1]$. The decision of accepting the transmission or not is taken according to the selected mechanism.

We have chosen default options — the line topology and Lognormal shadowing — and we only report on results for the other cases if they are illustrative (e.g. grid versus line topology) or if the differences are significant. For example, for a given channel model the comparison results obtained for both topologies are quite similar. However, for a given topology, the channel model may have a significant impact.

7.4.1 Spatial Reuse and Mean Rate

In this section, we compare the spatial reuse (SR) and the mean rate (MR) obtained by each mechanism. Let $\mathbf{1}_r(i)$ be an indicator function that takes the value 1 when link i is active during the time slot r and 0 otherwise. The considered indicators are

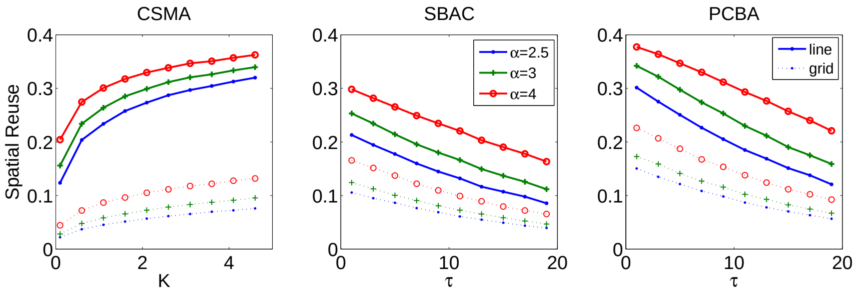


Figure 7.2: Spatial reuse comparison for the line topology (solid lines) and the grid topology (dotted lines) with Lognormal shadowing.

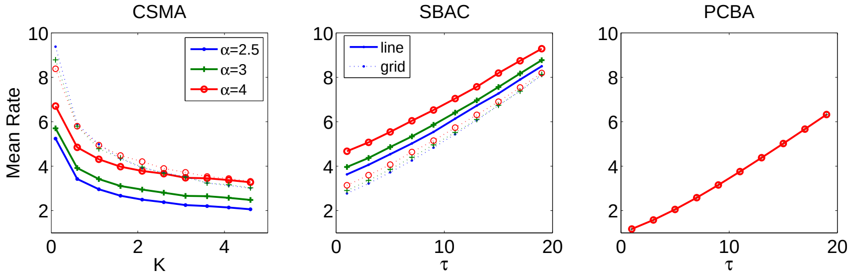


Figure 7.3: Mean rate comparison ($\rho = \log_2(1 + \text{SINR})$) for the line topology (solid lines) and the grid topology (dotted lines) with Lognormal shadowing.

defined as:

$$SR = \frac{1}{L}\sum_{i=1}^{L} p_i \quad \text{where} \quad p_i = \frac{1}{N}\sum_{r=1}^{N} 1_r(i) = \frac{1}{N}N_i,$$

$$MR = \frac{1}{L}\sum_{i=1}^{L} \rho_i \quad \text{where} \quad \rho_i = \frac{1}{N_i}\sum_{r=1}^{N} 1_r(i)\log_2(1 + \text{SINR}_r(i)). \tag{7.9}$$

Note that N_i is the number of time slots that link i accessed the channel, and $\text{SINR}_r(i)$ is the SINR obtained in the r-th slot (the rate was calculated as $\rho = \log_2(1 + \text{SINR})$). For the SINR-based algorithms, the minimum required SINR is $\tau \in \{1, \dots, 20\}$dB.

Results for the SR and the MR are reported in Fig. 7.2 and Fig. 7.3 respectively. Please note that for the same value of α, the spatial reuse increases (decreases) with K (τ). Small values of K correspond to large values of τ: in both cases, less connections are accepted. When K decreases, the number of contenders increases, resulting in lower access probability for each user. Analogously, large of values of τ translate into higher SINR requirements, thus reducing the possibility of accepting a new connection. Finally, before presenting the results, let us highlight that for PCBA the mean rate does not depend on α nor on the topology, since it is equal to $\log_2(1 + \tau)$ in any case.

For each value of τ, PCBA has higher SR than SBAC. However, its MR is smaller since, as we mentioned before, the rate obtained by all connections in SBAC is larger than the target minimum. On the other hand, CSMA achieves similar levels of SR and MR as PCBA. These observations are valid for both considered topologies. However, PCBA guarantees a minimum rate for all accepted transmissions. Consider for example $\alpha = 3$, where the maximum SR obtained by CSMA is 0.34. For the value of τ where PCBA obtains the same SR ($\tau = 1$), 44% of the connections accepted by CSMA obtain a SINR smaller than τ. On the other hand, when CSMA is compared with SBAC, it can be seen that the former can obtain levels of SR that the latter does not. Hence, if only the rate is considered, SBAC is the best solution. For the same SR level, a large number of CSMA connections obtain smaller SINR than the one guaranteed by SBAC. Then, for the same number of active links, better conditions will be obtained with PCBA or SBAC. These results are in accordance with our conjecture that substantial improvement is possible by means of SINR-aware algorithms.

Regarding the impact of the topology on the SR and MR, it can be seen that the comparative analysis between the proposed mechanisms and CSMA (equivalently RED) is similar for both topologies. Differences can be observed mainly in the SR, which is smaller for the grid topology due to the increase of potential interferers: in the line topology each "no border" node has two nearest neighbors, that become four in the grid one. However, the impact on the MR is less significative. The lower number of simultaneous active connections results in rate levels similar to those obtained on the line.

There is a clear tradeoff between MR and SR. For instance, if we choose the value of K that maximizes the SR we will obtain a small MR. On the contrary, if we choose K so as to maximize the MR we obtain a very poor SR. Which is then a good choice of K that considers both parameters at the same time? To evaluate more accurately this tradeoff, we will consider several utility functions, depending on the type of traffic present on the network. In particular we will concentrate in three types of traffic: Elastic Traffic (e.g. data), Elastic Traffic with minimum required SINR (what one would very much appreciate for data traffic in heavily loaded wireless LANs) and Constant Bit Rate (CBR) (e.g. voice traffic).

7.4.2 Elastic Traffic

For a given algorithm, link i has an access probability p_i and the mean rate obtained is ρ_i (see (7.9)). For elastic traffic, for which the importance is on the amount of information that can be send per time unit, a good performance indicator for link i is the product $p_i\rho_i$. To measure the overall performance, we will consider the following utility functions:

$$U_0(x) = x, \quad U_1(x) = \log(x), \quad \text{and} \quad U_2(x) = -\frac{1}{x}.$$

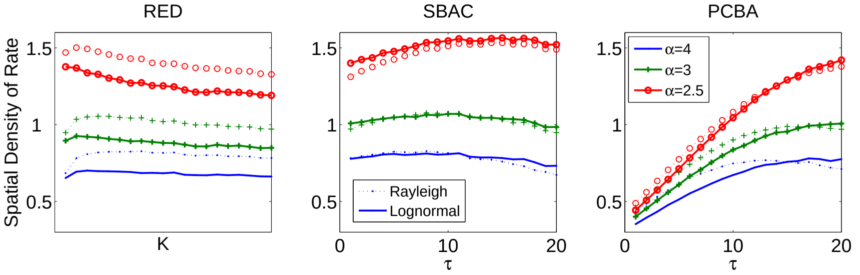


Figure 7.4: Spatial Density of Rate (U_0) for the line topology with Rayleigh fading (solid lines) and Lognormal shadowing (dotted lines).

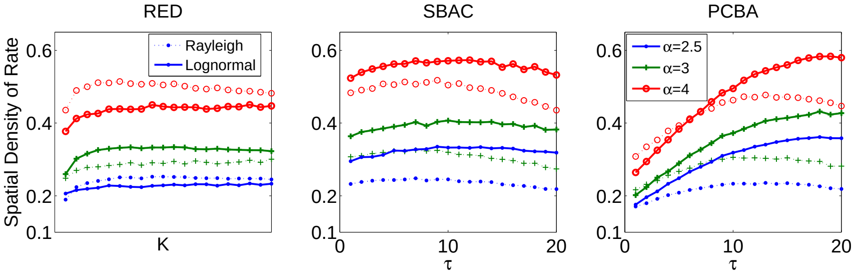


Figure 7.5: Spatial Density of Rate (U_0) for the grid topology with Rayleigh fading (solid lines) and Lognormal shadowing (dotted lines).

Considering the average of the users (links) utilities, we obtain the following performance indicators:

$$U_0 = \frac{1}{L}\sum_{i=1}^{L} p_i\rho_i, \quad U_1 = \frac{1}{L}\sum_{i=1}^{L}\log(p_i\rho_i), \quad \text{and} \quad U_2 = \frac{1}{L}\sum_{i=1}^{L} -\frac{1}{p_i\rho_i}.$$

U_0 may be interpreted as the spatial density of rate; a larger value of this function indicates a bigger total throughput. However, U_0 does not consider fairness among users, i.e. how is this total throughput distributed. U_1 is precisely a measure of such fairness (in the proportional sense). Finally, U_2 may be seen as a negative delay. Our simulations indicated that similar conclusions are obtained for the three utilities. As such, we will only show the results for U_0.

Fig. 7.4 shows the obtained results for the line topology with Lognormal shadowing (solid lines) and Rayleigh fading (dotted line). Corresponding results for the grid topology are reported in Fig. 7.5. Observe that for the CSMA, the values of K are not indicated since they should be interpreted differently depending on the channel model (see Sec. 5.4.1). In any case, values of K were chosen in order to obtain a similar mean number of contenders.

If we focus on the Lognormal shadowing (solid lines), and we concentrate on the maximum value obtained by each algorithm, we find that for both topologies and for all values of α, SBAC and PCBA obtain better results than CSMA. Moreover, SBAC outperforms PCBA for almost all values of τ; obtaining also the maximum for the line topology. For the grid topology the maximum achieved by PCBA (obtained for a high value of τ) is slightly larger than the one obtained by SBAC; however as we may see in the figure, the difference is almost negligible.

As mentioned before, in general, our results do not change significantly with the fading/shadowing distribution. However, this is not entirely the case for this metric. If we focus now on the results for the Rayleigh fading (dotted lines), we may see that results obtained by CSMA have improved with respect to the other mechanisms; obtaining similar results than SBAC specially for small values of α. In contrast, the results for the other mechanisms are in general worsened, specially for the grid topology and SBAC (for the line topology, results remains essentially the same).

More in detail, we observe that the results obtained by CSMA and SBAC are almost constant in K/τ (for a given value of α), whereas those obtained by PCBA strongly depend on τ. This effect is a direct consequence of the fact that for each value of τ, all connections accepted under PCBA obtain exactly the same SINR and no more (which is not the case for SBAC), thus limiting its performance when the spatial density of rate is considered; specially for small values of τ. The comparison between SBAC and CSMA is favourable to the former for all values of τ and K.

Summing up, the best results are always obtained by a SINR-based mechanism, but which of the two is the best depends on the topology and the channel model. If Rayleigh fading is assumed, for both topologies SBAC obtains the best results but the differences with CSMA are not significant. This means that a similar tradeoff between spatial reuse and mean rate as for CSMA can be obtained by SBAC with the added improvement that a minimum SINR is guaranteed for the accepted transmissions. For the Lognormal shadowing, SBAC and PCBA obtains very similar results, being the former slightly larger for the line topology and the inverse for the grid one.

Finally, we will consider fairness in the obtained rates in more detail. Even if U_1 already takes this into account, we also study the Jain's index, defined as follows:

$$FI_{\text{rate}} = \frac{\left(\sum_{i=1}^{L} \rho_i\right)^2}{L \sum_{i=1}^{L} \rho_i^2}.$$

The rationale behind this index is that the perfect rate distribution is one in which all rates are the same. The index measures the difference between the obtained rate distribution and the perfect one. It ranges from $1/L$ corresponding to the worst case, to 1 corresponding to the ideal rate distribution.

In Fig. 7.6 we report on the results obtained by calculating the fairness index for each slot and averaging these values. Since with PCBA all transmissions obtain the same SINR, this mechanism obtains the maximum fairness index independently of the value of α and τ (i.e. $FI_{\text{rate}} = 1$). As we observe, the results for the SINR-based algorithms largely outperform CSMA. This means that the rate distribution is more

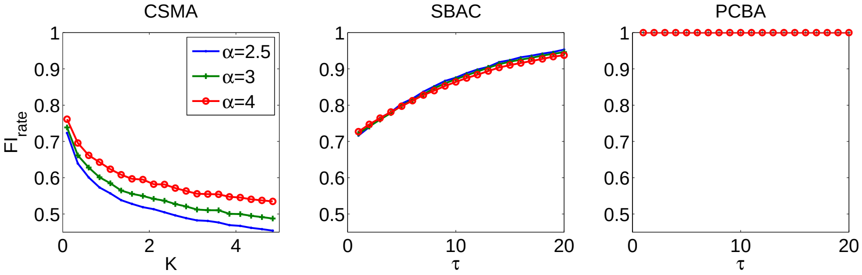


Figure 7.6: Mean Rate Jain's index FI_{rate}.

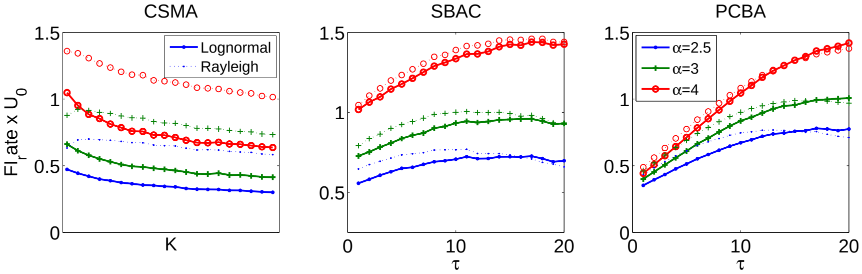


Figure 7.7: Product of the FI_{rate} and the Spatial Density of Rate (U_0) for the line topology with Lognormal shadowing (solid lines) and Rayleigh fading (dotted lines).

fair in each of the proposed mechanisms. Both proposed mechanisms can then achieve better or at least equal spatial density of rate but with a much more fair distribution of the rate.

Let us now consider both the spatial density of rate and the fairness index. For this we will weight U_0 by FI_{rate}. Results are shown in Fig. 7.7 for the line topology with Lognormal shadowing (solid lines) and Rayleigh fading (dotted lines). Note that, due to its perfect index PCBA now outperforms SBAC for $\alpha = 2.5$ and 3. On the other hand, CSMA suffers from the unfairness in its rate distribution; for instance if the Lognormal case is considered, the maximum value obtained by CSMA is smaller than the minimum obtained by SBAC. These results reinforce our previous conclusions about the advantages of the proposed mechanisms.

7.4.3 Elastic Traffic with Minimum Required SINR

The main difference between the considered algorithms is the guarantee or not of a minimum SINR. In this section we want to quantify this difference; for this, we define metrics that penalize situations where an active connection obtains a SINR smaller than the target minimum. Consider the following modified access probability and rate:

$$p_i^\tau = \frac{1}{N_i} \sum_{r=1}^{N_i} \mathbf{1}_r(i) \mathbf{1}_{\{\text{SINR}_r(i) \geq \tau\}},$$

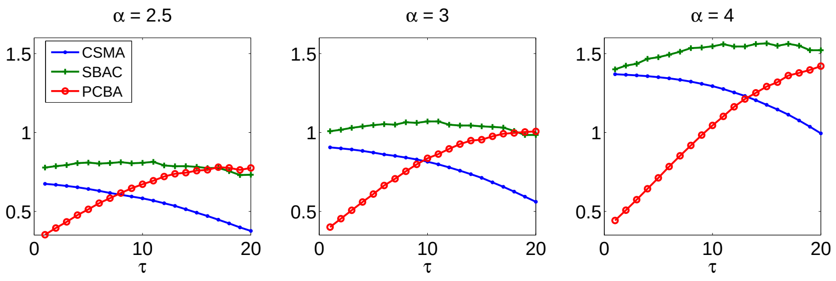


Figure 7.8: Spatial Density of Rate when SINR $\geq \tau$ (U_0^τ) for the line topology with Lognormal shadowing.

$$\rho_i^\tau = \frac{1}{N_i} \sum_{r=1}^{N} \mathbf{1}_r(i) \mathbf{1}_{\{\mathrm{SINR}_r(i) \geq \tau\}} \log_2(1 + \mathrm{SINR}_r(i)),$$

The first definition represents the probability to access the channel with a SINR larger than τ. Analogously, ρ_i^τ is the mean rate of these connections. The comparison metrics are:

(i) $U_0^\tau(x) = \frac{1}{L} \sum_{i=1}^{L} p_i^\tau \rho_i^\tau$: a modified spatial density of rate,

(ii) $SR^\tau = \sum_{i=1}^{L} p_i^\tau$: the SR but considering only those connections whose SINR is larger than τ.

Clearly, for SBAC and PCBA, these metrics coincide with that previously calculated (SR and U_0) since, by definition, a minimum is guaranteed for each connection. Yet, results for all mechanisms are reported to ease the comparison.

The results for the modified spatial density of rate are shown in Fig. 7.8, for different values of τ. For CSMA, we report on the maximum value since it depends on K. Note that for every value of τ, the best result is obtained by one of the SINR-based mechanisms. More precisely, SBAC essentially provides the best results, although it is in some cases slightly worse than PCBA (e.g. for $\alpha = 2.5$ and 3 and large values of τ). In fact, the difference between SBAC and PCBA decreases with τ and increases with α. The relative gain of SBAC over PCBA is mainly due to the SBAC property of guaranteeing a minimum SINR by actually providing more than the required minimum, from which this metric takes advantage (even if the same number of connections is accepted for both mechanisms they will obtain more rate in SBAC). Regarding the comparison between CSMA and PCBA, we see that, for the former, U_0^τ decreases with τ, whereas, for the latter, it increases. Actually, for small values of τ, CSMA outperforms PCBA, whereas it is exactly the contrary for large values of τ. This situation can be explained by the fact that for small values of τ, most of the connections accepted by CSMA obtain a MR that exceed these values (see Fig. 7.3). But, as we will see in what follows, the transmission power required for both mechanism can be very different.

We have observed that for a given channel model, the comparison results (in terms of U_0^τ) are the same for both considered topologies. Then, to complete the analysis we

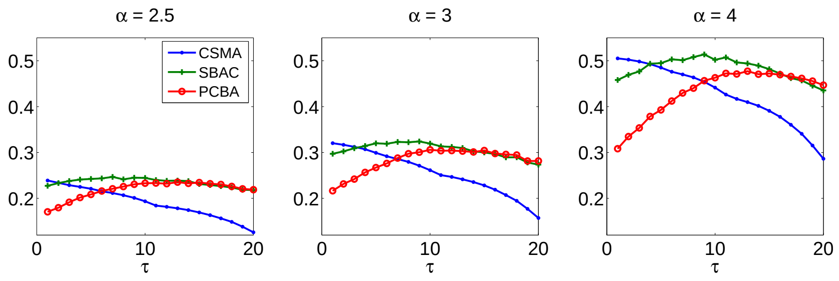


Figure 7.9: Spatial Density of Rate when SINR $\geq \tau$ (U_0^τ) for the grid topology with Rayleigh fading.

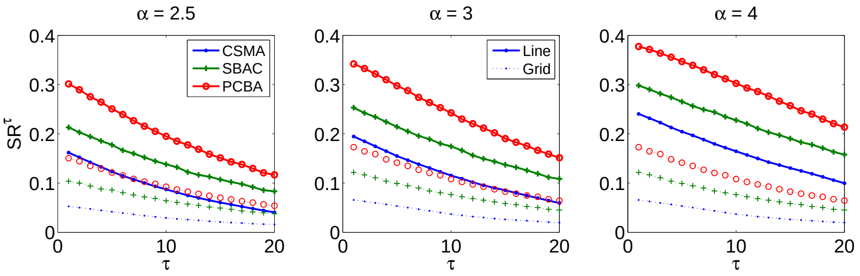


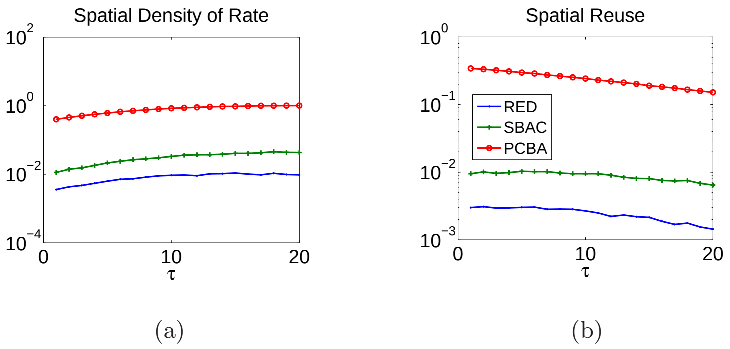
Figure 7.10: Spatial Reuse when SINR $\geq \tau$ for the line and grid topology with Log-normal shadowing.

report on the results for the grid topology with Rayleigh fading. In this case, the results are slightly different. In Fig. 7.9, we see that the performance obtained by the three mechanisms are now more similar than on the line. In particular, for some small values of τ, CSMA is the mechanism that obtains the best results, although the difference is not significant. The reason behind this behaviour is the same as before.

We now concentrate on the modified spatial reuse. In Fig. 7.10, we may see that PCBA outperforms the rest of the mechanisms for all values of τ and α, irrespectively of the topology and the channel model. We have already seen that PCBA obtains larger spatial reuse than SBAC (see Fig. 7.2) for all values of α and τ; and the condition that the SINR must be larger that τ makes that several of the connections accepted by CSMA are not taken into account anymore, decreasing its spatial reuse. This means that, if the target is to guarantee a certain minimum SINR (independently of the particular level), SBAC and PCBA can accommodate more connections. PCBA is the one that obtains the best results since for the same given set of links it can adjust the transmission powers to accept more connections.

7.4.4 Constant Bit Rate (CBR)

We consider now a different kind of traffic: Constant Bit Rate. This type of traffic needs a certain rate level and obtaining more than the required level is without value (e.g.



(a) (b)

Figure 7.11: Spatial Density of Rate (a) and Spatial Reuse (b) with SINR $= \tau$ for the line topology with Lognormal shadowing and $\alpha = 3$.

voice traffic). To evaluate the performance of the algorithms in the presence of such traffic, we consider the same metrics as before but imposing that the SINR is "equal" to a certain threshold τ. In this case, it is important to guarantee a minimum rate, but situations where much more larger values than the required minimum are obtained will be simply ignored. In fact we consider an interval of values near τ, since if we consider values exactly equal to τ, no algorithm will make it (due to numerical variations on the obtained SINR). In particular, we define the interval $I = [0.99^{*}\tau, 1.05^{*}\tau]$. Observe that the interval is not symmetric, smaller values than τ are more penalized than larger ones. More precisely, we consider SR^{τ} and U_0^{τ} but replacing p_i^{τ} and ρ_i^{τ} by:

$$p_i^{\tau} = \frac{1}{N_i}\sum_{r=1}^{N_i} \mathbf{1}_r(i)\mathbf{1}_{\{\mathrm{SINR}_r(i)\in I\}},$$

$$\rho_i^{\tau} = \frac{1}{N_i}\sum_{r=1}^{N_i} \mathbf{1}_r(i)\mathbf{1}_{\{\mathrm{SINR}_r(i)\in I\}}\log_2(1+\mathrm{SINR}_r(i)).$$

We evaluate the same metrics as in the previous section, but replacing $\mathbf{1}_{\{\mathrm{SINR}_r(i)\geq\tau\}}$ by $\mathbf{1}_{\{\mathrm{SINR}_r(i)\in I\}}$. In this case, the results coincide for both metrics and without surprise, PCBA provides (by far) the best results for all values of α and τ. We report results for $\alpha = 3$ in Fig. 7.11, where the y-axis is in log-scale to highlight the differences. It must be noted that CSMA obtains a very poor performance when this metric is evaluated (we have already seen that the rate fairness index can be very low, see Fig. 7.6). SBAC obtains intermediate results; they are largely better than CSMA's but still far off the very good results obtained by PCBA. It is not surprising that for this metric, the difference between SBAC and PCBA increases, since this metric prioritizes more equally distributed rates; and as we have already seen the actual rate obtained by SBAC is larger than the minimum required (see Fig. 7.3). The conclusions are valid for both topologies and both analyzed channel models.

7.4.5 Rate vs Transmission Power

A very important aspect in MANETs is power consumption, which is also a main difference between the analyzed access mechanisms. It must be noted that PCBA was not designed with power saving in mind, but to allow a larger number of simultaneous transmissions. In this section, we analyze the ratio between the mean rate and the required transmission power, i.e. how many bits per second can be transmitted with one power unit.

We define the following metric that takes into account the relation between rate and power each time a transmission takes place:

$$U^p = \frac{1}{L}\sum_{i=1}^{L}\frac{1}{N_i}\left(\sum_{r=1}^{N_i}\frac{\rho_i(r)}{P_i(r)}\mathbf{1}_r(i)\right) = \frac{1}{L}\sum_{i=1}^{L}\overline{\left(\frac{\rho_i}{P_i}\right)},$$

where $P_i(r)$ is the power of link i in slot r. Since for PCBA, the rate is constant and equal to $R = \log_2(1+\tau)$ and since for the rest of the algorithms, the transmission power is always constant and equal to P, the metric becomes respectively:

$$U^p_{pcba} = R\frac{1}{L}\sum_{i=1}^{L}\overline{\left(\frac{1}{P_i}\right)} \quad \text{and} \quad U^p = \frac{1}{LP}\sum_{i=1}^{L}\overline{\rho_i}.$$

Note that the comparison between CSMA and SBAC is the same as presented in Sec. 7.4.1 since the considered metric is simply the mean rate weighted by a constant.

The transmission power required to obtain the target level of SINR with PCBA depends on the value of d (distance between nodes). Results are shown in Fig. 7.12 for $d \in \{1, 10, 100\}$ and for different values of α. Clearly the required power increases with τ and α. From Fig. 7.12, we may conclude that for $d = 1$, PCBA is the best results in terms of the previously defined metric (the mean power is orders of magnitude smaller than the constant power $P = 10^{2.3}$mW assumed for the rest of the mechanisms). Fig. 7.13 reports on the results for $d = 10$. To ease the comparison, we plot all the algorithms together although they depend on different parameters: for CSMA the x-axis must be understood as K, whereas for SBAC and PCBA it must be understood as τ. As expected, the mechanism which obtains the best results is PCBA. However, the difference decreases with τ and α since in both cases the transmission power increases.

Furthermore, it can be expected that as d increases (and so the power required by PCBA) the gain of PCBA over the rest of the mechanisms will decrease. If P is the minimum power required to achieve the target SINR with $d = 1$, the corresponding value when $d \neq 1$ is $P' = Pd^{\alpha}$ (see Fig. 7.12). Then, for large values of d (e.g. $d = 100$), the performance of PCBA in terms of the rate/power relation decreases, specially for high values of α (see the most right part of Fig. 7.12). However, this parameter also impacts over the other mechanisms. For instance, when d increases, for CSMA there is no difference in the SR, but the MR decreases. SBAC experiences a decrease in the SR (less connections can be accepted). PCBA maintains its SR and MR, but at the expense

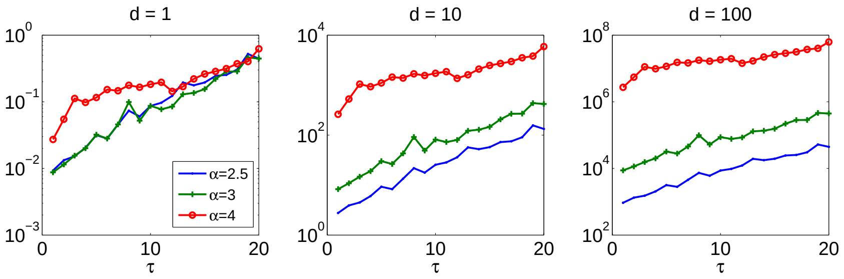


Figure 7.12: Transmission power obtained by PCBA (mW).

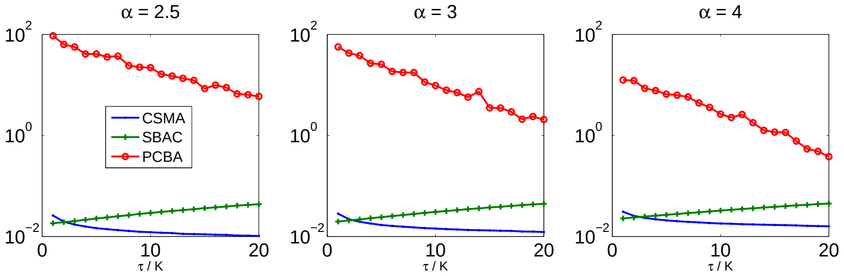


Figure 7.13: Comparison of U^p for all mechanisms with $d = 10$, for the line topology with Lognormal shadowing.

of an extreme increase of the power transmission. More precisely, if $d = 100$ and $\alpha = 2.5$ or $\alpha = 3$, for both the line and the grid topology under the Lognormal assumption, PCBA is still the mechanism that obtains the best results with this metric. For the other metrics previously analyzed an increase of d translates into larger differences in favour of the SINR-based mechanisms.

7.5 Implementation Issues

In this section we discuss which are the main implementation issues of the proposed mechanisms and we present possible solutions. Remember that our main constraint to the proposed solutions, is that they must be fully decentralized.

SBAC

In this algorithm, a transmission intending to access the channel must evaluate its SINR, and at the same time the already active transmissions must verify that the new SINR they will experience (if the new transmission is accepted) will still be larger than the target minimum. A possible solution is that the intending node sends a probing signal to the rest of the nodes, to give them the information necessary to evaluate their new SINR. In case a connection sees that its new SINR is not acceptable, this information must be sent back to the original node to cancel its transmission. However, it is hard to imagine a simple way to achieve this exchange of information. If we assume that is enough to consider only the maximum interferer the exchange of information is limited and can be feasible, but work is required to validate this assumption.

Alternatively, we may think the problem in the following reverse sense: all nodes intending to send data are active (in particular sending "HELLO" messages to their destinations) and they are *inactivated* in random order. When its turn comes, the tagged node stops transmitting and starts "listening". If it receives an ACK from its receiver, it means that the SINR is enough to successfully receive the data and the connection is activated. If this ACK is not received after a certain time, the node will start to listen for the HELLO probes, i.e. it verifies if it is not the intending destination of another node. If it receives such packets, it answers with an ACK when they stop. Note that after all nodes have stopped sending HELLO messages the resulting active transmissions will obtain a minimum SINR level, enough to correctly decode data. In order to obtain an arbitrary minimum SINR level, each receiving node must estimate its SINR and answer with an ACK only if the estimation is larger than the required minimum.

PCBA

As mentioned before, PCBA can be implemented in a totally distributed way. In the algorithm introduced by Foschini and described in Sec. 7.3.2 the transmission power evolves according to (7.8) which depends only on local measurements. For real implementations, the two step decision described before - first decide which transmission are feasible and then calculate the corresponding power - is not realistic. In place, a useful property of this algorithm can be used: if the feasibility condition is not satisfied, it diverges at an exponential rate [124]. Then, to decide if a new transmission can be accepted or not, some iterations of (7.8) are performed and if there is divergence, the connection is rejected; in other case it is accepted with the power obtained after these iterations.

7.6 Conclusions

In this chapter, we analyzed some weaknesses of CSMA/CA and we proposed two decentralized multiple access mechanisms: SBAC and PCBA. The main advantage of these mechanisms is to guarantee a minimum rate for all the accepted transmissions. We compared their performance assuming different topologies, traffic scenarios and propagation models and we devoted special attention to the comparison with CSMA.

We found that in all cases, irrespectively of the topology, traffic type and/or propagation model, one of the proposed mechanisms significantly outperforms CSMA, apart from a few cases where the differences are not significant. The rate distribution is also more fair for each of the proposed mechanisms than for CSMA.

If elastic traffic is considered, SBAC is the algorithm which provides the best results. When a minimum rate is to be guaranteed, the best one depends on the considered metric, the minimum rate level and α. If the comparison is made in terms of spatial reuse, PCBA largely outperforms the rest of the mechanisms. This is due to its capacity of controlling the transmission power, thus accommodating more simultaneous connections. At the same time, since it gives exactly the same rate to all the connections, its performance decreases when other metrics (that explicitly include the rate) are considered. For example, SBAC is the mechanism which provides the best spatial density of rate.

Finally, PCBA also brings the best results when constant bit rate traffic is considered, irrespectively of the metrics and the propagation model. When the ratio between rate and transmission power is considered, again PCBA obtains very good results as long as the distance between transmitter and receiver is limited.

These results encourage us to continue with the quest of an algorithm that guarantees minimal performance for the accepted transmissions. However, much work needs to be done, specially in the practical implementation of the decentralized algorithms. Among the most important open questions let us quote the impact of a maximum transmission power on PCBA and the extension of the proposed mechanisms to more dynamic scenarios. The analysis of the non slotted version of these algorithms is undoubtedly challenging and necessary. It would be interesting for instance to check whether the starvation phenomena experienced by CSMA is still present or not for the mechanisms proposed in this work. Furthermore, other irregular topologies must be included to obtain more general results. For instance, two-dimension networks with nodes distributed according to Uniform or Poisson distribution can be analyzed.

Chapter 8

Conclusions and Future Work

Telecommunications networks are embracing heterogeneity at service as well as technological level. The organization of this thesis tagged to reflect these two aspects and so was organized in two parts. The first one was devoted to the prediction and classification of traffic. Indeed, the first problem is of great interest, for instance, for online resource management or capacity planning. However, service convergence on the same network, together with the ever-increasing access rates (e.g. Fiber To The Home), has complicated enormously this problem. This increase on the access rate has also permitted the proliferation of new resource consuming network applications. As an example of such applications we consider P2P-TV traffic. In this sense, ISP are interested in identifying traffic generated by these applications. In particular, we have addressed these two problems using a Machine Learning technique known as Support Vector Machines (SVM).

Before going into details, some general remarks are in order. Very good results were obtained by SVM in both of the considered problems. This shows the versatility of the chosen learning technique, and encourages its use in other networking problems. In addition to the accurate results, some of its main advantages are its robustness and low computational complexity. Robustness refers to the parameter selection and tuning. We have shown that even if the optimal parameter selection may be somewhat difficult, the performance of SVM under suboptimal conditions is limitedly degraded. Its computational efficiency is a consequence of the fact that the final solution depends on the number of support vectors and not on the training set size. This characteristic turns SVM specially attractive for online applications as it is our case.

In particular, for the problem of the link load prediction, two different strategies were investigated. The first one, known as "embedding process" is based on a time series of averaged load values. The SVM embedding process uses an arbitrary number of past measurement of the time series in order to predict its future value. In this case a small time scale was used. This procedure gives the most accurate results when compared with a large set of parametric and non parametric techniques, such as Moving Averages, Auto Regressive models and the Nadaraya-Watson estimator. An extensive analysis of the parameter impact was performed from which we showed the robustness of SVM mentioned above. We have also analyzed the temporal evolution of the prediction error, which could be useful at the moment of deciding when to trigger a new training phase or detecting anomalous situations. Finally, the possibility of extending the forecast horizon by recursively applying the SVM predictor was evaluated.

Despite these good results, the gain of SVM over optimized versions of some of the considered alternatives was somewhat modest. This observation motivated our second strategy. In this case, we took a different approach and predict not the mean value of the link load at short timescales, but the maximum value or a percentile during a longer time interval. Moreover this prediction is based not on raw past observations, but on a summary of statistical properties of them. In particular, we have considered as SVM output the maximum and the 95th-percentile over a one minute length interval, and as input different combinations of the mean, standard deviation, maximum and percentile, over a past interval of equal length. The impact of several parameters (such as the input/output selection, timescale or traffic breakdown) on the SVM prediction accuracy was evaluated. We also considered several real-world traffic traces, representative of very different network scenarios (such as ISP, Ethernet, WiFi LAN and enterprise networks). We showed that this strategy performs better than other techniques, including the “embedding process” described above. In particular the use of parallel SVM (i.e. the combination of the results obtained by several machines trained with the same output but with different inputs) appears as a very interesting alternative to obtain good results, avoiding at the same time the selection of optimal settings. Our results indicate that although we do not improve the results obtained by the optimum input choice, the performance loss is insignificant. This property transforms parallel SVM in an excellent strategy when the best input is not clear.

It is our belief that some extensions are worthwhile and that this topic is not closed with the results obtained in this thesis. For instance, the “embedding procedure” with larger time scales may be considered. In particular, it would be interesting to investigate whether feeding SVM with features such as time-of-day and day-of-week would help in forecasting periodic load fluctuations (such as lunch breaks and weekends). With respect to the parallel SVM proposed here, other ways of coupling the results of different machines are available, such as Boosting or Bagging [63, 64]. Even if these techniques are more complex, whether they can bring further improvements or not should be assessed. Finally, it would be interesting to investigate if other alternative methods based on functional attributes, that take further advantage of the time series nature of our data, could improve the performance of the methods considered in this thesis.

For the problem of identifying P2P-TV traffic, the emphasis was put on the definition of a signature able to describe the operation of each application, and that can be used to discriminate them. We showed that such signature could be simply the count of packets and bytes exchanged between peers during small time intervals (in the order of seconds). This signature (which we called Abacus) combined with the discriminatory potential of SVM, is the key of the impressive results we obtained: 95% of P2P-TV traffic correctly classified and less than 1% of false alarms when non P2P-TV traffic is considered. Our proposal was largely validated in terms of accuracy and portability, based on the use of traces gathered in a large scale testbed or collected from real operational networks. The portability of Abacus (i.e. train the machine in a specific context and test it in a very different one), was verified by testing it in different network sites, access technologies, channel popularity and disruption situation (e.g. high packet loss). Furthermore, Abacus proved to be very lightweight in terms of memory consumption and computational complexity.

Regarding the Abacus framework, we have identified at least two aspects that deserve more attention. Firstly, given the simplicity of the signature (only counts packets and bytes) and the low computational cost of SVM, it is worth to analyze the inclusion of Abacus in monitoring tools such as Netflow (which is commonly deployed in operative networks). Secondly, even if the false rate alarm is very low with our rejection criterion based on the Bhattacharyya distance (less than 1%), other methods could be used. For instance, all the non P2P-TV traffic can be assimilated to a single class and the SVM could be trained accordingly (with the addition of this “unknown” class). In this way, the rejection criterion is not required anymore. This would be specially useful if large time windows are considered, since in this case most of the errors are generated by the rejection criterion.

The second part of this thesis was devoted to the the study of wireless ad-hoc networks. This was motivated by the increasing deployment of this kind of networks, and the potential new services that will be offered over them in the near future. In particular, we focused on MAC mechanisms, which are fundamental to their correct performance.

We performed a deep study of the most relevant models of CSMA/CA [21, 22, 100], which is the commonly used MAC mechanism. The first thing we noticed is that the assumptions made by these models and their correlation with the real protocol are not clear. For instance, the underlying assumptions on the CCA (Clear Channel Assessment) are not even mentioned in these references. Moreover, certain assumptions are over-simplistic, such as not considering the random variations of the channel condition. We then performed a classification of these models, making explicit in which context these models are valid, which we believe is a contribution on the understanding of the CSMA modeling.

We analyzed also the extension of some of these models to the more general case so that they include shadowing/fading effects. In this sense, we proposed an extension based on the definition of a random exclusion domain. We showed that significantly different results are obtained with this extension with respect to the original model. This shows that this aspect should be considered for modeling purposes. However, the techniques used to obtain analytical formulae in the analyzed proposals (such as the packing formalism or the subtracted singularities method) are not applicable in this more general setting.

In any case (considering or not shadowing/fading effects), these techniques only provide analytical formulae for simple cases, such as the regular line topology. Analytical results for other topologies can only be obtained, to the best of our knowledge, by means of Matérn like models. However, it is well known that these models are intrinsically conservative when compared with the real protocol. A quantitative evaluation of the magnitude of this difference was carried out for the first time in this thesis. We showed that even if the differences on the spatial reuse are significant (about 30%), when other performance metrics are considered the differences are less important (e.g. 10% for the spatial density of rate). An interesting aspect that could be worth to investigate, even if we are aware that it is of long term scope, is the possibility of using Spin Glasses techniques [101] to model the CSMA dynamics. It would be interesting to analyze if these techniques, used by physicists for the analysis of particle systems, could be adapted to this context.

One of the main aspects that remained open for us at the moment of modeling CSMA/CA was the inclusion of other CCA modes. The above mentioned models were implicitly conceived for the Carrier Detection (CD) CCA mode. In the alternative Energy Detection (ED) mode the use of exclusion domains is no longer valid, and the total interference created by all active transmissions must be evaluated (i.e. it is no longer a pairwise relation). We devoted then special attention to the modeling and comparison of these two modes. In particular, we have seen that they can be described in terms of an additive and an extremal shot noise (ASN and ESN). This observation implies that analytical results can be derived for the access probability in both cases, when the set of potential transmitters is assumed to follow a Poisson point process. Moreover, a quantification of the difference (loss) on the number of accepted transmissions when the ED is used (instead of the CD one) was provided. Interesting observations were also deduced when random or deterministic thresholds are considered (for both modes). We also evaluated the mean interference at each active transmitter, finding that the differences between the two modes can be very important, and that the ED mode is conservative enough to obtain mean interference values that are about a half of the threshold used to accept a transmission. Results on imposing a condition over the sum and the maximum at the same time were also derived, which resulted to be in close relation with the previous ones.

Several aspects still remain to be explored. Let us quote some of them. Since the results were derived under the Rayleigh fading, other shadowing/fading models should be analyzed. Other path-loss functions could be considered as well. A distribution of the transmitters location different than a Poisson point process could be assumed, such as a Poisson Cluster. The location of the receiver, not considered up to now, should be included in the analysis since the success of the transmission also depends on it. Finally, both ASN and ESN are intrinsically conservative and an evaluation of how conservative they are will be worthwhile.

Finally, we have addressed the problem of guaranteeing QoS in MANETs. Indeed, we have shown in our simulations that the performance of accepted connections in CSMA/CA can be very poor. We proposed two new MAC mechanisms that guarantee a minimum rate for all the accepted connections: SBAC and PCBA. Even if the ideas behind the proposed mechanisms are not new, we focused on its adaptation to the ad-hoc context and specially on its evaluation under different network scenarios by means of metrics defined accordingly.

In particular, to compare SBAC, PCBA and CSMA, we have considered different topologies (a line and a two-dimensional grid), different traffic scenarios (e.g. data and voice traffic) and different propagation models (e.g. Rayleigh fading and Lognormal shadowing). The objective was to determine which one is the "best" depending on the considered scenario. Our results showed that there is not a single answer to the question of which is the "best" mechanism.

For all the analyzed scenarios and performance metrics, it is one of our proposed mechanisms that obtains the best results. More in detail, we found that if elastic traffic is considered, SBAC should be chosen, since it provides rate levels that are generally larger than the target minimum. When a minimum rate is to be guaranteed, the best one depends on the considered metric, the minimum rate level and the attenuation coefficient. For instance, PCBA largely outperforms the rest of the mechanisms if the comparison is made in terms of spatial reuse. This is due to its capacity of controlling the transmission power, thus accommodating more simultaneous connections. At the same time, since it gives exactly the same rate to all the active transmissions, its performance decreases when other metrics (which explicitly include the rate) are considered. For example, SBAC is the mechanism which provides the best spatial density of rate. However, this particular property of PCBA is its main asset when constant bit rate traffic is considered: it obtains the best results irrespectively of the metric and the propagation model. When the tradeoff between rate and transmission power is considered, PCBA obtains very good results as long as the distance between transmitter and receiver is limited.

Implementation issues derived from the required decentralization of the proposed algorithms were discussed, and possible solutions were proposed. However, a more detailed analysis on their complexity and overhead should be performed. Since our analysis was focused on the slotted case, a future study should consider more dynamic scenarios. It would be interesting to evaluate if the well known "starvation phenomena" experienced by CSMA/CA is still present for the proposed mechanisms. Moreover, we considered regular topologies, but irregular ones assuming, for instance, random distribution of network nodes (e.g. Uniform or Poisson), should also be evaluated. Concerning the power control based mechanism, a maximum transmission power should be imposed to avoid non realistic solutions, and its impact on the performance of PCBA must be measured.

Appendix A

List of Publications

International Journals

- Paola Bermolen and Dario Rossi, “Support Vector Regression for Link Load Prediction (Extended version),” *Computer Networks*, Vol.53, Issue 2, p.p. 1389-1286, 2009.

- P.Bermolen, M.Mellia, M.Meo, D.Rossi and S.Valenti, “Abacus: Accurate Behavioural Classification of P2P-TV Traffic,” submitted to *IEEE Transactions on Network and Service Management*.

International Conferences

- Paola Bermolen and François Baccelli, “Multiple Access Mechanisms with Performance Guarantees for Ad-Hoc Networks,” submitted to *IEEE Secon* 2010.

- S.Valenti, D.Rossi, M.Meo, M.Mellia, and Paola Bermolen, “Accurate and Fine-Grained Classification of P2P-TV Applications by Simply Counting Packets,” in Proceedings of *Traffic Measurements and Analysis* (TMA), Springer Verlag LNCS 5537, p.p. 84-92, May 2009.

- S.Valenti, D.Rossi, M.Meo, M.Mellia, and Paola Bermolen, “An Abacus for P2P-TV Traffic Classification (Demo)” *IEEE Infocom* Demo, Rio de Janeiro, Brasil, April 2009,.

- Paola Bermolen and Dario Rossi, “Network Forecasting using support vector machines,” in Proceedings of *Traffic Management and Traffic Engineering for the Future Internet* (FiTRAMEn) - EuroNF workshop, Porto, Portugal, December 2008.

- Paola Bermolen and Dario Rossi, “Support Vector Regression for Link Load Prediction,” in Proceedings of the 4th *International Telecommunication NEtworking Workshop on QoS in Multiservice IP networks* (IT-NEWS), Venice, Italy, February 2008.

- Paola Bermolen and Dario Rossi, “Network Forecasting using support vector machines,” Extended Abstract at *EuroNGI Workshop on IP QoS and Traffic Control*, Lisboa, Portugal, December 2007.

Work in Progress

- Paola Bermolen and François Baccelli, “Modeling and Comparison of CCA Modes: Extremal versus Additive Shot Noise”.

Bibliography

[1] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, "Resource Management with Hoses: Point-to-cloud Services for Virtual Private Networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 679–692, 2002.

[2] G. Maruti and S. Suresh, "Greening of the Internet," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications.* ACM, 2003, pp. 19–26.

[3] "PPLive." [Online]. Available: http://www.pplive.com

[4] "SOPCast." [Online]. Available: http://www.sopcast.com

[5] "TVAnts." [Online]. Available: http://www.tvants.com

[6] "Joost." [Online]. Available: http://www.joost.com

[7] "Babelgum." [Online]. Available: http://www.babelgum.com

[8] "TVUnetworks." [Online]. Available: http://www.tvunetworks.com

[9] E. Leonardi, M. Mellia, A. Horvath, L. Muscariello, S. Niccolini, and D. Rossi, "Building a Cooperative P2P-TV Application over a Wise Network: the Approach of the European FP-7 strep NAPA-WINE [very large projects]," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 20–22, 2008.

[10] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1672–1687, 2007.

[11] A. W. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," in *SIGMETRICS '05: ACM SIGMETRICS international conference on Measurement and modeling of computer systems.* ACM, 2005, pp. 50–60.

[12] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service Mapping for QoS: a Statistical Signature-based Approach to IP Traffic Classification," in *IMC '04: 4th ACM SIGCOMM conference on Internet measurement.* ACM, 2004, pp. 135–148.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel Traffic Classification in the Dark," in *SIGCOMM '05: Conference on applications, technologies, architectures, and protocols for computer communications.* ACM, 2005, pp. 229–240.

[14] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling Internet Backbone Traffic: Behavior Models and Applications," *SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 169–180, 2005.

[15] V.Vapnik, *The Nature of Statistical Learning theory.* Springer, 1995.

[16] V. N. Vapnik, *Statistical Learning Theory.* Wiley-Interscience, September 1998.

[17] V. Vapnik, *Estimation of Dependences Based on Empirical Data [in Russian]*, 1979.

[18] L. Khan, M. Awad, and B. Thuraisingham, "A New Intrusion Detection System Using Support Vector Machines and Hierarchical Clustering," *The International Journal on Very Large Data Bases (VLDB)*, vol. 16, no. 4, pp. 507–521, 2007.

[19] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A Machine Learning Approach to TCP Throughput Prediction," *SIGMETRICS Performance Evaluation Review*, vol. 35, no. 1, pp. 97–108, 2007.

[20] F. Baccelli and B. Błaszczyszyn, *Stochastic Geometry and Wireless Networks.* Now Publishers. Foundation and Trends in Networking Series, 2009.

[21] M. Durvy, "Modeling the IEEE 802.11 Protocol in Wireless Multi-Hop Networks," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 2007.

[22] E. Pinsky and Y. Yemini, "The Asymptotic Analysis of Some Packet Radio Networks," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 6, 1986.

[23] E. Nadaraya, "On estimating Regression," *Theory of Probability and its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[24] C. J. C. Burges and B. Schölkopf, "Improving the Accuracy and Speed of Support Vector Machines," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997.

[25] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating Invariances in Support Vector Learning Machines," in *ICANN 96: Proceedings of the 1996 International Conference on Artificial Neural Networks.* Springer-Verlag, 1996, pp. 47–52.

[26] V. Blanz, B. Schölkopf, H. H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter, “Comparison of View-Based Object Recognition Algorithms Using Realistic 3D Models,” in *ICANN 96: Proceedings of the 1996 International Conference on Artificial Neural Networks*. Springer-Verlag, 1996, pp. 251–256.

[27] E. Osuna, R. Freund, and F. Girosit, “Training Support Vector Machines: an Application to Face Detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[28] J.-C. Terrillon, M. N. Shirazi, M. Sadek, H. Fukamachi, and S. Akamatsu, “Invariant Face Detection with Support Vector Machines,” in *International Conference on Pattern Recognition*, vol. 4, 2000, p. 4210.

[29] E. Leopold and J. Kindermann, “Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?” *Machine Learning*, vol. 46, no. 1-3, pp. 423–444, 2002.

[30] A. H. Sung and S. Mukkamala, “Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks,” in *SAINT ’03: Proceedings of the 2003 Symposium on Applications and the Internet*. IEEE Computer Society, 2003, p. 209.

[31] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, “Predicting Time Series with Support Vector Machines,” in *ICANN ’97: Proceedings of the 7th International Conference on Artificial Neural Networks*. Springer-Verlag, 1997, pp. 999–1004.

[32] S. Mukherjee, E. Osuna, and F. Girosi, “Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines,” in *IEEE Workshop on Neural Networks for Signal Processing*, 1997, pp. 511–520.

[33] E. E. Osuna and F. Girosi, “Reducing the Run-Time Complexity in Support Vector Machines,” *Advances in Kernel Methods: Support Vector Learning*, pp. 271–283, 1999.

[34] R. Beverly, K. Sollins, and A. Berger, “SVM Learning of IP address Structure for Latency Prediction,” in *MineNet’06: Proceedings of the 2006 SIGCOMM workshop on Mining network data*. ACM, 2006, pp. 299–304.

[35] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[36] M. Anthony and N. Biggs, “PAC Learning and Neural Networks,” *The handbook of Brain Theory and Neural Networks*, pp. 694–697, 1998.

[37] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[38] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory (COLT)*. ACM Press, 1992, pp. 144–152.

[39] A. Aizerman, E. M. Braverman, and L. Rozoner, “Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning,” *Automation and Remote Control*, vol. 25, pp. 821–837, 1964.

[40] J. Mercer, “Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations,” *Philosofical Transaction of the Royal Society, London*, pp. A209:415–446, 1909.

[41] U. H.-G. Kreßel, “Pairwise Classification and Support Vector Machines,” *Advances in Kernel methods: Support Vector Learning*, pp. 255–268, 1999.

[42] R. Rifkin and A. Klautau, “In Defense of One-Vs-All Classification,” *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

[43] J. C. Platt, N. Cristianini, and J. Shawe-taylor, “Large Margin DAGs for Multiclass Classification,” in *Advances in Neural Information Processing Systems.* MIT Press, 2000, pp. 547–553.

[44] A. Bordes, L. Bottou, P. Gallinari, and J. Weston, “Solving Multiclass Support Vector Machines with LaRank,” in *ICML ’07: Proceedings of the 24th international conference on Machine learning.* ACM, 2007, pp. 89–96.

[45] C.-W. Hsu and C.-J. Lin, “A comparison of Methods for Multiclass Support Vector Machines,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[46] T. Joachims, “Making Large-scale Support Vector Machine Learning Practical,” *Advances in Kernel Methods: Support Vector Learning*, pp. 169–184, 1999.

[47] C.-C. Chang and C.-J. Lin, *LIBSVM: a Library for Support Vector Machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[48] A.J.Smola and B. Scholkopf, “A Tutorial on Support Vector Regression,” *Statistics and Computing*, vol. 10, no. 3, pp. 199–222, 2004.

[49] Y. Guermeur and H. Paugam-Moisy, “Théorie de l’apprentisage de Vapnik et SVM, Support Vector Machines,” *Apprentissage Automatique*, pp. 109–138, 1999.

[50] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, “YALE: Rapid Prototyping for Complex Data Mining Tasks,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2006, pp. 935–940.

[51] B. Schölkopf, B. Scholkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, “Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers,” *IEEE Transactions on Signal Processing*, vol. 45, pp. 2758–2765, 1997.

[52] P. J. Brockwell and R. Davis, *Introduction to Time Series and Forecasting.* Springer, 1996.

[53] J. Beran, *Statistics for Long-memory Processes.* Chapman & Hall, 1994.

[54] B. Krithikaivasan, Y. Zeng, K. Deka, and D. Medhi, "ARCH-based Traffic Forecasting and Dynamic Bandwidth Provisioning for Periodically Measured Non-stationary Traffic," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 683–696, 2007.

[55] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-similar Nature of Ethernet Traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[56] Q. He, C. Dovrolis, and M. Ammar, "On the Predictability of Large Transfer TCP Throughput," *Computer Networks*, vol. 51, no. 14, pp. 3959–3977, 2007.

[57] S. Ruping and K. Morik, "Support Vector Machines and Learning About Time," in *ICASSP'03: IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2003, pp. 864–7.

[58] G. Watson, "Smooth regression analysis," *Sankhya, Series*, vol. A, no. 26, pp. 359–372, 1964.

[59] E.A.Nadaraya, *Non Parametric Estimation of Probability Density and Regression Curves*, ser. Soviet Series. Dordrecht,Boston: Kluwer Publishers Group, 1989, vol. 20.

[60] V. Cherkassky and Y. Ma, "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.

[61] M. A. M. B. J. L. V. Paxson, R. Pang and B. Tierney, "LBNL/ICSI Enterprise Tracing Project." [Online]. Available: http://imdc.datcat.org/

[62] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A First Look at Modern Enterprise Traffic," in *IMC '05: Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement.* USENIX Association, 2005, pp. 2–2.

[63] R. E.Schapire, "The Boosting Approach to Machine Learning: An Overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, 2002, pp. 2–2.

[64] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[65] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," in *IEEE Special Issue on Recent Advances in Distributed Multimedia Communications*, 2007.

[66] S. Sen, O. Spatscheck, and D. Wang, "Accurate, Scalable in-Network Identification of P2P Traffic Using Application Signatures," in *WWW '04: 13th international conference on World Wide Web.* ACM, 2004, pp. 512–521.

[67] A. W. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," in *Passive and Active Network Measurement*, 2005, pp. 41–54.

[68] J. Ma, K. Levchenko, C. Kreibich, S. Savage, and G. M. Voelker, "Unexpected Means of Protocol Inference," in *IMC '06: 6th ACM SIGCOMM Conference on Internet Measurement*. ACM, 2006, pp. 313–326.

[69] A. Mcgregor, M. Hall, P. Lorier, and J. Brunskill, "Flow Clustering Using Machine Learning Techniques," in *Passive and Active Network Measurement*, 2004, pp. 205–214.

[70] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing Skype Traffic: When Randomness Plays With You," *SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 37–48, 2007.

[71] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic Classification Through Simple Statistical Fingerprinting," *SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 5–16, 2007.

[72] L. Bernaille, R. Teixeira, and K. Salamatian, "Early Application Identification," in *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*. ACM, 2006, pp. 1–12.

[73] "Cisco Systems NetFlow Services Export Version 9," in *IETF RFC 3954*. Editor B. Claise, October 2004.

[74] Y.-X. Yang, R. Wang, Y. Liu, and X. yong Zhou, "Solving P2P Traffic Identification Problems Via Optimized Support Vector Machines," in *AICCSA '07: IEEE/ACS International Conference on Computer Systems and Applications*, 2007, pp. 165–171.

[75] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Transactions on Multimedia*, 2007.

[76] A. Bhattacharyya, "On a Measure of Divergence Between Two Statistical Populations Defined by Probability Distributions," *Bulletin Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[77] T. Kailath, "The Divergence and Bhattacharyya Distance Measures in Signal Selection," *IEEE Transactions on Communications [legacy, pre - 1988]*, vol. 15, no. 1, pp. 52–60, 1967.

[78] K. Matusita, "A Distance and Related Statistics in Multivariate Analysis," in *International Symposium on Multivariate Analysis*. Academic Press P.R. Krishnaiah (ed.), 1966, pp. 187–200.

[79] M. Mellia, R. Lo Cigno, and F. Neri, "Measuring IP and TCP Behavior on Edge Nodes with Tstat," *Computer Networks*, vol. 47, no. 1, pp. 1–21, 2005.

[80] "FastWeb Company Information," 2006. [Online]. Available: http://company.fastweb.it

[81] "IPP2P home page." [Online]. Available: http://www.ipp2p.org/

[82] Y. Kulbak and D. Bickson, "The eMule Protocol Specification," Leibniz Center, School of Computer Science and Engineering, TheHebrew University, Tech. Rep., 2005. [Online]. Available: http://www.cs.huji.ac.il/labs/danss/presentations/emule.pdf

[83] "Tshark." [Online]. Available: http://www.wireshark.org/

[84] T. Jebara and R. Kondor, "Bhattacharyya and Expected Likelihood Kernels," in *16th Annual Conference on Learning Theory (COLT) and 7th Annual Workshop on Kernel Machines*, ser. Lecture Notes in Artificial Intelligence, B. Schölkopf and M. Warmuth, Eds. Springer Verlag, 2003.

[85] T. Silerston and O. Fourmaux, "Measuring P2P IPTV Systems," in *NOSSDAV: International workshop on Network and Operating Systems Support for Digital Audio & Video*, 2007.

[86] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, and M. Meo, "P2P-TV Systems Under Adverse Network Conditions: a Measurement Study," in *IEEE Infocom*, 2009.

[87] D.Bertsekas and R.Gallager, *Data Networks.* Prentice Hall, 1988.

[88] S. Keshav, *An Engineering Approach to Computer Networking: ATM networks, the Internet, and the Telephone Network.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[89] X. Wang and K. Kar, "Throughput Modelling and Fairness Issues in CSMA/CA Ad-hoc Networks," *IEEE Infocom*, 2005.

[90] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[91] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed-point analysis of single cell ieee 802.11 wlans," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 588–601, 2007.

[92] S. Pollin, M. Ergen, S. Ergen, B. Bougard, L. Van der Perre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3359–3371, 2008.

[93] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, June 12 2007.

[94] I. Ramachandran and S. Roy, "On the Impact of Clear Channel Assessment on MAC Performance," in *IEEE GLOBECOM'06: Global Telecommunications Conference*, 2006, pp. 1–5.

[95] M. Durvy and P. Thiran, "A Packing Approach to Compare Slotted and Non-Slotted Medium Access Control," in *IEEE Infocom*, 2006, pp. 1119–1207.

[96] F. Kelly, "Loss Networks," *The Annals of Applied Probability*, vol. 1, pp. 319–378, 1991.

[97] P. Brémaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulation and Queues.* Springer-Verlag, New York, Inc., 1999.

[98] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes.* Springer-Verlag, New York, Inc., 1988.

[99] M.Haenggi, J.Andrews, F.Baccelli, O.Dousse, and M. (Editors), "Stochastic Geoemtry and Random Graphs for Wireless Networks," *Special Issue of IEEE JSAC*, 2009.

[100] H. Nguyen, F. Baccelli, and D. Kofman, "An stochastic geometry analysis of dense IEEE 802.11 networks ans its use in economic modeling," in *IEEE Infocom*, 2007, pp. 1119–1207.

[101] M. Talagrand, *Spin Glasses: A Challenge for Mathamaticians.* Springer Verlag, 2000.

[102] N.Campbell, "Discontinuities in light emission," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 15, pp. 117–136, 1909.

[103] M. Haenggi and R. K. Ganti, *Interference in Large Wireless Networks.* Now Publishers. Foundation and Trends in Networking Series, 2009, vol. 3, no. 2.

[104] F. Baccelli and V. M. Nguyen, "Best Signal Quality in Interference Fields," personal communication, 2009.

[105] F. Tournois, "Modélisation et Simulation de Réseaux CDMA par la Géométrie Aléatoire," Ph.D. dissertation, Ecole des Mines de Paris, 2002.

[106] B. Błaszczyszyn and D.Yogeshwaran, "Directionally Convex Ordering of Random Measures Shot Noise Fields and Some Applications to Wireless Communications," *Journal of Applied probability*, vol. 41, no. 3, pp. 623–646, 2009.

[107] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a Media Access Protocol for Wireless LAN's," *ACM SIGCOMM*, pp. 212–225, 1994.

[108] F. Calì, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, 2000.

[109] J. Deng, B. Liang, and P. K. Varshney, “Tuning the carrier sensing range of IEEE 802.11 MAC,” *IEEE GLOBECOM*, 2004.

[110] C. Chaudet, G. Chelius, H. Meunier, and D. Simplot-Ryl, “Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 MAC Layer,” *Fourth Annual Mediterranean Ad Hoc NetworkingWorkshop*, 2006.

[111] O. Goussevskaia, Y. Oswald, and R. Wattenhofer, “Complexity in Geometric SINR,” *ACM Mobihoc*, 2007.

[112] G. J.Foschini and Z. Miljanic, “A Simple Distributed Autonomous Power Control Algorithm ans its Convergence,” *IEEE Transactions of Vehicular Technology*, vol. 42, no. 4, pp. 641–646, 1993.

[113] S. Grandhi, R. Viyajan, and D. Goodman, “Distributed Power Control in Cellular Radio Systems,” *IEEE Transaction on Communications*, vol. 1, no. 4, 1995.

[114] T. Moscibroda and R. Wattenhofer, “The Complexity of Connectivity in Wireless Networks,” *IEEE Infocom*, 2006.

[115] S. A. Borbash and A. Ephremides, “Wireless Link Scheduling With Power Control and SINR Constraints,” *IEEE Transaction on Information Theory*, vol. 52, no. 11, 2006.

[116] R. Cruz and A. V. Santhanam, “Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Network,” *IEEE Infocom*, 2003.

[117] T. ElBatt and A. Ephremides, “Joint Scheduling and Power Control for Wireless Ad-hoc Networks,” *IEEE Infocom*, 2002.

[118] ——, “A Power Controlled Multiple Access Protocol for Wireless Packet Networks,” *IEEE Infocom*, 2001.

[119] T. Moscibroda, R. Wattenhofer, and Y. Weber, “Protocol Design Beyond Graph Based Models,” *Hot Topics in Networks (HotNets-V)*, 2006.

[120] G. Brar, D. Blough, and P. Santi, “Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks,” *ACM Mobicom*, 2006.

[121] D. Chafekar, V. A. Kumar, M. V.Marathe, S. Parthasarathy, and A. Srinivasan, “Approximation Algorithms for Computing Capacity of Wireless Networks with SINR constraints,” *IEEE Infocom*, 2008.

[122] O. Goussevskaia, R. Wattenhofer, M. M.Halldorsson, and E. Welzl, “Capacity of Arbitrary Wireless Network,” *IEEE Infocom*, 2009.

[123] J. Zander, “Performance of Optimumt Transmitter Power Control in Cellular Radio Systems,” *IEEE Transactions of Vehicular Technology*, vol. 41, no. 1, pp. 57–62, 1992.

[124] T. Bonald and A. Proutière, “Conservative Estimates of Blocking and Outage Probabilities in CDMA Networks,” *Performance Evaluation*, vol. 62, pp. 50–67, 2005.