An STDMA-Based Framework for QoS Provisioning in Wireless Mesh Networks

Mauro Leoncini
Dip. Ing. dell'Informazione
Univ. of Modena e Reggio Emilia
ITALY

Paolo Santi*
IIT-CNR
Pisa, ITALY

Paolo Valente
Dip. Ing. dell'Informazione
Univ. of Modena e Reggio Emilia
ITALY

Abstract

Providing strong QoS guarantees for wireless multi-hop networks is very challenging, due to many factors such as use of a shared communication medium, variability in wireless link quality, and so on. However, wireless mesh technology gives the opportunity to alleviate some of these problems, due to lack of mobility in the wireless infrastructure, and presence of natural centralization points in the network. The main contribution of this paper is the definition of a simple framework that exploits these features to provide provable, strong QoS guarantees to network clients. In particular, admitted clients are guaranteed a certain minimum bandwidth and maximum delay on their connections. The framework is based on STDMA scheduling at the MAC layer, which is periodically executed at the network manager to adapt to changes in traffic demand. While scheduling computation is centralized, admission control is performed locally at the wireless backbone nodes, thus reducing signaling. We propose two bandwidth distribution and related admission control policies, which are at opposite ends of the network utilization/spatial fairness trade-off. Through extensive simulations, we show that the proposed framework achieves its design goals of providing strong QoS guarantees to VoIP clients while not sacrificing throughput in a realistic mesh network scenario, also in presence of highly unbalanced load at the backbone nodes. To the best of our knowledge, this is the first proposal with similar features for wireless mesh networks.

1 Introduction

Wireless mesh networks are a very promising technology for providing ubiquitous broadband wireless access, mainly due to their ease of deployment and maintenance. However, several challenges are still to be faced in order for these premises to be fulfilled. The main challenges are related to an efficient use of the wireless spectrum to reduce interference and increase capacity, and to provide QoS guarantees to the network clients.

Providing QoS guarantees to clients of a wireless multi-hop network is a very challenging, long standing problem. Here, difficulties lie in the fact that a very basic component of any QoS provisioning framework (i.e., accurate characterization of link-level performance) is hard to achieve, due to factors such as use of shared communication medium, variability in radio channel conditions, possible node mobility, and so on. However, mesh networks have peculiar features with respect to other types of wireless multi-hop networks which may help a lot in the definition of such a QoS framework. In particular, mesh networks are characterized by the presence of a wireless infrastructure, and of natural centralization points (e.g., gateway nodes).

The main contribution of this paper is showing how to exploit these features (lack of mobility in the wireless infrastructure, and presence of centralization points) to build a relatively simple framework able to provide *strong* (i.e., worst-case) QoS guarantees to network clients. This is in sharp contrast with similar frameworks proposed for other types of wireless multi-hop networks, which typically only achieve *weak* (i.e., statistical) QoS guarantees.

Our framework is based on exploiting STDMA scheduling at the MAC layer. Using STDMA-based MAC is deemed feasible in a wireless mesh network, due to the quasi-static topology of the wireless infrastructure, and the presence of centralization points. Indeed, forthcoming standards for wireless mesh networks such as 802.16 and 802.11s are based on STDMA scheduling. Concerning presence of centralization points, we observe that the concept of managed mesh network is becoming very popular, e.g., in enterprises and campuses, where a single authority deploys and manages the entire network, deployment is planned and a centralized service is often available.

Exploiting STDMA at the MAC layer allows mitigating unpredictability of wireless link performance due to the shared nature of the communication medium, since transmissions are carefully scheduled in such a way that they do not conflict with each other. Hence, up to changes in radio channel properties (which are not addressed in this work), the performance of each network link becomes in principle

*The work of P. Santi has been partially supported by IIT-CNR under Project P056: "Networks for Critical Infrastructures and Web Analysis".

predictable in a very accurate way.

The key of our framework, which we call WoW, is to use accurate link performance estimation to provide strong QoS guarantees to the wireless backbone nodes composing the wireless mesh network infrastructure, which can be turned into strong QoS guarantees provided to the end user if proper scheduling and admission control policies are locally executed at the backbone nodes.

In this paper, we formally prove that the WoW framework, when complemented with simple backbone node-level policies, provides strong QoS guarantees to the end user in terms of minimum available bandwidth and maximum possible delay on her connection to a gateway node. Furthermore, we show through extensive packet-level simulation that WoW effectively provides strong QoS guarantees to VoIP users in a realistic mesh network scenario. The simulation results also show WoW's ability to provide these guarantees and to effectively use network bandwidth even in presence of highly unbalanced network load. To the best of our knowledge, WoW is the first framework for QoS provisioning in mesh networks with similar features proposed in the literature.

2 Related work and contribution

The issue of providing QoS guarantees in wireless multi-hop networks has been widely investigated in the literature, in particular with reference to wireless ad hoc networks. QoS constraints have been considered both at the MAC layer [18, 21, 23, 24], at the routing layer [15, 16, 17, 22], and in cross-layer solutions [9, 14]. For a survey on QoS approaches for wireless ad hoc networks, the reader is referred to [19]. Given the fully distributed, mobile nature of ad hoc networks, accurately characterizing bandwidth and delay characteristics of each link in the network (which is the basic ingredient for providing strong QoS guarantees) is very challenging. This is the reason why typical QoS approaches for ad hoc networks can provide only weak (e.g., statistical) guarantees on QoS level guaranteed to clients.

A few approaches specifically targeted to mesh networks have recently been introduced in the literature. In [3, 7], the problem of optimal selection of gateway nodes in order to provide QoS guarantees to clients is considered. A major limitation of these approaches is that only primary interference is considered when determining QoS guarantees. This means that the only constraint when determining feasibility of a transmission set is that a node cannot be at the endpoint of two or more active links. In [13], the authors are concerned with dimensioning a wireless mesh network in such a way that certain flow-level QoS requirements are fulfilled. Similarly to [3, 7], the approach of [13] is based on a TDMA MAC layer, and uses a simplified interference model.

The authors of [6] are concerned with fairness, and presents a max-min fairness approach based on TDMA scheduling. Fairness is also the main focus of [10], in which the authors show that nodes which are a few hops away from the gateway are severely penalized in 802.11-based mesh networks.

Differently from all the QoS approaches we are aware of, the framework presented in this paper is based on an accurate interference model, namely the physical interference model introduced in [11]. The difficulty in using this interference model lies in the fact that interference generated even by far-away nodes is accounted for when determining whether a set of concurrent transmission is feasible. Only recently a computationally feasible approach to build an STDMA schedule based on the physical interference model has been proposed in the literature [8]. The scheduling algorithm presented in [8], called GreedyPhysical, exhibits low running time when executed on networks of even moderate size (a hundred of nodes), and it is shown to achieve up to three-fold performance improvements in terms of throughput with respect to 802.11-based networks. Note that usage of an accurate interference model allows to accurately compute feasible transmission sets in the schedule, resulting in high link reliability. On the other hand, packet reception cannot be ensured with 100% probability, due to the well-known time-varying behavior of radio signal. To ease presentation, in the following we assume that this type of (unavoidable) link-reliability is dealt with standard techniques (e.g., packet retransmission at the application level). Thus, when we mention "strong" QoS-guarantees, we mean guarantees up to the above mentioned, seldom occurring, packet losses.

In this paper, we propose to use GreedyPhysical as a building block of a simple, centralized STDMA framework for providing strong, end-to-end guarantees in a wireless mesh network. End-to-end guarantees are on both minimum bandwidth and maximum delay, and are strong in that they are guaranteed in the worst-case. The framework is based on a very accurate estimation of minimum bandwidth and maximum delay on each network link, which renders the performance of the network in principle predictable as in a wired network. This is the reason why we have termed our framework WoW (Wired on Wireless). Accurate bandwidth/delay estimation of network links is made available by the use of very accurate interference modeling when building the STDMA schedule.

3 Network model

We consider a wireless mesh network composed of n nodes (wireless routers), some of which have direct, wired connection to the Internet and act as *gateway nodes*. For the ease of reading, in the rest of this paper we will refer to gateway nodes simply as *gateways*, while we reserve the term *node* as a shorthand for "non-gateway backbone node". All possible links between nodes (and gateways) are represented by the communication graph $G = (V, E)$, where $e = (u, v) \in E$ if and only if there exists a communication link (in the absence of interference) between u and v, where u, v are either node or gateways. We restrict our attention to bi-directional links only, i.e., we do not consider, for routing packets, possible uni-directional links in the communication graph.

We assume that packets to/from nodes from/to gateways are routed along shortest path trees rooted at the gateways. In other words, for each node u, we suppose that u routes its packets toward a closest gateway v, where distance is simply measured as hop count, and with ties broken by means of random choices. However, we require that the routing protocol guarantees that the final routing graph is a forest, with every tree rooted at a gateway. Note that our working assumption of routing along disjoint trees is realistic in a wireless mesh scenario.

The WoW framework is aimed at providing node-level QoS guarantees, and it is independent of the policies implemented within the nodes. However, to assess WoW effectiveness in providing QoS guarantees to final users, we assume that (mobile) *wireless clients* are present, which issue connection requests to a node within their radio range. Hence each node[1] u is associated with a dynamic set of clients, which we shall denote by $C(u)$. We assume clients are QoS sensitive. In particular, a generic client a is characterized by two QoS parameters, namely a *minimum bandwidth* requirement, $bw(a)$, and a maximum delay constraint $del(a)$, which we will together refer to as the *QoS demand* of a: $QoS(a) = \{bw(a), del(a)\}$. For each node u, the value $\sum_{a \in C(u)} bw(a)$ will be taken as u's *bandwidth requirement*.

As it is typically the case in wireless mesh networks, we assume that client-to-infrastructure (C2I) and infrastructure-to-infrastructure (I2I) communications use disjoint set of wireless resources, i.e., that C2I and I2I communications do not interfere with each other. This is made possible, for instance, by usage of at least two radios on the wireless routers, and, say, usage of different technologies (e.g., 802.11a and 802.11b/g) for C2I and I2I communications[2].

4 The WoW Framework

A system view of the WoW framework is depicted in Figure 1. Four types of actors are considered: a network manager, gateways, nodes, and clients.

The network manager, which may possibly coincide with one of the gateways, periodically computes the transmission schedule based on node bandwidth requirements[3]. Then, it

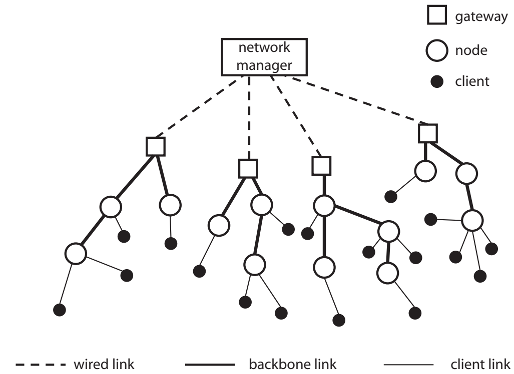


Figure 1. The *WoW* framework.

broadcasts to each gateway g the schedule and the bandwidth allocated to each node in the tree rooted at g.

In turn, each gateway takes care of delivering the transmission schedule and the bandwidth allocation to each node in its tree. Also, in the opposite direction, it collects bandwidth requirements from nodes, and delivers them to the network manager.

Nodes realize the important task of converting the aggregate QoS guarantees provided by the WoW framework into client-level guarantees.

Finally, as already pointed out, there are wireless clients, which issue connection requests to a node within their radio range. In case of successful request, they are allowed to send/receive packets to/from a gateway node.

4.1 MAC level scheduling

The WoW framework is based on the existence of an STDMA protocol at the MAC layer, according to which packet transmissions occur in *time slots* (*transmission opportunities*) of fixed length t_{slot}.

The scheduling algorithm we will use here is the GreedyPhysical protocol of Brar et al. [8], up to some changes described in Section 5. Our choice is motivated by the fact that GreedyPhysical is based on an accurate *physical model* of interference that accounts for bi-directional transmissions on a link. Accurate interference modeling has two advantages, i.e.: i) allowing a higher number of packets to be transmitted in parallel during a time slot with respect to less accurate interference models, such as the *protocol model* [8], and ii) providing better link quality estimation, thus enabling concurrent scheduling of high quality links only. Moreover, GreedyPhysical retains a sufficiently low computational cost that enables frequent recomputations of the schedule, a key ingredient of our proposed design.

GreedyPhysical takes as input a weighted communication graph, in which a weight ℓ on a link represents a demand for $\ell \in \mathbb{N}$ transmission opportunities on that link. The goal of the scheduler is to accommodate all the transmission

[1]For simplicity, we assume that clients cannot be directly associated with gateways.

[2]An alternative is to use disjoint set of orthogonal frequency channels for C2I and I2I communications.

[3]Given the much higher bandwidth and lower delay of a wired link as compared to a wireless link, we assume that message exchange between gateways and the network manager occurs with a negligible delay.

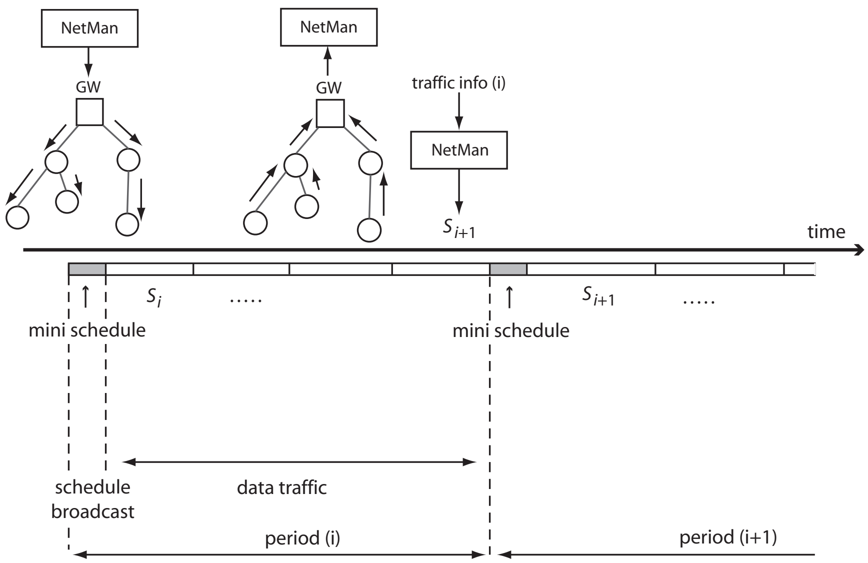


Figure 2. Adaptive schedule computation with WoW.

requests using the minimum possible number of slots, by parallelizing transmissions that do not interfere. However, clients typically do not a priori know the amount of traffic they have to transmit. In fact, in our framework client requirements are expressed in terms of bandwidth (and delay) requirements, which are then converted into aggregate link-level QoS requirements. The *key of our design* is thus to dynamically take as input these link-level requirements, to turn them into a set of link weights, and to execute GreedyPhysical in such a way to guarantee the satisfaction of the requirements. The way this is achieved is described in detail in the next subsection.

4.2 Network operations

As depicted in Figure 2, transmission opportunities are organized into consecutive *periods*. Each period i is in turn composed of an initial *mini-schedule* followed by a fixed number q of consecutive identical *schedules* S_i.

The schedule S_i is computed by the network manager (during period $i-1$) as follows. Each node u is first assigned a weight $w_i(u)$; node weights are then aggregated into link weights (see Figure 3) which are given in input to GreedyPhysical. The schedule returned by GreedyPhysical, call it $\hat{S}_i$, is then compared against a *reference schedule*, and the shortest one (in terms of number of allocated time slots) is returned as S_i. The latter is then broadcast to the gateways, via wired connections, and then to the nodes at the beginning of the i-th period (i.e., during the mini-schedule).

The exact choice of the reference schedule depends on the actual bandwidth distribution policy adopted, as will be explained in Section 5. However, we must stress here that, if the reference schedule is picked as S_i, then it must be able to accomodate all the transmission opportunities in $\{w_i(u)\}_{u\in\mathcal{N}}$, where $\mathcal{N}$ is the set of nodes.

Note that the transmission opportunities allocated in S_i to a given node u must be destined by u to both local and transit traffic. More specifically, it is mandated that each node u uses exactly $w_i(u)$ slots for local traffic, and the remaining slots for transit traffic. Regarding the local traffic, we assume nodes execute a *local scheduler*, which is in charge of distributing the node's bandwidth among the currently registered clients. The details of the adopted scheduling algorithm, as well as the node-level QoS guarantees it provides, are out of the scope of this paper. In the following, we assume a simple round-robin or fair-queueing policy is enforced. For what concerns transit traffic, we simply assume that *all* the traffic that is received/destined from/to descendants in the tree is transmitted in the transit traffic slots. As we shall see, this property is important for providing strong end-to-end delay guarantees to the final user.

The schedule S_i is not the only information delivered by the network manager to the nodes in the mini-schedule. Each node u also receives
1. the maximum weight $\hat{w}_{i+1}(u)$ that u will be allowed to require for period $i+1$;
2. the length $\hat{N}_{i+1}$ of the reference schedule that the network manager will use in the next period $(i+1)$, expressed in terms of number of slots. As a consequence of the shortest schedule selection step described above, $\hat{N}_{i+1}$ is also a bound to the length of the next schedule S_{i+1} (i.e., S_{i+1} will last at most $\hat{N}_{i+1}\cdot t_{slot}$ seconds).

We will refer to the pair $\langle\hat{w}_{i+1}(u), \hat{N}_{i+1}\rangle$ as to the *bandwidth distribution profile* of node u for period $i+1$.

Given the guarantee on S_{i+1}'s maximum length, we can define

$$BW_{i+1}(u) = \frac{\hat{w}_{i+1}(u)}{\hat{N}_{i+1}t_{slot}} s \text{ bytes/sec.} \tag{1}$$

as the link-level bandwidth guaranteed to node u in period $i+1$, where s is the number of (level 2) bytes transmitted in a slot. The weight aggregation rule used to compute the input to GreedyPhysical (recall Figure 3) ensures that $BW_{i+1}(u)$ is actually the end-to-end bandwidth from/to node u to/from its gateway. Hence, $BW_{i+1}(u)$ is the aggregate bandwidth available at node u for accommodating client requests.

This bandwidth is used by the nodes in the following basic admission control (AC) mechanism. Depending on the clients' connection requests and session expirations, node u repeatedly determines the desired bandwidth $BW^{des}_{i+1}(u)$ for the next period and hence, by inverting formula (1), the desired weight

$$w^{des}_{i+1}(u) = \left\lceil BW^{des}_{i+1}(u) \frac{\hat{N}_{i+1} t_{slot}}{s} \right\rceil . \qquad (2)$$

Finally, the updated value

$$w_{i+1}(u) = \max\left\{1, \min\left\{w^{des}_{i+1}(u), \hat{w}_{i+1}(u)\right\}\right\}$$

is piggybacked to the network manager in the data traffic at each transmission opportunity. Note that: (1) taking the minimum among $w^{des}_{i+1}(u)$ and $\hat{w}_{i+1}(u)$ implies that the node might reject or delay some client connection requests; (2) imposing that $w_{i+1}(u) \geq 1$ is required just for piggybacking the information, even in case of no traffic requirements in the current period.

As shown in Section 5, the above AC mechanism and rule for setting the desired weights, combined with proper computation of the maximum allowed weights $\hat{w}_{i+1}$ and the node internal scheduler, ensure that once a client a with QoS demand $\{bw(a), del(a)\}$ is admitted in the network, its QoS demand is *guaranteed* to be satisfied during the entire session duration. Also, in Section 5 we will see that the node AC mechanism leads to different AC policies when combined with different network-level bandwidth distribution policies (i.e., with the actual values $\hat{w}_{i+1}(u)$ and $\hat{N}_{i+1}$ delivered by the network manager to the nodes).

At system startup, the network manager sets $w_0(u) = \vartheta \in \mathbb{N}$, $\forall u$. The parameter ϑ is critical in our design, and it is used to tune the maximum delay vs. bandwidth distribution granularity tradeoff. Intuitively speaking, ϑ determines the duration of the schedule: a higher value of ϑ results in a longer schedule, which is bad for reducing delay, but allows a finer granularity in the computation of the desired weights (see equation (2)), which in turns achieves a better network bandwidth utilization. In practice, ϑ should be set to the maximum possible value compatible with the application delay requirements. For instance, in the case of VoIP applications, our simulation results show $\vartheta \leq 5$ is feasible.

With regard to delay guarantees, the number q of schedule repeats in each period is set to be at least the maximum

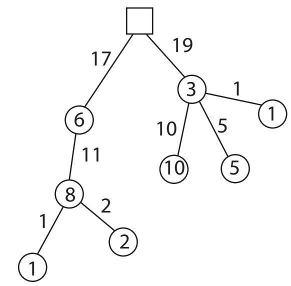


Figure 3. Example of weight aggregation: nodes are labeled with weights $w_i(u)$; links are labeled with the resulting aggregated link weights.

height of the trees in the forest. Such a choice, together with the node slot allocation rule (i.e., the fact that, in each schedule S_i, a node u uses its transmission opportunities in excess of $w_i(u)$ for transit traffic) and the weight aggregation rule (recall Figure 3), guarantees that a packet originating at any node reaches its gateway within one period. We stress that, given the accurate interference model used to compute the schedule, hop distance of a node to its gateway is the main factor affecting packet delay.

We conclude this section by observing that, while schedule and bandwidth allocation is computed by the network manager, client connection requests are locally dealt by nodes, i.e., they are managed in a decentralized way. This ensures a considerable signaling reduction with respect to centralized solutions for managing connection requests.

4.3 Setting the period length

Another important parameter in our design is the duration of a period, whose actual value must take conflicting requirements into account. On the one hand, the period length must be long enough for the network manager to compute the schedule and bandwidth distribution profiles for the next period. From this viewpoint, it is feasible to think of periods as lasting a few hundred milliseconds. Also, having a long period is good for reducing the impact of the overhead due to the presence of the mini-schedule, which has fixed length. On the other hand, a short period (subject to have q at least as large as the maximum tree heights) is desirable to increase the rate of adaptation of the schedules to the observed traffic.

We have verified through simulations that the length of the mini-schedule is actually very short for networks of reasonable size (around 100 nodes). This implies a negligible signaling overhead even when the period length is set to the minimum duration required to execute the network manager algorithms, which is thus the value we have picked.

5 Bandwidth distribution and AC policies

In this section we show two basic bandwidth distribution policies, that stand at opposite ends of the "spatial fairness –

network utilization" scale. As we will show, these policies, combined with the AC mechanism described in Section 4.2, give rise to two different AC policies.

At network startup, both policies compute a first *worst-case* schedule S_0, assuming that all the nodes require the maximum possible weight ϑ (equivalent to continuously backlogged links). Its length is denoted as N_0. In S_0 any node u is guaranteed the same bandwidth $BW_0(u) = BW_0 = \frac{\vartheta}{N_0 t_{slot}} s$ bytes/sec.

5.1 Static bandwidth distribution

The first bandwidth distribution/AC policy, which we call SFC (Spatial Fair, Conservative policy) is a *static* one. In SFC, S_0 is used as the reference schedule in each schedule computation. That is, the schedule S_i is obtained by comparing the schedule $\hat{S}_i$, computed by GreedyPhysical, against S_0 and choosing the shortest one (see Section 4.2).

It is now the appropriate time to explain the need for this additional shortest schedule selection step. The reason is that, given two sets of link weights W_1 and W_2, such that $SW_1 \leq SW_2$, where $SW_i = \sum_{w \in W_1} w, i = 1, 2$, GreedyPhysical does not guarantee (actually, cannot guarantee) that the schedule for W_1 is not longer than that for W_2, as the following example shows.

Consider the weighted communication graph of Figure 4, which also reports the node weights $w_i(u)$. Suppose also that, because of interference, the schedule produced by GreedyPhysical is $S_i = \langle\{1,3,5\},\{1,3,6\},\{2,4\},\{4,5\}\rangle$, whose length is 4. Suppose now that $w_{i+1}(3) = 1$ while $w_{i+1}(u) = w_i(u)$, for $u \neq 3$. Then it is perfectly conceivable[4] that interference dependencies impose $S_{i+1} = \langle\{1,3,5\},\{1,4\},\{2,4\},\{5\},\{6\}\rangle$, which is longer than S_i.

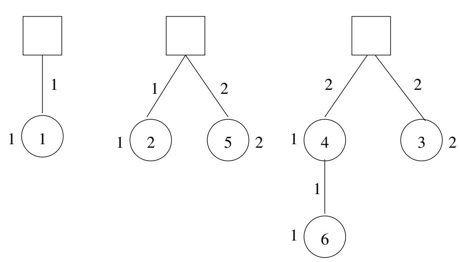


Figure 4. A communication graph for GreedyPhysical.

Unfortunately, the schedule length *monotonicity* property described above is required for SFC to provide strong QoS bandwidth and delay guarantees, as we will see next.

As we already know, at the beginning of each period i, the network manager broadcasts, together with the schedule S_i, also the bandwidth profile $\langle\hat{w}_{i+1}(u), \hat{N}_{i+1}\rangle$, which in SFC is the same for each node u and it is also independent of i, since $\hat{w}_{i+1}(u) = \vartheta$ and $\hat{N}_{i+1} = N_0$. This implies, in particular, that any node u is always guaranteed the same bandwidth BW_0, which it can use for admission control. It is now clear that, were the actual schedule S_{i+1} longer than S_0, the bandwidth received by the nodes in the next period would be smaller than BW_0, and this would lead to a violation of the assumed guarantee. On the other hand, if the reference schedule S_0 is picked by the selection step, then we know that it can always accomodate all the transmission opportunities requested by each node u since $w_i(u) \leq \vartheta$.

Note, on the contrary, that the actual bandwidth node u receives during the generic period i – which we recall is equal to $\frac{w_i(u)}{N_i t_{slot}} s$ bytes/sec, where N_i is actual length of S_i – might be higher than $BW_0(u)$: in fact, if some of the nodes v is not using all its reserved bandwidth $BW_0(v)$, schedule S_i is likely to be shorter than S_0, resulting in a higher bandwidth for some of the nodes. However, our framework guarantees that BW_i is never *below* $BW_0(u)$.

Similarly, the maximum delay for node u is guaranteed to be at most $Del_{max}(u) = h(u) \cdot N_0 \cdot t_{slot}$, where $h(u)$ is the hop-distance between node u and its gateway. This immediately follows by the weight aggregation rule, and by the fact that nodes use slots in excess to their own weight to transmit transit traffic.

Now, once $BW_0(u) = BW_0$ and $Del_{max}(u)$ are set for each node u, the SFC AC policy is straightforward: a new client a requesting a connection with QoS demand $\{bw(a), del(a)\}$ is admitted if and only if $BW_0(u) - BW^{cur}(u) \geq bw(a)$ and $del(a) \geq Del_{max}(u)$, where $BW^{cur}(u) \leq BW_0$ is the aggregated bandwidth which is reserved for clients currently registered at node u.

The above results can be summarized in the following

Proposition 1. *Under the SFC policy, if a client a with QoS demand $\{bw(a), del(a)\}$ is admitted into the network at node u, then a is guaranteed to receive at least $bw(a)$ bandwidth and to incur delay at most $Del_{max}(u) \leq del(a)$.*

Note that SFC is spatially fair when distributing bandwidth among nodes. On the other hand, different nodes in the network have different maximum delay guarantees depending on how close they are to their gateways, resulting in spatial unfairness which is not dealt with by SFC. In other words, SFC ensures spatial fairness for what concerns bandwidth, but it is still not fair for what concerns delay.

It is also worth observing that highly loaded nodes can receive bandwidth in excess to BW_0. However, owing to SFC's conservative nature, this extra bandwidth can be utilized only for best-effort traffic, and it cannot be used to allocate more clients. In fact, in SFC the aggregate bandwidth guaranteed to a node is computed once and for all at the beginning of the network operation under worst-case assumptions, and it is the same for all the nodes – which ensures spatial fairness in bandwidth allocation.

[4] Suppose, for instance, that nodes 5 and 6 are very close to each other.

5.2 Dynamic bandwidth distribution

The second bandwidth distribution/AC policy, called SUA (Spatial Unfair, Aggressive policy), sacrifices spatial fairness to improve bandwidth utilization. Differently from SFC, SUA implements dynamic bandwidth allocation to the nodes[5]. The idea is that the extra aggregate bandwidth which is possibly available at the nodes in a certain period can be used to admit more QoS-sensitive clients.

At the beginning of each period i, the network manager broadcasts $\hat{w}_{i+1}(u) = w_i(u)$ and $\hat{N}_{i+1} = N_i$ to each node u. The latter, in particular, is a consequence of the fact that, in the SUA policy, we take the current schedule (say, schedule S_i) as the reference schedule for the next period (i.e., period $i+1$). In other words,

$$S_{i+1} = \mathrm{argmin}\{\mathrm{length}(S_i), \mathrm{length}(\hat{S}_{i+1})\}.$$

Similarly to what happens for the static policy, this guarantees that S_{i+1} will not be longer than $N_i \cdot t_{slot}$, and also that the reference schedule can accomodate all node demands, since $w_{i+1}(u) \leq \hat{w}_{i+1}(u) = w_i(u)$, for each node u.

Each node then enforces the following criterion for admission control (the delay criterion being the same as in SFC): a client a requesting a connection with minimum bandwidth $bw(a)$ is accepted only if $BW_i(u) - BW^{cur}(u) \geq bw(a)$, where $BW_i(u)$ is the bandwidth of node u during period i (which, in general, can be different than BW_0). That is, differently from SFC the available aggregate bandwidth is adapted to the current network load.

SUA ensures that nodes can only decrease their weights. As a consequence of this, it has the disadvantage of decreasing the granularity of the bandwidth allocation as time goes by. In the extreme case, all nodes end up with weight 1, implying that from that point on bandwidth allocation is no longer adaptive to changes in network load. To circumvent this problem, SUA implements the following (weight) *push-up* rule. Let w_i^{max} be the maximum among node weights during period i; then, *before* invoking GreedyPhysical, all the weights are modified as follows

$$w_{i+1}(u) \leftarrow w_i(u) \cdot \left\lfloor \frac{\vartheta}{w_i^{max}} \right\rfloor . \tag{3}$$

Proposition 2. *Under the SUA policy, if a client a with QoS demand $\{bw(a), del(a)\}$ is admitted into the network, then a is guaranteed to receive at least $bw(a)$ bandwidth and to incur delay at most $del(a)$.*

Proof. The only difference with respect to the case of SFC is that here we must take possible weight push up into account. Let G be a communication graph and let $W(G)$ be the set of link weights in G. Also, for an integer t, let $t \cdot W(G)$ denote the set obtained by multiplying each element of $W(G)$ by t. It is not difficult to prove that the length of the schedule produced by GreedyPhysical on $t \cdot W(G)$ is exactly t times the length of the schedule obtained on input $W(G)$ (the formal proof is deferred to the full paper). However, this implies that the in equation (1), both the numerator (i.e., $\hat{w}_{i+1}(u)$) and the denominator (i.e., $\hat{N}_{i+1}$) are possibly multiplied by the same factor, which clearly implies that the bandwidth guaranteed remains unchanged. $\square$

Observe that under the SUA policy it is possible that a node u accepts clients in excess to the fair share BW_0. This unavoidably compresses the aggregate bandwidth available to some other nodes in the network (which are currently lightly loaded), possibly even below BW_0. Hence, contrary to SFC, SUA does not guarantee spatial fairness in bandwidth allocation.

6 Simulations

In this section we describe our simulation setup and report the obtained results. Simulation results are intended to show that the WoW framework: 1) effectively enables a real-world QoS-sensitive application to run, 2) imposes a negligible overhead and fully utilizes the available total bandwidth, 3) *properly* distributes this total bandwidth to each node.

We implemented the WoW framework inside the Georgia Tech Network Simulator [2]. More specifically, our starting point was the modified version of the simulator used in [8], which implements the physical interference model and contains the GreedyPhysical scheduler. For each scenario, 20 simulation runs of 20 minutes each (simulated time) have been performed. To skip the initial transient period, all average quantities have been computed only for the last 15 minutes of simulation.

Nodes are placed at grid points in a square of side $1400m$. For each simulation run, 10 nodes are randomly selected as gateways, and routing trees are formed as described in Section 3. Each node is equipped with a single 54 Mbps bidirectional link. Parameters for the physical interference model are as follows: $200mW$ transmission power, $-90dBm$ noise, $22dB$ SINR threshold. Radio signal propagation obeys log-normal signal propagation, with path-loss exponent 3 and variance $6dB$. Each slot (transmission opportunity) carries 512 application-level bytes. We used UDP as the transport protocol, which led to a 620 level-2 packet size. Slot duration has been dimensioned to 94 μsec, namely the minimum possible duration to prevent transmissions in a slot from interfering with the ones of the next slot (i.e., to let the last transmitted bit *leave* the network before the beginning of the next slot).

With these figures, negligible spatial reuse is possible (the average reuse over all simulations was 0.1%). Moreover, each packet reaches the destination gateway in one hop. Such a simple setup has been defined to let the results

[5] We stress that SUA is actually only an instance of the possible dynamic allocations strategy that can be used in combination with WoW.

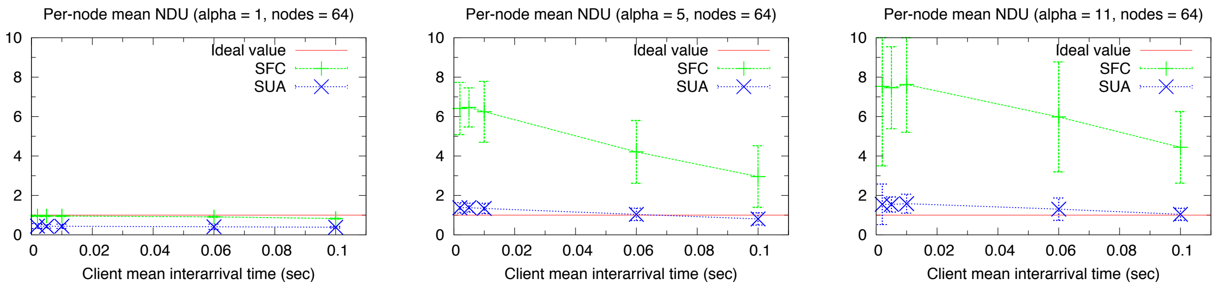
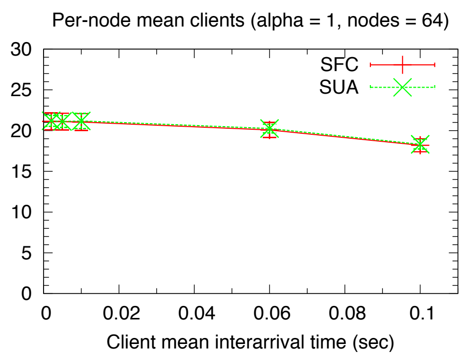
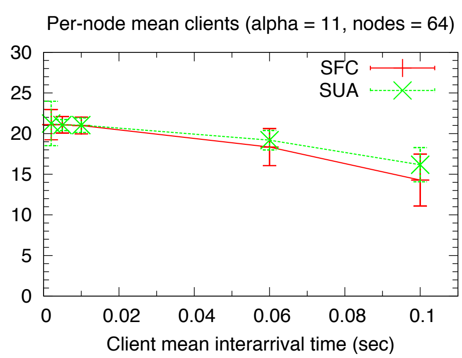


Figure 6. Mean NDU index (99% conf. interval) with $\alpha = 1$ (left), $\alpha = 5$ (center), and $\alpha = 11$) (right).



(a)



(b)

Figure 5. Mean number of clients (99% conf. interval).

be clearly due to the WoW framework, without any performance distortion due to unrelated factors such as diversity and spatial reuse. The expected performance with new technologies and/or new underlying schedulers should then be easy to compute. We leave as future work the evaluation of the WoW framework in a more challenging network setting.

As the reference QoS-sensitive application we chose VoIP: *arriving* clients are randomly associated to a *target* node, and ask the target node to be admitted to perform a VoIP call. For each simulation run, each node is randomly associated with a *popularity* index ranging from 1 to 10. The fact that different APs in a wireless network display different popularity levels is well documented in the literature [4, 5, 12]. Given two nodes n_i and n_j with popularities x_i and x_j, n_i's probability to be chosen as target node is $\frac{x_i}{x_j}$ times higher than the one of n_j. Node popularity is computed starting from the following *truncated power law* cumulative distribution $F(x) = x^{\frac{1}{\alpha-1}}$. Here α measures the *skewness* of the distribution, with $\alpha = 1$ meaning that all the nodes have the same popularity. α is a simulation parameter.

Client arrival times follow a Poisson distribution. The mean inter-arrival time τ is another simulation parameter. Once a client has been admitted, it remains *active* (i.e., the call lasts) for a random time interval, which is chosen according to an exponential distribution with mean equal to 2 minutes. Policies SFC and SUA are used to perform admission control.

The G.729 Cisco CoDec is used to simulate the VoIP call: each call requires 16 Kb/s (8 Kb/s per audio flow) application-level bandwidth, packets are 78 bytes long and carry a 40 bytes payload. As a simple compression scheme, as many VoIP packets as possible (namely, 6) are encapsulated in each UDP packet. The bandwidth required to accept a new client is computed according to this compression scheme.

Parameter ϑ in the WoW framework has been set to 5, and the whole set of different simulated scenarios is given by all the combinations of the following three values: number n of nodes in $\{36, 49, 64, 81, 100\}$, $\alpha \in \{1, 3, 5, 7, 9, 11\}$, client mean inter-arrival time $\tau \in \{0.01, 0.6, 0.1\}$ seconds (several extra points have been considered only for the 64 nodes case), SFC and SUA AC policy.

6.1 Application feasibility

To evaluate the effectiveness of the framework in implementing the target application, consider that a maximum delay and a maximum jitter of, respectively, $150ms$ and $50ms$ are recommended for a good quality VoIP call (ITU-T G.114 standard). Taking into account all the components of the end-to-end delay/jitter in a G.729 packet transmission [1], the mesh network is allowed to introduce a delay/jitter no higher than 35 ms. This constraint is largely fulfilled for

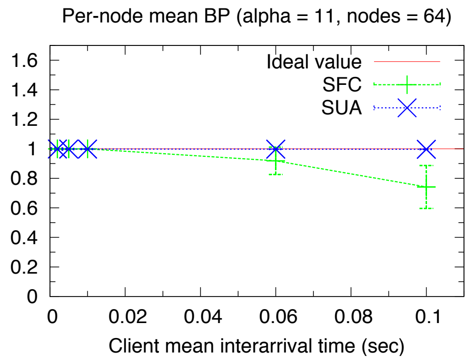


Figure 7. Mean BP index (99% conf. interval).

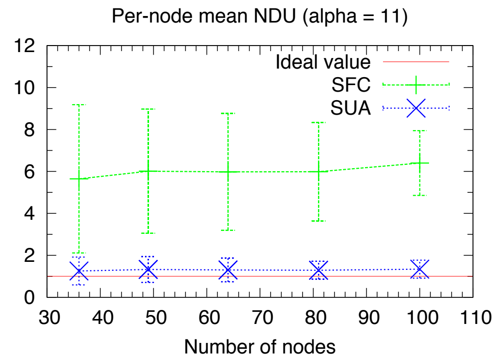


Figure 8. Mean NDU index (99% conf. interval) with mean interarrival time equal to $60ms$.

network sizes up to 64 nodes, because the maximum observed delay over all simulations for no more than 64 nodes was at most 27 ms. On the contrary 35 ms and 44 *ms* were observed, respectively, for 81 and 100 nodes.

6.2 Overhead and bandwidth utilization

The next important figure of merit is the overhead introduced by the framework, which is induced by the presence of the mini-schedule at each period. The bandwidth distribution profile broadcast to each node was coded on 20 bytes. The resulting maximum observed length of the mini-schedule over all simulations was 155 μsec. To measure the actual overhead, we can compare the recorded maximum total number of simultaneously active clients against the maximum number achievable in absence of overhead.

An optimistic 3% reduction of the schedule duration due to spatial reuse, and the hypothesis that each packet be delivered to the destination gateway in one hop would yield a slot rate of 10454 slots/sec. This would allow the transmission of up to 62729 G.729 packets/sec, i.e. the simultaneous service of 1225 clients. Of course, the maximum total number achieved in a simulation run depends both on how high the offered load is, and on how 'unlucky' the client-node association pattern is. I.e., it depends on how frequently a node with no free bandwidth is selected even in presence of not yet saturated ones.

The recorded maximum total number of simultaneously active clients over all simulations is 1200. It occurred in a run with 64 nodes, $\tau = 2ms$, $\alpha = 11$ and SUA. For all the simulation runs with $\tau \leq 10ms$, both the maximum and the mean total number of clients is higher than 1100. Postponing to the next subsection further investigation on the variance of the number of active clients, we can conclude that the overhead introduced by the framework is practically negligible, and that almost all the available bandwidth is fully utilized. Moreover, $\tau \leq 10$ ms happens to be a *saturation* load condition for the network.

6.3 Bandwidth distribution

Figure 5 shows the per-node mean number of active clients, and the associated 99% confidence interval, for $\alpha = 1$ and $\alpha = 11$, respectively. As can be seen, for $\alpha = 1$ both policies exhibit low variance around the mean value. In contrast, for $\alpha = 11$, the variance dramatically increases as the load gets away from saturation. This is a natural consequence of a highly skewed popularity distribution (intermediate values of the variance would be observed for the other values of α). The critical issue is whether, in presence of highly skewed popularity distributions, the framework still distributes the available bandwidth to each node proportionally to its popularity.

To investigate this point, for each node and for each period, we computed the following *normalized demand utilization* (NDU) index: ratio between the node demand during the period and the maximum possible demand (i.e., 5), divided by the ratio between the node popularity and the maximum possible popularity (i.e., 10). Of course, for each node the ideal value of this index should be 1 in stationary, full load condition.

Figures 6 (a), (b) and (c) show the mean value of the index, computed over all the nodes, and the associated 99% confidence interval, for 64 nodes and $\alpha = 1, 5, 11$. The accuracy of the static policy significantly degrades as α increases. This follows from the fact that, when it comes to admit or not a new client, each node must compute the expected bandwidth for the next round assuming that all the other nodes have maximum demand. This may lead to overly pessimistic predictions. On the other hand, SUA achieves a near optimal value of the index under all values of α.

To get a quantitative measure of this phenomenon, for each node and for each period, we have computed the following *bandwidth prediction* (BP) index: ratio between the node expected bandwidth in the next period, and the actual node bandwidth recorded in the next period. Figure 7 shows the mean value of the index, computed over all the nodes, and the associated 99% confidence interval, for 64 nodes and $\alpha = 11$. Also for the other values of α (plots not reported for brevity), SUA policy is very close to optimum, whereas SFC performance significantly degrades as α increases.

Nevertheless, even if at a considerable lower extent, the SUA policy deviates from the ideal distribution as well, especially for $\alpha = 1$ and $\alpha = 11$ (Figure 6). This deviation stems from the inaccuracies introduced by forcibly capping the next possible maximum demand of a node to its current value. A finer granularity should be recovered by periodically pushing up all the weights according to (3). However, from Figure 6 (a) it can be deduced that the mean node demand ends up being quite low, thus coarsening the granularity of the possible node demand values. This fact progressively degrades the performance as the skewness increases (Figures 6 (b) and (c)). Basically, due to the coarse demand granularity, low popularity nodes unfairly *steal* more bandwidth than needed.

Finally, to show the accuracy of the framework in distributing the network bandwidth for the different network sizes, the NDU index for $\alpha = 11$ and $\tau = 60ms$ (the configuration for which both SUA and SFC showed the worst NDU values) is reported as a function of the number of nodes in Figure 8.

7 Conclusions

In this paper, we have introduced the WoW framework for providing strong QoS guarantees in wireless mesh networks. To complement the proposed framework, we have proposed two simple backbone node-level AC policies that stand at opposite ends of the spatial fairness/bandwidth utilization tradeoff. Through extensive simulations, we have verified WoW's effectiveness in providing strong QoS guarantees to VoIP users in a realistic mesh network scenario. Furthermore, we have shown WoW's ability to provide these guarantees and to effectively use network bandwidth even in presence of highly unbalanced network load. To the best of our knowledge, WoW is the first framework for QoS provisioning in mesh networks with similar features proposed in the literature.

The work presented in this paper leaves several issues open for further research, such as the definition of more sophisticated backbone node-level policies. In particular, AC policies for dynamic bandwidth allocation allowing a better weight management than SUA are needed.

References

[1] http://www.cisco.com/en/US/tech/tk652/tk698/-technologies_white_paper09186a00800a8993.shtml-#buildingdelaybudget

[2] http://www.ece.gatech.edu/research/labs/-MANIACS/GTNetS/

[3] B. Aoun, R. Boutaba, Y. Iraqi, G. Kenward, "Gateway Placement Optimization in Wireless Mesh Networks with QoS Constraints", *IEEE JSAC*, to appear.

[4] A. Balachandran, G.M. Voelker, P. Bahl, P. Venkat Rangan, "Characterizing User Behavior and Network Performance in a Public Wireless LAN", *Proc. ACM Sigmetrics 02*, pp. 195–205, 2002.

[5] M. Balazinska, P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network", *Proc. ACM MobiSys 03*, pp. 303–316, 2003.

[6] N. Ben Salem and J.P. Hubaux, "A Fair Scheduling for Wireless Mesh Networks", *Proc. IEEE WiMesh*, 2005.

[7] Y. Bejerano, "Efficient Integration of Multi-hop Wireless and Wired Networks with QoS Constraints", *IEEE/ACM Trans. on Networking*, Vol. 12, n. 6, pp. 1064–1078, 2004.

[8] G. Brar, D. Blough, and P. Santi, "Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks," *Proc. ACM Mobicom*, pp. 2–13, 2006.

[9] D. Dharmaraju, A.R. Chowdhury, P. Hovareshti, J.S. Baras, "INORA: A Unified Signalling and Routing Mechanism for QoS Support in Mobile Ad Hoc Networks", *Proc. ICPPW 2002*, pp. 86–93, 2002.

[10] V. Gambiroza, B. Sadeghi, E.W. Knightly, "End-to-End Performance and Fairness in MultiHop Wireless Backhaul Networks", *Proc. ACM Mobicom*, pp. 287–301, 2004.

[11] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory,* Vol. 46, No. 2, pp. 388–404, 2000.

[12] D. Kotz, K. Essien, "Characterizing Usage of a Campus-Wide Wireless Network", *Proc. ACM Mobicom*, pp. 107–118, 2002.

[13] P. Lassila, A. Penttinen, J. Virtamo, "Dimensioning Wireless Mesh Networks with Flow-Level QoS Requirements", *Proc. ACM PE-WASUN*, pp. 17–24, 2006.

[14] S.B. Lee, A. Gahng-Seop, X. Zhang, A.T. Campbell, "INSIGNIA: An IP-based Quality of Service Framework for Mobile Ad Hoc Networks", *Journal of Parallel and Distributed Computing*, Vol. 60, n.4, pp. 374–406, 2000.

[15] C.R. Lin, J. Liu, "QoS Routing in Ad Hoc Wireless Networks", *IEEE JSAC*, Vol. 17, n. 8, pp. 1426?1438, 1999.

[16] C.R. Lin, "On-demand QoS routing in Multihop Mobile Networks", *Proc. IEEE Infocom*, pp. 1735–1744, 2001.

[17] L. Luo, M. Gruteser, H. Liu, K. Huang, S. Chen, "A QoS Routing and Admission Control Scheme for 802.11 Ad Hoc Networks", *Proc. ACM Diwans*, pp. 19–27, 2006.

[18] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service", *Proc. European Wireless*, pp. 32–39, 2002.

[19] T.B. Reddy, I. Karthigeyan, B.S. Manoj, C. Siva Ram Murthy, "Quality of Service Provisioning in Ad Hoc Wireless Networks: a Survey of Issues and Solutions", *Ad Hoc Networks*, Vol. 4, n. 1, pp. 83–124, 2006.

[20] H. Shetiya, V. Sharma, "Algorithms for Routing and Centralized Scheduling to Provide QoS in IEEE 802.16 Mesh Networks", *Proc. ACM WMuNeP*, pp. 140–149, 2005.

[21] S. Sheu, T. Sheu, "DBASE: A Distributed Bandwidth Allocation/Sharing/Extension Protocol for Multimedia over IEEE 802.11 Ad Hoc Wireless LAN", *Proc. IEEE Infocom*, pp. 1558–1567, 2001.

[22] P. Sinha, R. Sivakumar, V. Bharghavan, "CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm", *IEEE JSAC*, Vol. 17, n. 8, pp. 1454–1466, 1999.

[23] S. Sriram, T.B. Reddy, B.S. Manoj, C. Siva Ram Murthy, "On the End-to-end Call Acceptance and the Possibility of Deterministic QoS Guarantees in Ad Hoc Wireless Networks", *Proc. ACM MobiHoc*, pp. 169–180, 2005.

[24] R. Zhao, B. Walke, "Decentrally Controlled Wireless Multi-Hop Mesh Networks for High Quality Multi-media Communications", *Proc. ACM MSWiM*, pp. 200–208, 2005.