

A Practical View on Quality-of-Service Support in Wireless Ad Hoc Networks

Michael Gerharz, Christian de Waal, Matthias Frank
Institute of Computer Science IV, University of Bonn
Römerstr. 164, D-53117 Bonn, Germany
{gerharz,dewaal,matthew}@cs.uni-bonn.de

Paul James
Nokia Research Center
Meesmannstr 103, D-44807 Bochum, Germany
paul.james@nokia.com

Abstract

Whereas routing protocols for mobile wireless ad hoc networks are well advanced, the support of Quality-of-Service in such networks has only recently emerged as a major research topic. Several proposals combine routing concepts with mechanisms for QoS support, at the same time making assumptions as have been made for the QoS support in wired networks. This paper examines the practical applicability of approaches to QoS with respect to the differences between wired and wireless ad hoc networks. Furthermore, a novel approach for service differentiation in wireless networks is proposed and early simulation results on the performance are presented.

1. Introduction

Substantial progress has been achieved in solving the routing challenge in mobile wireless ad hoc networks. The *Ad hoc On demand Distance Vector* protocol (AODV) and the *Dynamic Source Routing* protocol (DSR) [23] are among the most prominent ad hoc routing protocols. These protocols provide a basic routing functionality that is sufficient for conventional applications such as file transfer or e-mail download. However, ad hoc networks are also an interesting platform for more demanding applications such as Voice over IP (VoIP), which are very susceptible to larger delays, jitter, and packet losses. In order to support such applications, it is not sufficient to provide a basic routing functionality alone.

Several proposals for routing schemes exist that are supposed to find routes fulfilling certain QoS demands of applications. In these, many assumptions have been adopted from wired networks. This article discusses the fundamental differences between wired networks and wireless ad hoc networks which are important for QoS provisioning.

A fundamentally different aspect of Quality-of-Service support in mobile wireless ad hoc networks is that the characteristics of the routing should meet the demands of real-time applications to the largest extent possible.

One approach for improving the quality of communication sessions is to enhance the quality of the route selection process. *Associativity Based Routing* (ABR) [28] and *Signal Stability Adaptive Routing* (SSA) [12] both try to establish *stable routes*, i.e. routes that have a high probability of being available for a long period of time. [14] provides a detailed analysis of the implications of different mobility models on the stability of links.

A somewhat complementary approach to provide more robust connections is taken by *Preemptive Routing* [15] and *Routerlifetime Assessment Based Routing* (RABR) [2] which try to detect a link break before it actually happens in order to issue a new route discovery before the old route breaks.

Another approach to achieve robust connections is multipath routing, where several alternative paths are established at once. Upon a link break, this approach has the advantage to have a fallback route at hand. The *Ad Hoc On-demand Multipath Distance Vector* protocol (AOMDV) [22] and the *Split Multipath Routing* protocol (SMR) [19] represent this approach.

However, the scope of this paper is on a separation of QoS provisioning and the routing strategy in use. In present approaches, the most important challenges of QoS support are to acquire the available bandwidth in an ad hoc network and to maintain accurate values with dynamics of such a

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) as part of the IPonAir project (<http://www.IPonAir.de/>).

network. In opposition to this, our concept uses relative service differentiation without the need of information on available bandwidth.

The rest of this paper is structured as follows: The following section illustrates relevant bandwidth reservation issues for both wireless and wired networks, showing the difficulty of determining the available bandwidth of wireless links in an ad hoc network. Section 3 highlights the advantages of service differentiation for ad hoc QoS support and section 4 introduces our approach for a lightweight provision of relative Quality of Service between several service classes. This section also presents early simulation results on the performance without and with different possible variants of service differentiation. Finally, section 5 concludes the paper and gives an outlook on future work.

2. Bandwidth Reservation in Wired and in Wireless Networks

Quality-of-Service approaches in wired networks rely on the possibility to make bandwidth reservations, e.g. Integrated Services (IntServ) [9] provides guaranteed bandwidth for flows while Differentiated Services (DiffServ) [7] provides hard guarantees for service classes.

Applying these concepts to wireless ad hoc networks is difficult, because many assumptions in wired QoS approaches do not hold in wireless networks. This section takes a closer look at those assumptions and discusses fundamental differences between wired networks and wireless ad hoc networks which affect QoS provisioning.

2.1. Existing Approaches for Wireless Networks

Inspired by common techniques for Quality-of-Service provisioning found in the wired Internet, several researchers have proposed a bandwidth reservation scheme for wireless ad hoc networks. These approaches basically split into two groups, those that demand a tight integration of QoS provisioning into the routing protocol and those that try to be independent of the underlying routing protocol. The *Core-Extraction Distributed Ad hoc Routing* protocol (CEDAR) [27], the MMWN (*Multimedia support for Mobile Wireless Networks*) [24] protocol, and ticket based probing [10] are examples for the first category while INSIGNIA (*In-band Signalling for QoS in Ad-Hoc mobile networks*) [18] belongs to the latter.

The idea of CEDAR, MMWN, and ticket based probing is to distribute link state information (which in MMWN may be an abstraction over links inside of station clusters) to enable other stations to find routes fulfilling certain QoS criteria, e.g. a minimum bandwidth. In contrast, INSIGNIA is a signalling protocol that piggybacks resource reservations onto data packets, which can be modified by intermediate

stations to inform the communication endpoints in case of lacking resources.

The central idea in all of these approaches is that the links between stations have certain QoS related properties, in particular a known amount of available bandwidth, and that stations are able to give guarantees for traffic traversing these links. Throughout the rest of this section, we take a closer look at the characteristics of wired and wireless networks to verify if these assumptions are actually met.

2.2. Bandwidth Reservation in Wired Networks

Guaranteeing a certain amount of bandwidth for a certain flow or service class requires that the station providing that guarantee is in control of that bandwidth. This is certainly the case in a wired network with full-duplex point-to-point links (e.g. a switched Ethernet).

It is also possible to agree on a determined share of bandwidth in a shared wired medium. All stations within this domain are able to communicate with each other directly and whenever one station transmits data to another station, all the other stations on the medium are aware of that. Thus, a wired network with a shared medium is a well defined, closed collision domain. If all the stations on the shared medium collaborate, it is possible to elect a station that centrally manages the available bandwidth within their domain. [31] introduces the *Subnet Bandwidth Manager* (SBM) concept for that purpose.

Since a wired network is comprised of well-defined subnetworks, bandwidth guarantees for flows or service classes can be met by enforcing them in every involved subnetwork.

2.3. Bandwidth Reservation in Wireless Networks

The situation is completely different for wireless ad hoc networks consisting of devices with a single network interface. Networks consisting of devices with multiple network interfaces (such that each interface handles one link exclusively) could possibly overcome the drawbacks we discuss in the following, but also, new challenges would have to be solved. Discussing such ad hoc networks is beyond the scope of this article, since ad hoc networks in the near future will most certainly consist of devices with a single network interface (or perhaps a few interfaces for different radio technologies).

Wireless ad hoc networks can be based on two different MAC technologies. With a *single-channel* protocol (e.g. IEEE 802.11 [16]), all stations communicate on the same channel and therefore potentially interfere with each other. With a *multi-channel* protocol in contrast (e.g. Bluetooth [8] or CDMA [25]), stations can communicate on several channels (“piconets” in Bluetooth terminology) simultaneously. Note that this theoretical assumption holds in our definition

of multi-channel networks despite the inter-piconet interference in Bluetooth. A station can only be active in a single channel at any given point in time.

For a multi-channel MAC, any two devices that are within each others transmission range could form a closed collision domain. Transmissions from different domains are separated from each other by assigning a different channel to each of them. The fundamental difference to wired networks is that a station does not have a separate interface for each subnetwork it participates in. This means that the devices have to switch channels regularly, leading to the absence of a well-defined, fixed subnetwork structure.

In the single-channel case, the attempt to identify collision domains fails altogether. These domains would span entire connected components of the ad hoc network, since any two neighbours belong to the same collision domain. However, it depends exactly on the sender-receiver pair which devices are potential interferers.

Both of the discussed cases have in common that a bandwidth reservation mechanism requires a *transmission schedule* defining time slots which take their turns periodically. For each slot, its duration and a set of possible simultaneous transmissions must be defined. This need for a transmission schedule is the fundamental limitation in contrast to wired networks. The problem of finding an optimal schedule is even NP-complete [32].

Several proposals have been made for distributed Bluetooth scatternet scheduling [4], but this task is very difficult and none of these approaches is able to make guarantees. The requirement that devices need to be synchronised is particularly difficult to realise practically, because device clocks generally drift against each other considerably.

For single-channel MAC protocols, the scheduling component is not as critical, because stations do not have to decide when to listen on which channel. MACA/PR (*Multiple Access Collision Avoidance with Piggyback Reservations*) [21] is an IEEE 802.11-like MAC layer that includes a mechanism to reserve the channel for certain periodic time windows. Scheduling is provided implicitly by the need to reserve non-overlapping time spans for transmissions of neighbouring stations. In [32], a QoS routing protocol is introduced that creates such a schedule, but again, this requires a global synchronisation of the stations.

2.4. An Example

To illustrate difficulties in making bandwidth reservations in wireless ad hoc networks with both single- and multi-channel MAC layers, observe the example topology in figure 1. Neighbours in this topology are able to communicate with each other directly, and we make the idealised assumption that transmissions can interfere only at neighbouring nodes.

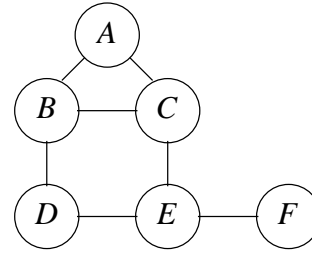


Figure 1. Example scenario

First, we assume a multi-channel MAC layer. With such a MAC layer, any two neighbouring devices can communicate, but each device cannot participate in more than one transmission. A first flow requesting one third of the available link bandwidth originates at A and is routed through C and E to F . This is accomplished by a schedule with three time slots of equal length, with transmission $A \rightarrow C$ in the first time slot, transmission $C \rightarrow E$ in the second time slot, and transmission $E \rightarrow F$ in the third. Note that stations C and E must reserve twice the requested bandwidth, because they must handle both reception and forwarding of the data. Now a second flow requesting the same bandwidth originates at D and is routed through B to A . Assume that the transmission $D \rightarrow B$ is scheduled in the second time slot and the transmission $B \rightarrow A$ in the third. This results in a situation where both B and C still have one third of their bandwidth available, yet they are unable to communicate with each other without rescheduling other transmissions. While in this simple example, it suffices e.g. to reschedule the transmission $B \rightarrow A$ from the third to the first slot, one can easily construct more complex examples with extensive dependencies that would require rescheduling transmissions over much greater distances. In general, rescheduling is a complex task and might have unforeseeable consequences when performed distributedly. Note that a valid schedule might not even exist.

Now, assume a single-channel MAC layer similar to IEEE 802.11. Transmissions are further constrained by the requirement that a receiver cannot be the neighbour of a different transmission's sender. This makes things even more complicated. Say the schedule of the first flow is the same as in the previous example. In this situation, the requested bandwidth is already the maximum bandwidth available on this route, because transmission $E \rightarrow F$ prevents C from receiving another transmission.

Now suppose again that a second flow from D to A requests a third of the link bandwidth. Transmission $D \rightarrow B$ can only be scheduled in the third slot. However, the transmission $B \rightarrow A$ cannot take place in any of the time slots:

The first flow consumes not only the entire bandwidth on its own path, but also on links that are not even involved in this flow. The serious performance degradation of multihop communication with IEEE 802.11 is analysed in [20].

Note that we have yet made idealised assumptions: In reality, the transmission $E \rightarrow F$ might corrupt a simultaneous transmission $D \rightarrow B$ even though E cannot directly communicate with B . This can happen because the strength of E 's signal at B is too weak for a successful decoding, but strong enough to jam D 's signal. This possible interference beyond the transmission range results in a further reduction of capacity.

The discussion shows that the bandwidth available on a certain link depends on many factors. An intermediate node reserving bandwidth for some flow must take into account at least the reception of this flow's packets, and in case of a single-channel MAC also transmissions elsewhere on this route, including at least the transmission on the next hop. The available bandwidth on a certain link does not only depend on this link's own activity, but also on the activity of other links in its vicinity. The two neighbouring stations must have this bandwidth available at common time spans, or they must initiate some far-reaching rescheduling of transmissions. Therefore, if stations wanted to learn the amount of bandwidth available on their links, some kind of scheduling scheme would explicitly have to be implemented. In single-channel networks, it is a further open question how a station would learn about other stations that are not within transmission range, but that can potentially interfere with packet receptions.

Up to now, we have considered static scenarios. Be reminded that in a dynamic environment, a new scheduling scheme has to be found for every single topology change, where previous reservations become unsustainable, and the available bandwidth will be altered on links that are not even in proximity of the topology change.

We can conclude that the available bandwidth of links in wireless ad hoc networks is very difficult to determine. The previous work on QoS provisioning in wireless ad hoc networks introduced above does not address this problem, because it implicitly assumes a link concept as in wired networks.

3. Service Differentiation

Apart from the general difficulties outlined in section 2, there is no chance to integrate classical QoS (based on bandwidth reservations) in wireless ad hoc networks which are to be deployed with off-the-shelf hardware in the near future. And in the face of the complications elaborated above, it can be seriously doubted that it will ever be worthwhile to implement bandwidth reservation mechanisms in mobile wireless ad hoc networks as long as a single network inter-

face is used to communicate on several links. As a consequence, IntServ [9] based approaches are not applicable in wireless ad hoc networks.

As an alternative to IntServ, the DiffServ approach [7] has been developed for the wired Internet world for scalability reasons. In this approach, flows are classified into several service classes whose packets are treated differently at the routing nodes. As opposed to the wired Internet, it is not possible to provide a hard separation of different service classes in ad hoc networks for the reasons outlined in section 2, but relative prioritisation is possible in such a way that traffic of a certain class is given a higher or lower priority than traffic of other service classes.

In third generation wireless telecommunication systems, a distinction of four traffic classes with different characteristics and demands has proven to suit a wide range of requirements [1]:

- The *conversational class* is intended for traffic with a stringent and low delay demand, e.g. voice- or video-telephony, or video games.
- The *streaming class* is intended for traffic without demand on the delay itself, but on the delay variation, e.g. multimedia streaming.
- The *interactive class* is intended for traffic that only has a soft delay constraint, e.g. web browsing or network games.
- The *background class* is intended for traffic that does not have any time constraints, e.g. a background file or e-mail download.

The conversational and the streaming class are provided with hard delay as well as minimum bandwidth guarantees. Furthermore, the packets of the interactive class are transmitted with a higher priority than packets of the background class. Thus, these service classes split into two categories, one providing "hard" guarantees and the other only "soft".

As argued above, hard guarantees are impractical in wireless ad hoc networks. Thus, a different classification is required. One possibility would be to divide the traffic into delay sensitive and insensitive applications, in other words *realtime* traffic and *bulk* traffic. Certainly, the conversational class would belong to the realtime class and the background class would belong to the bulk traffic class.

Realtime traffic should be given priority in case of network congestion. On the other hand, if a realtime packet suffers large delays, it may be dropped already by an intermediate node in order to save resources.

A third traffic class corresponding to the interactive class is imaginable. This could be implemented by giving its packets a higher priority than bulk packets without dropping them in case they are delayed in intermediate stations.

This way, the soft delay constraint is supported by the prioritisation over the bulk packets.

Some work already exists that is based on service differentiation rather than resource reservations, e.g. SWAN (*Stateless Wireless Ad hoc Networks*) [3] and FQMM (*Flexible QoS Model for Mobile Ad hoc Networks*) [29].

SWAN is a stateless protocol that applies a distributed rate-control algorithm in conjunction with a source-based admission control scheme to prioritise realtime traffic. It is independent of the routing protocol and does not rely on the MAC layer to support Quality-of-Service. In SWAN, realtime traffic is admitted a certain maximal amount of bandwidth. Forwarding nodes try to regulate best-effort traffic such that together with the admitted realtime traffic, the link capacity is optimally utilised. Realtime traffic is admitted based on probes that record the minimum residual realtime bandwidth along the discovered source-destination path.

FQMM aims at a flexible service differentiation by allowing an arbitrary number of service classes. A class is explicitly allowed to consist of only a single flow. The authors describe FQMM in close relation to DiffServ. The QoS policies are individually defined at each source station, which plays the role of an ingress node for its own packets: It classifies and meters them, and marks them accordingly. The source and intermediate stations perform traffic shaping according to those marks; a priority buffer scheme and a priority scheduling scheme are suggested for that purpose by the same authors in [30].

The next section will take a deeper look on the traffic shaping and policing strategies of these two protocols and compare them with an alternative approach.

4. A Lightweight Approach to Service Differentiation in Wireless Networks (DLite)

This section introduces a lightweight approach to service differentiation (*DLite*), discusses the capabilities and limitations of several design alternatives (sec. 4.1, sec. 4.2), and provides early simulation results (sec. 4.4).

The basic idea is to have a predefined set of service classes which are defined by their relative bandwidth fraction and delay bounds. No absolute bandwidth guarantees are provided due to the reasons outlined in section 2.

The DLite algorithm is summarised in figure 2. Traffic flows are classified into service classes each of which is buffered in a separate queue. Prioritisation is achieved with a token balancing scheme similar to existing fair queueing strategies, as described in section 4.1. To save bandwidth, late packets of service classes with delay constraints are already dropped in intermediate routers (see section 4.3).

A main advantage of this algorithm is its simplicity: It is easy to implement, the computational overhead is small, and it is universally applicable, because there is no need for

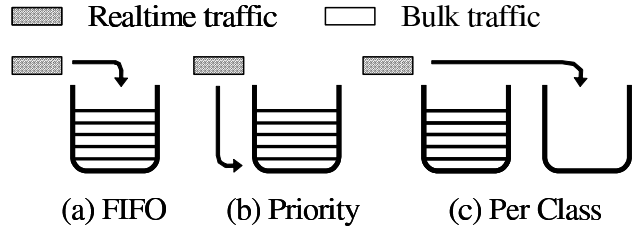


Figure 3. Effect of different queueing strategies on the delay of realtime packets

the devices to operate in promiscuous mode, it is applicable together with any routing protocol, be it single-path or multi-path, and finally, it is not even required that every station implements this approach, although with a rising number of unaware stations, the impact declines.

Furthermore, DLite implements a philosophy that is quite different from that of previous approaches in that no effort is made to conserve the conditions for established connections. Rather, adaptive applications may react to quality degradations by increasing their compression rate, and unimportant connections will be terminated with a higher probability than important ones. This is detailed in section 4.2.

4.1. Priority Queueing

An important factor in service differentiation is the queueing strategy. Several approaches have been proposed in the literature.

In [30], two prioritisation schemes for FQMM are proposed. Firstly, a simple priority queue ensures that high-priority packets are given unconditional preference over low-priority packets. Secondly, they consider a FIFO queue which they enhance with a RIO buffer management (Random Early Discard with IN/OUT).

SWAN also conceptually utilises a priority queue, but limits the amount of realtime traffic in order to protect the lower-priority traffic from starvation.

In our approach, we implement a separate queue for each traffic class. The queues are scheduled according to a token bucket scheme as described later in this section. Buffer space which is not occupied by a service class may be assigned to other classes. However, this additional buffer space must be released, if new packets of the original class arrive.

All of these three queueing approaches have different advantages and disadvantages related to the scheduling of high-priority traffic.

With a FIFO queue it might occur that a high-priority packet is scheduled after a burst of low-priority traffic even

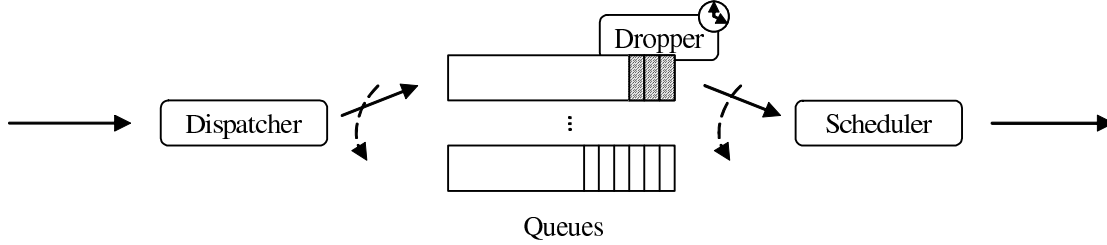


Figure 2. Overview of the DLite algorithm

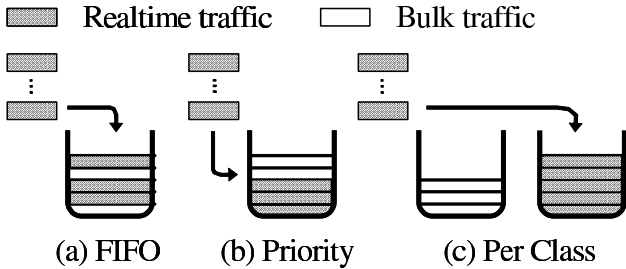


Figure 4. Effect of different queueing strategies on the bandwidth share of bulk traffic

if it is the only high-priority packet currently in the queue, regardless of whether a drop-tail queue or a more sophisticated approach like RIO is used for traffic shaping. Thus, a high-priority packet may suffer large delays due to an already filled up buffer, regardless of the class priority. This is illustrated in figure 3.

With a priority queue, this is obviously prevented. A high-priority packet is scheduled before any lower-priority packets. Thus, its delay depends solely on the amount of high-priority packets currently in the queue and the medium contention in the node's neighbourhood.

The same holds if per-class queues are used, as long as the amount of high-priority traffic remains below its designated share, since the high-priority packets are queued into their own queue, separate from the bulk packets.

On the other hand, a priority queue might result in a starvation of low-priority traffic which is illustrated in figure 4. When the amount of high-priority traffic is exceptionally high, low-priority traffic might not be transmitted at all or at a very low rate, because all the high-priority traffic is scheduled ahead of any low-priority traffic. Both SWAN and FQMM solve this problem by admitting only a predefined amount of high-priority traffic to be injected into the network. (Be reminded that FQMM differentiates in-profile and out-of-profile traffic.)

With FIFO and per-class queues, low-priority traffic will

receive a minimum share of bandwidth in any case due to an interleaving of low-priority and high-priority packets.

We now describe how packets are scheduled in our approach. Each traffic class has a balance of tokens, and the higher the balance of a class is, the higher is also its priority when it comes to dequeue the next packet for transmission. For each transmission of a packet of class i , an amount of $c_{b,i} = w_i \cdot (f + b)$ tokens is subtracted from the classes token balance and an equal fraction thereof is added to every other classes balance such that the sum of all tokens is always zero. In this formula, the weight values w_i control the fraction of the bandwidth assigned to the different classes, and f is a constant value that accounts for the MAC overhead which is considerable e.g. in the case of IEEE 802.11. Note that a higher weight value corresponds to a lower bandwidth fraction, since this weight is multiplied to the cost. Of course, it should be taken care that the balance of a traffic class stays within a certain interval. The size of this interval together with the values w_i determines the maximal length of a burst of traffic from one class.

Actually, the described method is rather a fair queueing than a prioritisation approach, but in this context, it has a very favourable characteristic: If the amount of traffic of a certain class is low enough in contrast to that of other classes, then its packets are always prioritised. This means that as long as the amount of delay-sensitive traffic does not grow too large, it is forwarded as quickly as possible, and if it does grow too large, starvation of other traffic classes is prevented. Setting the upper bound of a classes token balance depending on its delay-sensitivity is a further way to tune the described method.

A possible alternative might be Weighted Fair Queueing [11] using a sufficiently large δ , but a requirement of this algorithm is that the effective data rate is known, which we have previously shown to be very difficult. The token balance scheme described is somewhat similar to Deficit Round Robin [26], which might be used to achieve a similar prioritisation effect by a slight modification (not resetting a classes deficit counter upon emptying its queue), but this also requires clever choice of the absolute quantum sizes, whereas only the relation of the weights is important with

the token balance scheme. Here, the prioritisation of low-volume traffic is simply more straightforward.

4.2. Dealing with Congestion

FQMM tries to limit network congestion by policing the traffic at the traffic sources. The sources are the equivalent of ingress routers in DiffServ networks. To regulate the traffic, a source node implements a token bucket which determines whether a packet is in-profile or out-of-profile.

The source stations have to take great care in regulating their traffic. If the rate of in-profile packets is not chosen properly in areas with little activity, it might accumulate and cause congestion in bottleneck areas. In contrast to classical DiffServ, FQMM actually lacks service level agreements. For the token bucket metering suggested by the FQMM authors, it remains an open question how to calculate the dynamic parameter C_t that expresses the share of the effective link bandwidth at time t ; we have already shown how the bandwidth available for a certain stream depends on the network topology and on other traffic streams in a much more complex way than in wired networks. A single topology change can alter the available bandwidth significantly.

SWAN uses a strict admission control scheme for real-time packets. Bulk-traffic is unconditionally admitted, but only receives the remaining resources. Realtime traffic is admitted by the source node depending on the outcome of probing the network for resources. If the probe packet passes a link on which the total amount of realtime traffic exceeds a certain threshold, the session will not be admitted. This way, it is prevented that the realtime traffic crowds out the bulk traffic. Furthermore, it prevents that realtime traffic suffers too large delays.

This approach has the advantage that realtime flows have decent chances of having a good quality once they are admitted. However, the admission control scheme might reject flows although the remaining capacities would have sufficed. Thus, existing sessions are given priority over new sessions in order to protect them against a quality degradation caused by network congestion.

A different notion is to give every user the same chances to establish a high-priority connection, regardless of who was first to start his session, i.e. to admit every session request. Again, care must be taken to not crowd out lower-priority traffic. As a consequence, high-priority traffic should not be given unconditional priority over low-priority traffic. This is taken care of in our per-class queueing scheme described in the previous section.

Since our approach does not restrict the high-priority traffic, it results in a quality degradation of high-priority flows as their volume increases. In effect, *all*, and not just single users suffer from an excess of high-priority traffic. It is however likely that a regulation of the number of ac-

tive sessions will result from users that terminate their high-priority connections because they are not willing to bother with the bad transmission quality. Furthermore, it is likely that multimedia applications change their coding scheme to a higher compression when they experience many late or lost packets.

This can be seen from a fairness aspect especially if one assumes that in this way, unimportant connections will be terminated before the important ones.

Both approaches, protecting existing sessions or admitting every session request, have their advantages and disadvantages and which one fits to the requirements better depends on the specific scenario.

4.3. Dealing with Excessive Delays

Certain applications have stringent delay bounds for their traffic. This means that packets arriving too late are useless. From the application's point of view, there is no difference between late and lost packets.

This implies that it is actually useless to forward real-time packets that stay in a router for more than a threshold amount of time, because they will be discarded at the destination anyway. Dropping those packets instead has the advantage of reducing the load in the network.

It would also be possible to accumulate the delay a packet has experienced in each router by adding this information in the packet header, but this would make the approach more complicated. Nevertheless, it could be verified in future work if this is worth the effort. For now, only the delay in each intermediate station is considered separately.

This is another component of our service differentiation approach.

4.4. Preliminary Simulation Results

The previously introduced service differentiation scheme has been implemented in the ns-2 simulator [13] as part of a new implementation of the AODV routing protocol conforming to Internet Draft version 12. This section presents early results of our evaluation of the DLite algorithm.

The traffic consisted of realtime and bulk packets. Realtime traffic was modelled as VoIP phone calls by bidirectional CBR sessions with a data rate of 9.6 Kbit/s. The inter-arrival time of calls was exponentially distributed with a mean of 7.5 seconds (high amount of realtime traffic) or 10 seconds (low amount of realtime traffic). The length of a call was modelled according to a lognormal distribution with $\mu = 3.287$ and $\sigma = 0.891$, resulting in an average call length of about 40 seconds. This distribution was found to model observed channel holding times in a cellular network well [5].

Bulk traffic was modelled as the transmission of a random amount of data with TCP NewReno, uniformly distributed between 100,000 and 5,000,000 bytes. The time between the initiation of data transfers was exponentially distributed with a mean of 25 seconds (high amount of bulk traffic) or 30 seconds (low amount of bulk traffic).

For both traffic types, sources and destinations were chosen uniformly distributed from all nodes.

The mobility scenario consisted of 200 stations moving on a $1000 \times 1000 \text{ m}^2$ area according to the Random Waypoint model [6] with a speed uniformly distributed between 0.5 and 1.5 m/s and a pause time uniformly distributed between 0 and 300 seconds. The stations had a common transmission range of 200 metres. IEEE 802.11b was simulated as the MAC layer.

The service differentiation strategies simulated were:

- No service differentiation.
- Differentiation of realtime and bulk traffic, with $w_{\text{realtime}} = w_{\text{bulk}} = 1$.
- Differentiation of realtime and bulk traffic, with $w_{\text{realtime}} = w_{\text{bulk}} = 1$ and discarding of realtime packets queued for more than 0.1 seconds.
- Differentiation of realtime and bulk traffic, with $w_{\text{realtime}} = 1$, $w_{\text{bulk}} = 10$ and discarding of realtime packets queued for more than 0.1 seconds.

In all simulations, routing protocol packets were given unconditional priority before other packets.

As previously noted, a delay of over 150 ms in a voice transmission is felt as disturbing by most users, and a delay above 250 ms is felt as unbearable [17]. Due to the unfavourable characteristics of wireless ad hoc networks, only realtime packets exceeding this maximum delay of 250 ms (including packetisation) are considered to have arrived too late and thus are dropped.

The fractions of lost (or late) realtime packets are plotted in figure 5(a) for a higher amount of realtime traffic (IAT 7.5s) and in figure 5(b) for a lower amount (IAT 10s). We see that with a higher amount of bulk traffic, the differentiation schemes have a great impact on these results. The simple differentiation alone already greatly reduces the loss of realtime packets. We even observe that it also increases the bulk traffic's overall throughput (figure 6). Discarding realtime packets that are delayed by more than 100 ms in an intermediate station's queue additionally saves resources that benefit other packets, further reducing the overall loss of realtime packets.

An interesting observation is that assigning a higher weight (and thereby a lower bandwidth fraction) to bulk traffic does not have the desired effect to reduce the loss ratio of realtime packets at the expense of the bulk traffic's

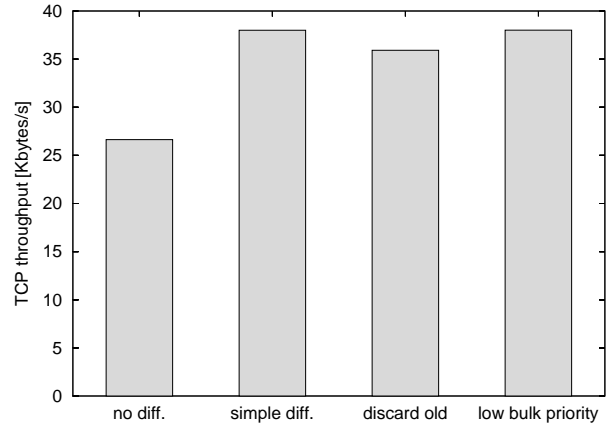


Figure 6. Bulk throughput with a high amount of realtime and bulk traffic

throughput. One cause for this might be that the presented service differentiation approach cannot completely separate the traffic classes due to the following reason. The approach is capable of providing service differentiation within a single station. However, it fails to provide it across neighbouring stations sharing the same medium and thereby the effective bandwidth. Suppose, a high amount of bulk data and a low amount of realtime data are all routed through the same station. After this station receives a realtime packet, this packet will be prioritised, and the other bulk packets are delayed. But the bulk packets routed through neighbouring stations are not delayed, and the realtime packet has to contend for the medium access with several bulk packets. This may also be the reason for the significantly smaller loss rate of realtime traffic with a lower amount of bulk traffic.

This shows that merely scheduling packet transmissions within a station has only a limited power, though it does achieve much better results than without any service differentiation. To achieve even better results, transmission scheduling between neighbouring stations would have to be implemented. This problem is addressed by the rate controller component in SWAN.

On the other hand, every approach to service differentiation is limited by the mobility of nodes in a wireless ad hoc network. Route failures caused by the mobility account for a minimal number of packet losses which cannot be reduced by any of these approaches.

5 Conclusions & Further Work

In this paper, we have taken a practical view on the Quality-of-Service capabilities of wireless ad hoc networking technologies. We have outlined that it is a very complex task to reserve or even measure available bandwidth in wire-

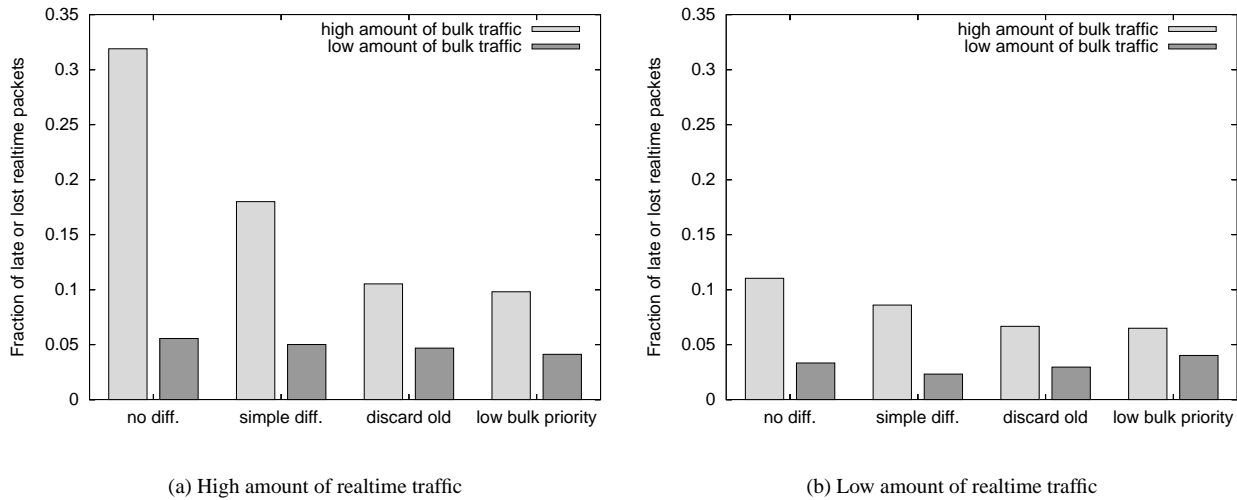


Figure 5. Lost (or late) realtime packets

less multihop networks. With contemporary hardware it is even impossible.

Therefore, QoS approaches that rely on bandwidth reservation seem unpromising. Applicable alternatives are based on service differentiation which provides a prioritisation of service classes without giving hard guarantees.

We have introduced and evaluated the DLite algorithm, a novel approach to service differentiation in ad hoc networks. It applies a fair queueing scheme with separate queues for each service class. Late packets of delay constrained classes are dropped in intermediate routers.

DLite is easy to implement and requires low computational overhead. It allows for adaptive multimedia applications and permits gradual deployment.

The first simulation results are very promising, but still preliminary: Future work will comprise thorough studies with a variation of several parameters. The variation of the relative weight of each service class revealed hardly any impact on both the bulk TCP throughput and the fraction of realtime packet loss. This has to be studied in detail, in particular together with a variation of the amount of generated realtime and bulk traffic. Discarding realtime packets was based on a local observation of queueing time in a forwarding node. Additional resources may be saved by applying an accumulative delay measurement (e.g. using hop-by-hop options): Packets may be precautionary dropped when the sum of queueing delays exceeds a certain threshold. Finally, the impact of dynamics in ad hoc networks, namely the mobility models that are used for our studies, will be subject to further analysis.

References

- [1] Quality of service (QoS) concept and architecture (release 5) (TS23.107 v5.7.0). TS 23.107, 3GPP, December 2002.
- [2] S. Agarwal, A. Ahija, J. P. Singh, and R. Shorey. Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks. In *Proc. IEEE International Conference on Communications 2000 (ICC'00)*, volume 3, pages 1697–1701.
- [3] G.-S. Ahn, A. T. Campbell, A. Veras, and L.-H. Sun. Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN). *IEEE Transactions on Mobile Computing*, 1(3):192–207, July–September 2002.
- [4] S. Baatz, M. Frank, C. Kuehl, P. Martini, and C. Scholz. Bluetooth scatternets: An enhanced adaptive scheduling scheme. In *Proc. IEEE INFOCOM 2002*, pages 782–790. IEEE, June 2002.
- [5] F. Barceló and J. Jordán. Channel holding time distribution in cellular telephony. In *Proc. of the 9th International Conference on Wireless Communications (Wireless '97)*, pages 125–134, July 1997.
- [6] C. Bettstetter and C. Wagner. The spatial node distribution of the random waypoint mobility model. In *Proc. 1st German Workshop on Mobile Ad-Hoc Networks (WMAN'02)*, pages 41–58, Ulm, Germany, March 2002.
- [7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC 2475: An architecture for differentiated services, 1998.
- [8] Bluetooth SIG. *Specification of the Bluetooth System – Version 1.1 B*, February 2001.
- [9] R. Braden, D. Clark, and S. Shenker. RFC 1633: Integrated services in the internet architecture, 1994.

- [10] S. Chen and K. Nahrstedt. Distributed quality-of-service in ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 17(8), August 1999.
- [11] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM*, pages 1–12, 1989.
- [12] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. *IEEE Personal Communication*, February 1997.
- [13] K. Fall and K. Varadhan, editors. *The Ns Manual*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, April 2002.
- [14] M. Gerharz, C. de Waal, M. Frank, and P. Martini. Link stability in mobile wireless ad hoc networks. In *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN'02)*, pages 30–39, Tampa, FL, November 2002.
- [15] T. Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu. Preemptive routing in ad hoc networks. In *ACM Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM'01)*, pages 43–52, Rome, Italy, July 2001.
- [16] IEEE LAN/MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999. ANSI/IEEE Std. 802.11.
- [17] ITU. *Recommendation G.114 – One Way Transmission Time*, February 1996.
- [18] S.-B. Lee, G.-S. Ahn, and A. T. Campbell. Improving UDP and TCP performance in mobile ad hoc networks with insignia. *IEEE Communications Magazine*, pages 156–165, June 2001.
- [19] S.-J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *Proc. IEEE International Conference on Communications 2001 (ICC'01)*, pages 3201–3205, June 2001.
- [20] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 61–69, Rome, Italy, July 2001.
- [21] C. R. Lin and M. Gerla. Real-time support in multihop wireless networks. *ACM Wireless Networks*, 5(2):125–135, March 1999.
- [22] M. K. Marina and S. R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proc. of IEEE International Conference on Network Protocols*, pages 14–23, November 2001.
- [23] C. E. Perkins, editor. *Ad Hoc Networking*. Addison-Wesley, 2001.
- [24] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop wireless networks for quality-of-service support. *ACM Mobile Networks and Applications*, 3(1):101–119, June 1998.
- [25] T. S. Rappaport. *Wireless Communications – Principles & Practice*. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall PTR, 1996.
- [26] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *SIGCOMM*, pages 231–242, 1995.
- [27] R. Sivakumar, P. Sinha, and V. Bhargavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communication*, 17(8), August 1999.
- [28] C.-K. Toh. Associativity based routing for ad hoc mobile networks. *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, 4(2):103–139, March 1997.
- [29] H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua. A flexible quality of service model for mobile ad-hoc networks. In *Proc. IEEE Vehicular Technology Conference Fall 2000*, pages 445–449, May 2000.
- [30] H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua. On service prioritization in mobile ad-hoc networks. In *Proc. IEEE International Conference on Communications 2001 (ICC'01)*, pages 1900–1904, June 2001.
- [31] R. Yavatkar, D. Hoffman, Y. Bernet, F. Baker, and M. Speer. RFC 2814: SBM (subnet bandwidth manager): A protocol for RSVP-based admission control over IEEE 802-style networks, May 2000.
- [32] C. Zhu and M. S. Corson. Qos routing for mobile ad hoc networks. In *Proc. IEEE INFOCOM 2002*, pages 958–967, June 2002.