A Real-Time Traffic Packet Scheduler for a Novel TDMA MAC Protocol

Veljko Pejovic
University of California, Santa Barbara
veljko@cs.ucsb.edu

Heather Zheng
University of California, Santa Barbara
htzheng@cs.ucsb.edu

Abstract

Wireless mesh networks are being rapidly deployed in both urban and rural areas impacting a large fraction of the world's population. As the profile of the average user changes so does the observed traffic: voice and video communications are grabbing a bigger share since the networks are used as a key communication tool, especially in the areas without any other informational infrastructure. However, real-time applications require strict performance guarantees that cannot be provided by the current off the shelf solutions. Understanding the behavior of voice and video traffic in the wireless environment sets good foundations for designing an efficient solutions on all networking layers. In this paper we focus on analysis of large scale network traces, experimenting with voice and video flows over the wireless medium and, based on the derived results, designing a packet scheduler for a TDMA-like MAC layer protocol.

1 Introduction

Due to their ubiquitousness, low cost and the ease of installation, wireless networks based on 802.11 family of protocols are being rapidly deployed in scenarios radically different from a standard office setup. Nowadays, mesh networks are used to provide connectivity to remote areas, substitute expensive land-line telephone grids in the developing world and even cover small towns with wireless signal.

The traces from the community mesh networks show that voice and video conferencing represents a large portion of the overall network traffic [14]. Deploying networks in rural areas of the developing world puts even more pressure on implementing efficient voice and video solutions having in mind that a large part of the population is illiterate or uses the provided technology as a replacement for more expensive or unavailable telephone lines. Finally, applications such as remote health service rely on the real-time correspondence of doctors, nurses and patients over wireless links [19].

One of the major constraints of the current wireless mesh technology is that there are no bandwidth and delay guarantees for the real-time applications. The crucial problem is that every packet must contend for the medium leading to the unpredictable delivery delay and packet loss. To cope with this problem, various TDMA solutions are being designed for both structured and unstructured networks. These protocols allocate resources exclusively for a single user (node/flow) thus providing quality of service guarantees; at the same time the solution should maximize the network utilization by allowing as many concurrent flows as possible.

However, in order to satisfy the strict delay and jitter restrictions imposed by voice and video traffic, it is crucial to have a clear picture of the actual behavior of the real time codecs and protocols in different environments. There is a big gap between the theoretical application level behavior and the real world network measurements that points to the space for major protocol improvements.

Our goal is to conduct a thorough survey of traffic characteristics (delay, jitter, packet size and packet loss) in various network environments and use the results, along with the analytical view on the application specifics, as a base for designing the packet scheduler. Having in mind the varying degree of predictability for different types of flows it is not a straightforward task to schedule packets and guarantee the delivery quality for real time applications; however we feel that merging the analytical and the empirical approach can provide substantial improvement over the existing solutions. The scheduler is not envisioned as an isolated layer in the network stack. Rather, it is aware of the underlying MAC protocol [16] (currently under development) as well as of the different application needs.

With that in mind, the contributions of this paper are two-fold:

- Providing insights on the characteristics and anomalies observed in the real-time traffic scheduling over the standard 802.11 protocol
- Designing a packet scheduler that explicitly targets improvement of voice and video traffic delivery over wireless mesh networks

The rest of the paper is organized in the following fashion: chapter 2 examines previous work in the fields of traffic characterization, TDMA MAC protocols and real-time packet scheduling in wireless networks, chapter 3 describes the methodology used; findings from the traces and the experiments are presented in chapter 4, a novel MAC protocol is sketched in chapter 5, while the corresponding packet scheduler is described in chapter 6. Finally, a brief discussion of the paper contributions and the guidelines for the future work can be find in chapter 7.

2 Related Work

One of the first large scale traffic analysis results were provided in the seminal paper [9] by Henderson et al. Although they provide only a broad picture of the usage patterns ad different traffic types observed in a campus networks, the authors also made guidelines towards further analysis of the traces which were made freely available [1]. The traffic profiling similar to the one in our work was done by Sun et al. [18], but only for FTP and HTTP traffic. Okabe et al, examined identification of VoIP flows based on packet sizes and inter-arrival times, which is one of the goals of our analysis, yet they do not consider wireless networks nor video traffic.

Quality of service guarantees provisioning via MAC layer improvements is a very prolific field, and the proposed work ranges from defining better interference models [21] to utilization of multiple non-overlapping channels [6]. Protocols such as [22] and [17] are specifically targeting VoIP traffic. The former introduces a novel TDMA protocol that uses cross-layer information while the latter provides dynamic TDMA-like behavior by having nodes sense the existing transmissions. Since VoIP traffic shows well defined periodic behavior, a newly joined node can infer its existence just by sensing the medium and marking the slots used by VoIP flows. The new flows allocate the slots not used by VoIP flows therefore implicitly providing QoS guarantees for them. Unfortunately, not all real-time traffic is periodic with a uniform packet sizes: video flows consist of packets of different, unspecified sizes, and possibly different inter-packet departure times. Concluding about the existence of video flows by sensing the medium is very hard and the reservation of the future slots might not be feasible.

Scheduling real-time traffic over a wireless channel is examined in [15] and the authors show that the Feasible Earlies Due Date (FEDD) strategy gives better performance than the Head of the Line (HOL) policy. However, knowing the packet deadline is not always possible, and we see this method as a pointer to the right direction, but not a ready-to-implement solution.

3 Methodology

There is a limited number of freely available wireless network traces that contain real time traffic, among them the following were selected for the analysis:

- *Dartmouth*: a full tcpdump trace from the Dartmouth College campus [1]. The specific trace was collected during the fall semester 2003/2004. The trace is attractive because of the abundance of voice traffic, since the campus has a well developed network of Cisco and Vocera VoIP solutions.
- *Megaconference* (World's Largest Internet Videoconferences) TCPdump traces collected at the switch at The Ohio State University, although not necessarily wireless, provides good insight on the different protocols used for video conferencing.
- *IETF* meeting traces collected by the University of California Santa Barbara researchers during the 2004 meeting. The trace allowed us to identify anomalies that can lead to fake-positives when classifying traffic.

To get more insights about the behavior of standard 802.11 protocol stack, the set of experiments was conducted in a quiet, controlled lab environment. The hardware and software configuration used is the following: two IBM Thinkpad laptops equipped with Atheros 5212 chipset based wireless NICs, running 2.4 Linux kernels and Madwifi 0.9.4 [2]. Two applications were used for generating the desired traffic: *rude* (Real-time UDP Data Emitter) [3] that allows sending packets whose size and departure times are defined in a user supplied file, and *iperf* [4], a well know tool for generating traffic used for network performance measuring. *Rude* traffic generator was fed with the real life web camera traces generated by Technical University Berlin Video group [13], and VoIP like traffic that corresponds to G.711 protocol definition. *Iperf*, on the other hand, was used as a source of a light UDP background traffic.

The experimental scenarios included both video and voice real-time flows, a varying number of flows and varying intensity of the background traffic, more details on the experiments are given in section 4.

From a set of possible metrics that can be gathered we focused on those that are the most important for good real-time traffic performance, and those allowing easy traffic profiling: inter-packet arrival time, packet jitter, packet delivery delay, packet loss. We also focused on identifying distinctive traffic patterns from the examined traces.

4 Analysis Findings

The Dartmouth network is an access point based network and as such represents a conservative estimation of the problematic mesh networks that this paper is targeting. Therefore, the performance of this network is not worse than that of the target networks. The VoIP traffic observed in the network is based on either Cisco or Vocera solutions, yet the graphs shown in this sections refer specifically to the Cisco implementation. Since the trace does not allow us to look at the packets at their source and destination explicitly, we cannot calculate packet delay or packet loss. For that reason, only jitter and inter-packet arrival time distributions are shown in figures 1 and 2 for the longest flows observed in the trace. The results show satisfying performance: 99% of the inter-arrival times are bellow 20 ms (protocol defined), and the jitter is almost always less than 2 ms. However, mapping of the inter-arrival times in figure 3 shows an unusual anomaly, some packets' arrival seem to be alternating between more and less separated, and the difference gradually diminishes leading to the expected performance after a short stabilization time. The zoom-in figure 4 reveals that pairs of packets not conforming with the expected behavior are separated by groups of three packets that have 20ms intervals between them. One possible explanation of this phenomenon is that one of the devices along the path allocates specific processing times to different flows/applications. This leads towards direct processing of the observed flow at one time (when the groups of three packets are observed) and buffering its packets at another time (when the pairs of packets not lying on 20ms line are observed). Moreover, the processing time is adjusted dynamically since the pattern diminishes after some time.

Megaconference trace contains real-time flows from unidentified types of networks, most probably both wired and wireless. Therefore we concentrate on the analysis of possible modification

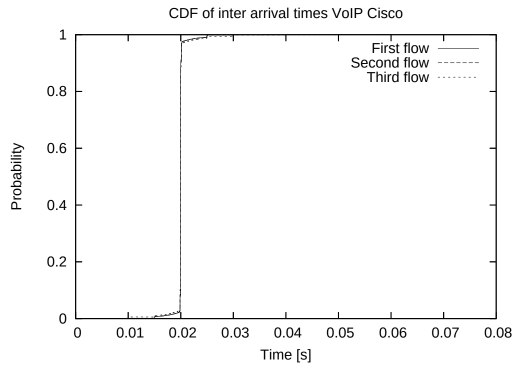

Figure 1: The Inter-arrival time distribution for voice traffic observed in the Dartmouth trace

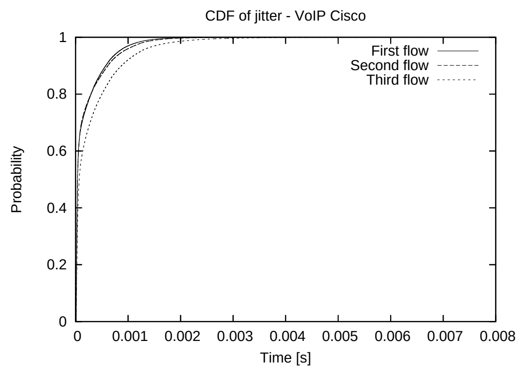

Figure 2: The packet jitter distribution for voice traffic observed in the Dartmouth trace

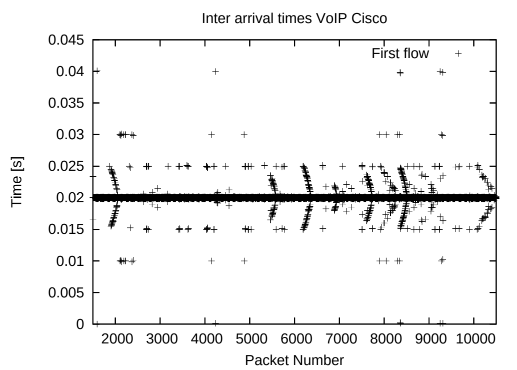

Figure 3: The Inter-arrival time for a representative flow observed in the Dartmouth trace

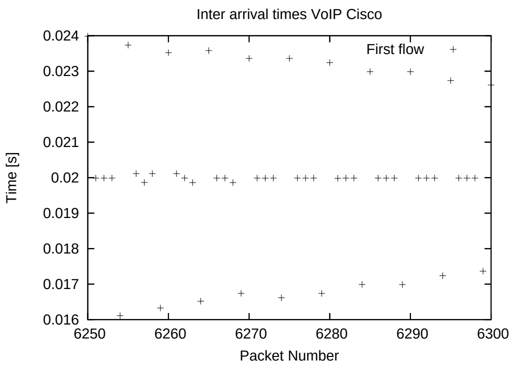

Figure 4: Detail: The Inter-arrival time for a representative flow observed in the Dartmouth trace

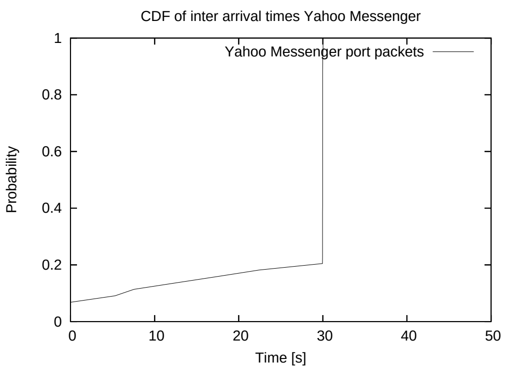

Figure 5: The Inter-arrival time distribution for Yahoo Messenger traffic observed in the IETF trace

to video and voice traffic from the global perspective. We find a plethora of known protocols for both voice(G.711) and video (H263) traffic. We also noted an interesting phenomenon: periodic (T=60ms) fixed-size UDP flows are observed. The most probable explanation would be the existence of a "smart" intermediate router that combines two packets and sends them with the halved periodicity.

As stated before, IETF traces do not contain enough real-time flows to be analyzed, yet they do have a variety of applications that may produce real-time traffic. For example, the Yahoo Messenger application among the other things provides voice/video conferencing capabilities. Besides real-time flows, information such as chat messages and keep-alive messages are sent via the same port. Figure 5 shows that a majority of traffic on a port dedicated to this application was periodic, however a closer examination shows that the observed traffic is not voice/video but periodic keep-alive messages. This has a significant impact on the way we classify traffic in section 6.

The experiments ran on the testbed aimed profiling and explaining traffic anomalies when using 802.11 protocol. The following runs between the two laptops were conducted, five times each:

- a single 5 min VoIP flow with no background traffic
- a single 5 min VoIP and a single video flow with no background traffic
- multiple various length (between 1 min and 5 min) VoIP flows with no background traffic
- multiple various length (between 1 min and 5 min) VoIP and video flows with no background traffic
- a single 5 min VoIP flow with background traffic
- a single 5 min VoIP and a single video flow with background traffic
- multiple various length (between 1 min and 5 min) VoIP flows with background traffic
- multiple various length (between 1 min and 5 min) VoIP and video flows with background traffic

Due to space constraints we show graphs only for the first case, but note that all the results are in the borders of expected for the first four test cases; it should be also noted that packet loss was virtually non-existent throughout the experiments. In figure 6 we see that most of the packets retain 20 ms inter-arrival time defined by the VoIP protocol used. The jitter is almost always under 3.5 ms and the relative delay does not show great variations. It should be stated that every point in the delay figure 7 represents the difference between the minimum observed reported delay and the current packet's reported delay. Therefore we refer to that value as the *relative* delay. The absolute delay cannot be measured since the two laptops' clocks are not synchronized, moreover the slope on the graph shows that the two clocks tend to drift.

For the remaining four cases we can again extract the common behavior: inter-arrival times can be classified in three groups very short (less than 1ms), expected (around 20ms) and very long (more than 100ms). This unusual behavior in presence of the background traffic can be explained once we examine per packet behavior. Figure 8 shows a single voice flow packets' inter-arrival times detail, clearly intervals when the packets are received with 20 ms periodicity are alternating with the intervals of multiple packets being buffered and sent at once resulting in the enormous inter-arrival time for the first one in the group and immediate delivery for those that follow it. This can be explained with either the operating system or MadWiFi driver processing

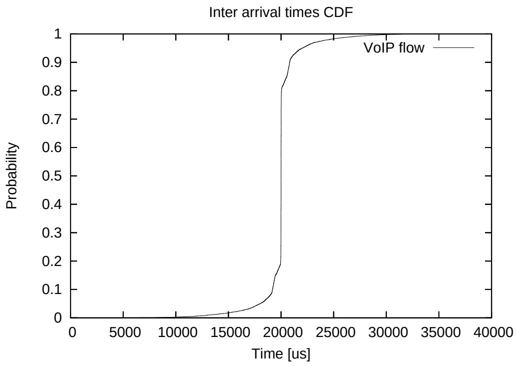

Figure 6: The Inter-arrival time distribution for voice traffic in the experimental testbed (no background traffic)

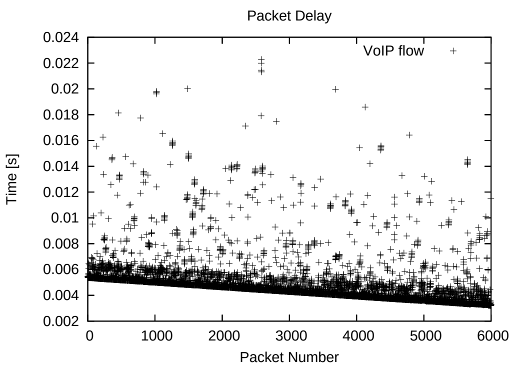

Figure 7: Relative packet delay for voice traffic in the experimental testbed (no background traffic)

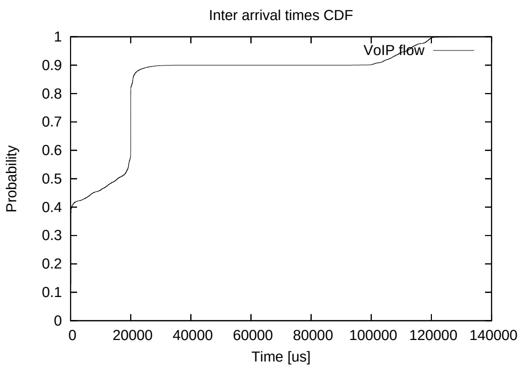

Figure 8: The inter-arrival time distribution for voice traffic in the experimental testbed (with background traffic)

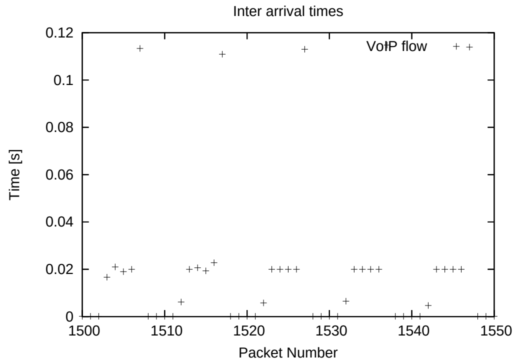

Figure 9: Detail: The inter-arrival time for voice traffic in the experimental testbed (with background traffic)

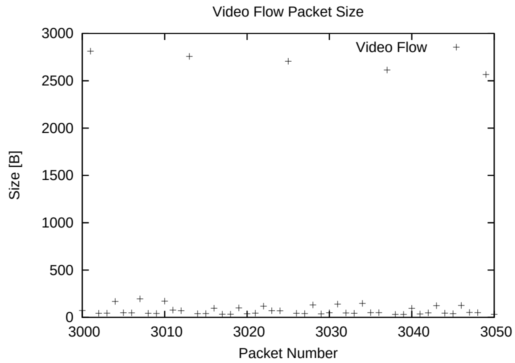

Figure 10: Detail: The packet size for video traffic in the experimental testbed

two different applications: iperf and rude.

However, the relative delay reveals no anomalies, and since the delay is calculated based on the timestamps acquired right after a packet is sent to the driver it can be concluded that the operating system keeps separate processing queues for the two applications leading to disrupted real-time flows. This finding is extremely valuable as it shows that quality of service guarantees for voice and video traffic cannot be made if only a single (MAC) layer is considered. Finally we examined the distribution of packet sizes for web camera video flow used in the experiments. The packet sizes are either very small, corresponding to P and B-frames, or very big, corresponding to I-frames. A closer look at the flow (figure 10) reveals a distinct fingerprint formed by one I-frame and eleven P/B-frames.

5 A Novel MAC Protocol

Strict delay, jitter and packet loss requirements of voice and video flows can be satisfied if network resources are explicitly granted to them, so that the packets belonging to these flows do not have to contend for the medium. Time division solutions have been suggested for providing QoS guarantees but face challenges in forms of time synchronization, dynamic adapting to varying traffic, and good network utilization. On the other hand 802.11 amendment to the protocol introduces TxOP (transmission opportunity): in a nutshell, when a node wins a medium it gets the opportunity to transmit more than a single packet as long as its dedicated TxOP lasts. By having different TxOP period lengths for different types of traffic, real-time flows can be prioritized to the delay tolerant flows. However, 802.11e does not give any guarantees for prioritized traffic, only a probabilistic assurances that the real-time traffic will get prioritized treatment. The protocol that is being proposed by Sharma [16] extends the above mentioned approach to another level where a node gets a TxOP opportunity not only immediately after it wins the medium, but also periodically after that. Periodic real-time flows such as voice and video would benefit from such a scheme if the length of the TxOP period, its periodicity and the number of cycles correspond to the flows' needs. Without going further into complex details of the protocols and obstacles for its implementation in the rest of the paper we assume that the MAC layer behaves according to the description presented in this paragraph.

6 Designing the Scheduler

The following constraints should be met in order to provide good real-time traffic performance:

- End to end delay should be less than 150ms
- End to end jitter should be less than 20ms
- Packet loss should be less than 0.5%

Note that in case of video flows there is a big difference between loosing an I-frame, carrying the complete image, or a B/P-frame that only carries the difference from the previous image.

6.1 Traffic Classification

An efficient scheduler needs to recognize delay sensitive traffic and that can be done in two ways:

- Breaking the layering and getting the information directly from the packet bits, similarly to the way the packets are decoded by Ethereal
- Keeping the track of the packets and by comparing the observed flow profile with one the profile of voice/video traffic categorize a flow as a real-time or not

The first approach is based on registering the flow transport layer information such as a protocol type (TCP/UDP) and a port number. Some ports are reserved for real-time traffic: Apple Quicktime uses port 458, RTP flows usually use ports 5004 and 5005 and so on. Unfortunately, our findings from the traces show that Yahoo Messenger uses the same port for sending real time packets and other, delay tolerant, data such as keep-alive messages. Moreover, some applications like Skype use random ports for voice and video traffic making it virtually impossible to distinct voice-video flows by examining only transport layer information.

The second method is based on periodicity, packet sizes and traffic patterns observed in real-time flows. As stated before VoIP traffic has a simple distinctive profile: every 20-30ms (depending on the codec used) a fixed size packet is being sent. Again, the traces we examined also showed that sometimes the packets can be aggregated and those cases should also be registered as real-time traffic. Video traffic on the other hand is more complex, but consists of periodic groups of frames (Groups of Pictures - GoPs) that can be used as its fingerprint. The distribution of the frame sizes very distinctive for video flows: 90% of the frames are very small and the other 10% are very large. This distribution holds for the examined web-cam trace, and is slightly different for traces that are made by a moving camera or the traces of highly dynamic scenes. We advocate using the second approach with these guidelines:

- The scheduler is fed with the know periodicity and size distributions, and distinctive patterns for the well known real time protocols
- The flows are monitored until the confidence level of a scheduler is high enough to categorize the flow after which the scheduling is performed

The major challenge is setting the right period for training the scheduler: it should be as short as possible so that the guarantees can be quickly provided but it should also gain enough confidence before categorizing a flow.

6.2 Packet Selection

The next issue we consider is choosing the packets to schedule or drop. Here we assume that the flows have already been classified and discuss the scheduling of the real-time traffic only. Obviously the packets whose deadline has expired should be dropped and those whose deadline is near should be delivered to the destination as soon as possible. If a node can decode a packet's

RTP information the current delay and deadline information can be approximated in the following manner:

- For every packet the sending time (t_{Tx}, extracted from the RTP payload) and the reception time (t_{Rx}) is recorded
- The difference between the smallest observed delay ($min(t_{Rx} - t_{Tx})$) so far in the flow and the current packet's calculated delay ($t_{Rx} - t_{Tx}$) is taken as a conservative approximation of the actual packet's delay. The logic behind this approach lies in the fact that the nodes are not synchronized, therefore $t_{Rx} - t_{Tx}$ can be arbitrarily offset from the actual packet delay. If we assume that at least one packet has been received with no delay the aforementioned difference gives us good approximation of the actual delay.
- The packet whose delay is already higher than the acceptable delay is dropped, while all the other packets are ordered according to their descending experienced delay, and the first one in the line is scheduled first

Not all real-time traffic uses RTP protocol that allows easy decoding. If the delay information cannot be extracted we can still perform the optimization in case of a video flow:

- the frames recognized as I-frames are scheduled first since they carry more information than B and P frames, additionally, all the other frames are useless if the corresponding I-frame is not delivered

Finally, in case of multiple flows per node all flows are treated fairly, having their packets scheduled according to the frequency of the arrivals and the delay experienced so far.

6.3 MAC Layer Protocol Adjustment

The parameters t, T and N of the MAC protocol described in section 5 need to be adjusted to match the properties of the flow that is being scheduled. We propose the following values:

- t is set to the value that accommodates: either *2k* voice packets where k is an integer, in case of voice flows or a single GoP if video traffic is observed. The even number of voice packets is scheduled since in Section 4 we show how VoIP packets can be of the double size, possibly aggregated, while a GoP is scheduled since it represents a single meaningful unit of a video flow.
- T set according to the expected number of hops. Maximum waiting time at an intermediate node is *T-t*, if T is set to a large value the cumulative delay guarantees cannot be provided while, a small value leads to underutilization of the medium.
- N corresponds to the expected session length. The most simple heuristic we suggest is for N to take value that makes *T*N* equal to the length of the training period for the flow categorizer.

At this point the MAC protocol does not specify if the slot reservations are made on a per node or a per flow basis. It the former case the flows should be aggregated and the values for t, T and N should be set accordingly, while in the latter a node gets more flexible QoS provisioning opportunities.

7 Conclusions and Future Work

A growing interest in enabling real-time applications over wireless mesh networks opens a new venue for research on all layers but especially for work on packet scheduling and medium access. In this paper we presented a design of a packet scheduler that in cooperation with a novel MAC protocol should provide a basis for the efficient delivery of voice and video traffic over a wireless medium. The scheduler design is based on a thorough analysis of large scale data traces and a small scale experimental testbed results. We point out specific characteristics and anomalies of real-time traffic and show how they can be used for scheduling packets. Specifically the solutions for correct traffic classification, packet selection and MAC parameter adjustments are elaborated. We also feel that the analysis of the traces can serve as a starting point for the future packet scheduling research. A number of questions are still open: *1) more flexible traffic classification*, although addressed in section 6, it requires an in-depth study that would identify all real-time protocols and possibly come up with a generalized set of descriptions that can be used for traffic classification. *2) flow length determination*, parameter N from the MAC protocol description needs to take the value that corresponds to the current flow's length, however this cannot be known upfront, a probabilistic approach would set N to the value that would guarantee that a large number of flows would get the QoS guarantees. *3) Maximum number of admitted flows* , this problem needs a dynamic solution that would adapt to the network conditions and traffic requirements.

In the end we need a fully functioning implementation of the scheduler and the MAC protocol, and a thorough analysis of the behavior in the real-life situations.

8 References

[1] http://crawdad.cs.dartmouth.edu/.

[2] http://madwifi.org.

[3] http://rude.sourceforge.net/.

[4] http://sourceforge.net/projects/iperf.

[5] http://www.megaconference.org.

[6] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, Philadelphia, PA, USA, September 2004.

[7] P. Calyam and C.-G. Lee. Characterizing Voice and Video Traffic Behavior over the Internet. In *International Symposium on Computer and Information Sciences (ISCIS)*, Istanbul, Turkey, October 2005.

[8] K. K. R. I. Dragos Niculescu, Samrat Ganguly. Performance of voip in a 802.11 wireless mesh network. In *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.

[9] T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-Wide Wireless Network. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, Philadelphia, PA, USA, September/October 2004.

[10] P. Ikkurthy and M. A. Labrador. Characterization of mpeg-4 traffic over ieee 802.11b wireless lans. In *LCN '02: Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, Washington, DC, USA, November 2002.

[11] T. Okabe, T. Kitamura, and T. Shizuno. Statistical Traffic Identification Method Based on Flow-Level Behavior for Fair VoIP Service. In *1st IEEE Workshop on VoIP Management and Security*, April 2006.

[12] B. Sat and B. Wah. Analysis and Evaluation of the Skype and Google-Talk Voip Systems. In *IEEE International Conference on Multimedia and Expo*, Toronto, Ontario, Canada, July 2006.

[13] P. Seeling, F. Fitzek, and M. Reisslein. *Video Traces For Network Performance Evaluation – A Comprehensive Overview and Guide on Video Traces and Their Utilization in Networking Research*. Signals & Communication. Springer, December 2006.

[14] S. Sen, S. Kole, and B. Raman. Rural Telephony: A Socio-Economic Case Study. In *International Conference on Information and Communication Technologies and Development (ICTD-2006)*, Berkeley, CA, USA, May 2006.

[15] S. Shakkottai and R. Srikant. Scheduling Real-Time Traffic with Deadlines vver a Wireless Channel. *Wireless Networks*, 8(1):13–26, August 2002.

[16] A. Sharma. A novel mac protocol. Private conversation, April 2008.

[17] S. Singh, P. Aravinda, K. Acharya, U. Madhow, and E. M. Belding-Royer. Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks. *Ad Hoc Networks*, 5(6):744–768, August 2007.

[18] Z. Sun, D. He, L. Liang, and H. Cruickshank. Internet QoS and Traffic Modelling. *IEE Proceedings - Software*, 151:248– 255, October 2004.

[19] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer. Beyond Pilots: Keeping Rural Wireless Networks Alive. In *ACM USENIX NSDI*, San Francisco, CA, USA, April 2008.

[20] T. Suzuki and S. Tasaka. Performance Evaluation of Integrated Video and Data Transmission with the IEEE 802.11 Standard MAC Protocol. In *Global Telecommunications Conference GLOBECOM '99*, Rio de Janeireo, Brazil, December 1999.

[21] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder. Efficient interference-aware tdma link scheduling for static wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, Los Angeles, CA, USA, September 2006.

[22] X. Wang, A. Patil, and W. Wang. Voip over wireless mesh networks: challenges and approaches. In *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*, Boston, Massachusetts, August 2006.