A Case for Application Aware Channel Access in Wireless Networks

Ashish Sharma, Elizabeth M. Belding
Department of Computer Science
University of California, Santa Barbara CA 93106
{asharma, ebelding}@cs.ucsb.edu

ABSTRACT

The increasing reliance of users on wireless networks for Internet connectivity has posed two significant challenges for mobile networking research. The first challenge is to provide high quality of service for interactive real-time applications such as VoIP and video conferencing. The second challenge is to reduce the energy consumption of mobile devices and improve battery life. Past research has focused on separately addressing these seemingly conflicting goals in distributed medium access based wireless networks. Contrary to the traditional tiered networking approach, we argue that an application aware approach to medium access and power saving has the potential to significantly improve the performance of real-time applications and conserve battery power on mobile devices. As a proof of concept, we present the design and implementation of *Rendezvous* - an application aware MAC protocol. *Rendezvous* uses short term dynamic channel reservations to achieve higher quality of service and increased power saving opportunities for mobile devices running VoIP and real-time video applications. A preliminary evaluation from our testbed implementation reveals promising results, motivating the need and opportunity for future research in the direction of application awareness at lower layers of the networking stack.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols, Wireless communication; C.2.5 [**Local and Wide-Area Networks**]: Access schemes

General Terms

Design, Performance

Keywords

Wireless Networks, Application Awareness, Medium Access

1. INTRODUCTION

Mobile devices today offer support for a rich set of interactive applications such as VoIP, live video conferencing and online gaming, among many others. Several new applications, such as "AOL Buddies near me", Neighborcast [1], OLPC mesh networks, music sharing and gaming over WiFi, utilize peer-to-peer connections among nearby nodes. End users expect the same level of service from their wireless networks as they do from wired networks. However, shorter battery life of mobile devices and the inability to satisfy the strict performance guarantees required by interactive multimedia applications, such as low delay and jitter, often lead to a poor end user experience.

Present day smart phones and PDAs are equipped with multiple network technologies such as WiFi and Bluetooth in addition to the cellular interface (GPRS or CDMA). Past research has shown that it is better to use WiFi instead of the cellular data connection for real-time applications [2]. This is because WiFi offers higher throughput, lower end to end delay, and reduced per-bit power consumption. However, the energy consumption of a mobile device increases dramatically when the WiFi interface is turned on. Contention-based medium access requires the WiFi interface to stay powered on much longer than, for instance, TDMA based scheduled access. Furthermore, the presence of hidden terminals results in packet collisions and retransmissions that again consume more power.

The IEEE 802.11 Power Save Mode (PSM) allows wireless clients to enter a low power sleep mode and wake up periodically (usually 100ms) to listen for beacons and retrieve any buffered packets. However, the increased latency of PSM affects the performance of real-time applications. Thus, most mobile devices do not use this feature and constantly keep their WiFi interface in awake mode. Further, the PSM mode cannot be used in a multi-hop mesh setting due to the absence of a central controller. TDMA, on the other hand, can provide high quality of service and reduced power consumption, but it is difficult to implement due to the requirement of global synchronization and centralized scheduling. These features are a challenge to achieve not only in a peer-to-peer multi-hop network but also in a WLAN setting as nearby clients may be connected to different access point controllers. Hence, it is difficult to simultaneously improve quality of service and conserve energy in distributed channel access based wireless networks.

In this paper, we argue that information about the application characteristics can be used in promising ways to influence and modify the behavior of lower layers of the networking protocol stack. Our argument extends beyond just giving priority to real-time traffic, such as in the IEEE 802.11e protocol. We claim that there are interesting new possibilities for future research by enabling meta-information flow between layers of the networking protocol stack. To substantiate our claim, we show that information about the characteristics of the application traffic can be used at the MAC layer to reduce contention and achieve quasi-TDM channel access behav-

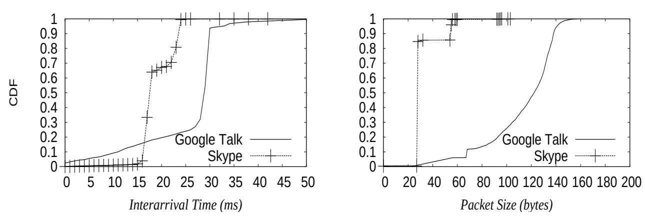
HotMobile 2009, February 23-24, 2009, Santa Cruz, CA, USA

ior. This helps improve the quality of service for real-time applications and conserve energy of mobile devices by putting them in low power sleep mode more often. As a proof of concept, we design and implement *Rendezvous* – an application aware adaptive MAC protocol that can be implemented on current 802.11 hardware, requiring only changes to the software. Based on the knowledge of real-time application traffic characteristics such as periodicity and duration of data bursts, the proposed MAC protocol uses dynamic periodic scheduling by means of explicit reservations, without the need for a central controller or global time synchronization. The intuition is that, by maintaining soft state about the predicted channel usage in the near future, the MAC protocol can make appropriate reservations for medium access in advance. Such scheduled channel access reduces contention and allows devices to sleep during periods of inactivity and save power.

We present some promising results from a preliminary evaluation of the application aware MAC protocol using OPNET simulations and an implementation on a testbed comprised of commodity 802.11 cards and Linux laptops. Our approach of using explicit reservation messages to schedule multiple periodic transmissions has two distinct advantages. First, unlike per-packet RTS/CTS or individual TXOP transmissions in 802.11e, *Rendezvous* reduces the control overhead by reserving multiple periodic TXOPs, based on real-time traffic characteristics. Second, by maintaining state information, *Rendezvous* promotes scheduled transmissions and reduces the unpredictability of random access, thereby exhibiting quasi-TDM behavior and reduced power consumption.

2. APPLICATION AWARENESS

Application awareness has been used in the past to put the wireless interface to sleep during periods of inactivity [3]. We now seek to understand the characteristics of common application traffic with the intention of using this knowledge not only to improve battery life of mobile devices but also reduce packet collisions due to contention in a wireless network. The performance of VoIP and streaming video depends on low end-to-end delay and jitter for smooth operation. We measure the inter-arrival times and packet size distributions of some popular VoIP and video applications.

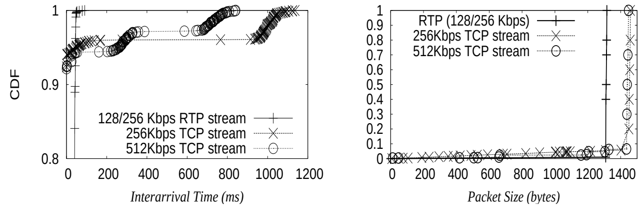


(a) (b)

Figure 1: Characteristics of VoIP applications.

Voice traffic: Voice traffic is comprised of periodic transmissions of small packets. For example, the G.711 voice codec sends 160 byte packets every 20ms. We study the characteristics of two popular VoIP applications - $GoogleTalk$ and $Skype^{TM}$. As shown in Figure 1, Skype traffic has a high degree of periodicity with a time period of 20ms, while GTalk sends a majority of its packets at an interval of 30ms. The two dominant packet sizes in Skype traffic are 28 and 55 bytes. GTalk has a variable packet size distribution; however, all packets are smaller than 160 bytes.

Video streaming traffic: Video is streamed over the Internet using either the RTP/UDP or TCP transport layer protocols. Different codecs create video (I/B/P) frames of variable size. For transport over packet-switched networks, the video frames can be packetized in one of two ways - constant bit rate (CBR) with variable quality or variable bit rate (VBR) with constant quality [4]. Due to the high overhead of packet headers in the VBR approach, CBR is dominant in current networks wherein a network packet may contain multiple small video frames.



(a) (b)

Figure 2: Characteristics of video applications.

We study the characteristics of both RTP and TCP based video streaming services. For RTP streaming, we use the VLC media player to transmit an *mp4v* encoded video file at 128/256 Kbps using MPEG TS encapsulation. The audio codec used is *mpga*. We observe fixed sized packets (1324 bytes) at an inter-arrival time of 20ms-40ms for individual packets in the 128 Kbps and 256 Kbps RTP streams, as shown in Figure 2(a).

For TCP-based video transmission, we analyze the characteristics of two live video streams from a news channel website (CNN-IBN) [5] that uses the Microsoft Media Server (MMS) protocol for video streaming. Data from the media server arrived in periodic bursts with fixed size TCP segments (1448 bytes), as shown in Figure 2(b). The TCP PUSH flag of the last packet in each burst was set to "push" all the buffered data to the receiving video player application. Each burst contained a fixed number of packets to maintain a constant bit rate. We conducted two experiments to study MMS video streams at 256Kbps and 512Kbps data rates. In both experiments, the overall periodicity of bursts was observed to be 1s. While packets arrived in a single burst each second in the 256Kbps TCP stream, the 512Kbps stream contained two bursts separated by approximately 300ms, as shown in Figure 2(a). Li *et al.* observed similar trends in the packet size distribution, inter-arrival times and packet bursts in Windows Media Player^{TM} streams [6].

2.1 Case for dynamic periodic scheduling

Figures 1 and 2 show that both voice and video traffic have predictable inter-arrival times and packet sizes. While VoIP and RTP video packets arrive at fixed intervals (20ms-40ms), TCP-based video streams are characterized by periodic bursts of fixed size packets. This suggests that the use of dynamic periodic scheduling based on real-time application characteristics can reduce contention and allow mobile devices to save power by entering into sleep mode during periods of inactivity. In the past, periodic channel reservations for voice and video traffic have also been proposed for networks with a central controller, such as cellular networks [7, 8].

Since packets in a wireless network may suffer transmission delays due to random backoff and retransmissions, the 802.11 protocol does not provide any guarantee on the time of packet transmission. Thus, most non-interactive applications are tolerant of short term buffering and transmission delays up to several milliseconds. Many control packets and routing updates are periodic, such as beacons and gateway advertisements. Other network traffic, such as DNS queries, ARP requests, and UDP packets, are insensitive to small delays. Hence, in addition to real-time traffic, traffic from other applications that can tolerate short term delays can also be buffered temporarily to exhibit an artificial periodic traffic profile, without significantly affecting their performance or functionality.

3. RENDEZVOUS: AN APPLICATION AWARE MAC

In a distributed contention based medium access protocol such as IEEE 802.11, each packet has to undergo a random backoff before transmission to avoid packet collisions. This adds a variable amount of delay before a packet is transmitted. Furthermore, the presence of even a single hidden node can cause transmission failures due to collision. Each retransmission attempt is preceded by a random backoff interval determined by an exponentially increasing contention window. Thus, the performance of voice and video is contingent upon traffic from other nodes in the vicinity and may vary significantly during a session. IEEE 802.11e strives to provide quality of service through assignment of higher priority to voice and video traffic by means of reduced contention window values. However, lower contention parameters alone are insufficient to ensure low delay and jitter values. When multiple nodes have traffic of the same priority, the probability of two nodes transmitting at the same time increases, resulting in packet collisions.

To reduce contention delays and collisions, we argue that a MAC protocol can reserve the medium for future transmissions based on the application traffic profile. As a case study, we describe the design and implementation of one such MAC protocol that is well suited to VoIP and video applications in a wireless network. We use the real-time application characterization presented in Section 2 to build *Rendezvous* based on a simple yet powerful optimization of periodic channel reservations to reduce contention. Because of the inherent periodicity of multimedia voice and video streaming traffic, as shown in Section 2, *Rendezvous* reserves the channel for multiple future *Rendezvous-TXOPs*[1] ($RTXOP$) using a three-way $REQ/REP/RES$ exchange. Transmission of multiple frames within an $RTXOP$ and scheduling of multiple periodic $RTXOP$s based on application characteristics allows *Rendezvous* to avoid the large overhead of using an RTS/CTS exchange on a per-packet basis. Explicit reservations permit scheduled medium access, resulting in reduced contention delay due to random backoff. The reduced delay and jitter values due to the timely transmission of application traffic result in improved quality of service for voice and video applications. In addition to these benefits, a node can go into sleep mode during the $RTXOP$ intervals of neighboring nodes to save power. Note that unlike TDMA, *Rendezvous* does not require global synchronization or a central coordinator and requires only software changes to existing clients. We have implemented a prototype version of *Rendezvous* on a Linux testbed using commodity 802.11 cards as described in Section 4.

Rendezvous has three modes of medium access - *scheduled, random and opportunistic*. During its reserved periodic $RTXOP$, a node operates in *scheduled* mode and transmits packets with a higher channel access priority without following random backoff. During an unreserved period, nodes operate in the *random* access mode and follow the usual 802.11 backoff procedure. If a node has no more data to send during its reserved $RTXOP$, other nodes in the network enter the *opportunistic* access mode. In this mode, nodes wait for a minimum duration before assuming that the owner node of the $RTXOP$ has finished its transmission. They can then access the medium according to the 802.11 random backoff mechanism. The end result is that *Rendezvous* reduces contention in the medium and exhibits dynamic TDMA behavior. We now describe the operation of each access mode in detail.

[1]As defined in the IEEE 802.11e standard, a Transmit Opportunity (TXOP) is an interval during which a node can send multiple frames separated by SIFS interval without performing random backoff.

3.1 Scheduled access

In scheduled access mode, packets are transmitted with a higher channel access priority (without random backoff) during a reserved $RTXOP$. Sequential packets are transmitted with small interframe spacings (SIFS) within the reserved transmission time. Ongoing transmissions from conventional 802.11 clients can still delay the start of a scheduled $RTXOP$, penalizing the scheduled owner of the $RTXOP$. However, a node must ensure that its transmissions do not exceed beyond the $RTXOP$ duration. If the remaining time is insufficient to transmit the next frame, the node can either fragment the frame, refrain from sending the frame until the next $RTXOP$, or send the frame as a best-effort transmission following random backoff during an unreserved slot. Normal data traffic is transmitted using a best effort approach while high priority voice and video data is transmitted during $RTXOP$s.

Each node in the network maintains information about the current reservations in its vicinity in the form of a *reservation map*. To schedule a set of transmissions, a reservation request (REQ) is sent by the transmitter to the receiver containing the duration *d* of the $RTXOP$ and a list of start times that do not interfere with the existing reservations in the vicinity of the sender. The receiver selects a non-conflicting start time for the new $RTXOP$ from the proposed list based on the information contained in its *reservation map* and responds with a reply message (REP). Finally, the transmitter sends a reservation message (RES) to confirm the successful reservation of the channel. A node is considered either a *sending owner* (identified by an REQ reservation message) or a *receiving owner* (identified by an REP reservation message) of a $RTXOP$ depending on whether the node acts as a sender or as a receiver during the reserved period. This distinction is necessary for the proper functioning of *opportunistic* access (refer to Section 3.2). During an $RTXOP$ the sender can send multiple frames separated by an SIFS interval and receive a bulk acknowledgement at the end of the $RTXOP$, similar to the EDCF mode in 802.11e.

To ensure fairness of channel access among nodes, periodic reservations have a short life-span and have to be re-established after brief time periods (typically ranging from hundreds of milliseconds to a few seconds). Short-term reservations also provide the ability to dynamically adjust to variable clock-drift rates, and changes in topology and channel conditions. Scheduled access over brief duration for real-time traffic, as opposed to per-packet medium contention, induces dynamic TDM medium access behavior. A three way handshake also minimizes interference from hidden terminals during a $RTXOP$.

3.2 Opportunistic access

Applications that exhibit periodic transmission of packets can have periods of irregular traffic that deviate from the overall periodicity of the traffic flow. The differences in packet size and periodicity (due to silence suppression) of application traffic make it likely that a simple approach of periodic reservations of fixed durations would fail to respond to the dynamism of network conditions and result in poor performance. Fixed size proactive medium reservations have the drawback of wasting the network bandwidth in the scenario where a node has insufficient or no data to send during its reserved TXOP. To prevent such bandwidth wastage, *Rendezvous* provides a mechanism to allow opportunistic medium access by other nodes.

If the *sending owner* of an $RTXOP$ finishes transmitting before the end of the reserved period, neighboring nodes can contend for the medium using random backoff. To maintain ownership, the sender does not backoff after a successful transmission during a periodic $RTXOP$. Instead, it transmits all the packets with inter-

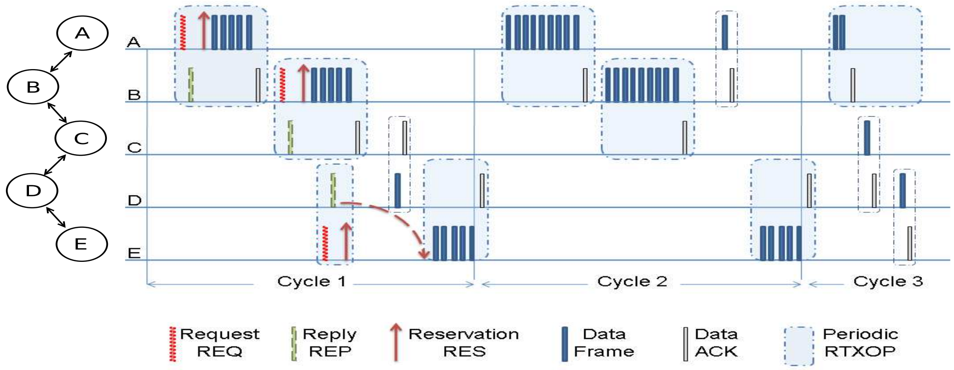


Figure 3: Rendezvous MAC in a multi-hop network: An example scenario showing establishment of periodic reservations in Cycle 1, multiple bulk acknowledged frame transmissions separated by SIFS intervals during a scheduled RTXOP in Cycle 2, and opportunistic access during a partially occupied RTXOP in Cycle 3. Cycles 1 and 2 also contain transmission of individual frames using random access during unreserved periods.

frame durations of SIFS. Other nodes in the network wait for at least a duration of EIFS (which is longer than SIFS) before assuming that the *sending owner* of the reservation period has no more traffic to send and the medium is now open for random access if no transmission is overheard.

The case of opportunistic access during a scheduled $RTXOP$, established at the neighboring nodes of the *receiving owner*, needs to be treated differently. The neighboring nodes may be hidden from the sender. Hence they must continue to refrain from transmitting to avoid hidden terminal interference at the *receiving owner* even though they sense the medium to be idle.

3.3 Random access

During an unreserved period, all nodes access the medium using conventional 802.11 protocol. Each node contends for medium access and follows the random backoff procedure before transmitting a packet. This mode of channel access can be used for transmission of best-effort traffic, as well as any high-priority packets that could not be transmitted during the scheduled $RTXOP$. This mode also ensures backward compatibility of *Rendezvous* with conventional 802.11 clients.

3.4 Example scenario

Figure 3 shows a sample network topology and the operation of the *Rendezvous* MAC protocol. In the five node topology, node A sends data to node C over a two hop path via node B. In the first time-cycle, node A sends a periodic reservation request (REQ) to B and establishes a periodic scheduled transmission opportunity ($RTXOP$). Node B establishes a non-conflicting $RTXOP$ to forward A's traffic to C. Node E, which is outside the transmission range of A, B and C, sends a reservation request to D. To prevent interference at node C, D responds with a RES message to schedule the $RTXOP$ at a non conflicting time in the future. Note that based on its application requirements, node E reserves a smaller but fixed duration $RTXOP$ as compared to the $RTXOP$s of nodes A and B.

In the second time-cycle, nodes A, B, and E send data packets in their scheduled $RTXOP$s, based on their medium reservation from time-cycle 1. Node A also sends an additional data packet to B, outside its $RTXOP$. This is an example of a node using random access for additional data during an unreserved period. The third time-cycle shows an example of opportunistic access where nodes D and E utilize a partially occupied $RTXOP$ reserved by node A. Before assuming that node A is done transmitting, D and E wait for an EIFS period. They then follow the random backoff procedure to capture the medium for transmission.

4. TESTBED IMPLEMENTATION

Rendezvous does not require any changes in the hardware and can be easily implemented on commodity 802.11 clients using a software upgrade. We have implemented the three access modes of *Rendezvous* on standard Linux machines equipped with Atheros (AR5212) chipset-based commodity 802.11 cards using the FreeMAC protocol development framework [9], Click modular router [10] and MadWiFi driver [11]. The traffic shaping module written in

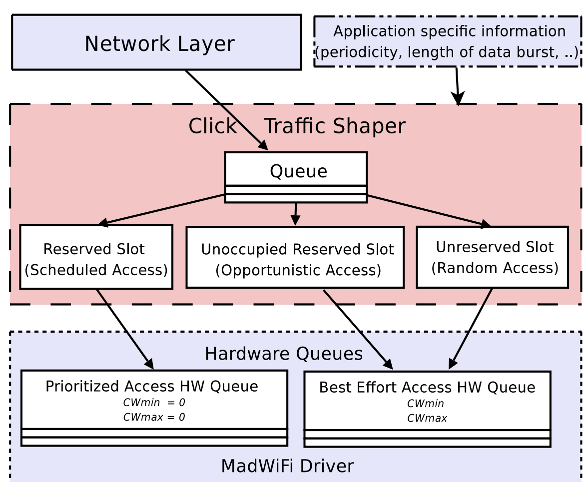


Figure 4: Rendezvous MAC architecture.

Click enqueues packets from the network layer and transmits higher priority voice and video packets during scheduled periodic $RTXOP$s. Data packets that could not be transmitted during an $RTXOP$ can either be buffered until the next $RTXOP$ arrives or sent along with best-effort traffic using opportunistic or random access. The traffic shaper is also responsible for aggregating the demands of various applications and initiating the $RTXOP$ reservation mechanism using REQ/REP messages based on individual application characteristics. As shown in Figure 4, the traffic shaping module directs the incoming packets from the application layer into different

	Medium Access Delay (sec)			Backoff Slots (32 μs each)			ReTx Attempts per second		
	Node A *initial*	Node C *hidden*	Node A *hidden*	Node A *initial*	Node C *hidden*	Node A *hidden*	Node A *initial*	Node C *hidden*	Node A *hidden*
$802.11_{RTS/CTS}$	0.224	0.27	**6.459**	2924	2918	**11625**	0	16	**99**
$802.11_{w/oRTS/CTS}$	0.122	0.14	**5.792**	4022	4064	**11660**	0	18	**96**
$802.11e_{RTS/CTS}$	0.072	0.12	**0.51**	361	600	**2864**	0	54	**348**
$802.11e_{w/oRTS/CTS}$	0.065	0.11	**0.589**	353	736	**2622**	0	78	**321**
$Rendezvous$	0.064	0.125	**0.127**	359	210	**220**	0	1.5	**4**

Table 1: Average Medium Access Delay, Backoff Slots and ReTx Attempts per second (*MAC Delay = Backoff + Queueing delay + Retransmissions*)

hardware queues with varying contention parameters. A detailed description of the traffic shaper design is omitted due to space constraints.

Coexistence with existing 802.11 clients: To minimize the impact of a possible delay at the start of an $RTXOP$ due to ongoing transmissions, the owner *Rendezvous* node grabs the medium as soon as it becomes idle, but ensures that its transmissions do not extend beyond the scheduled end time of the $RTXOP$. To prevent any conventional 802.11 clients from transmitting during a scheduled $RTXOP$, the *duration* field of each transmitted packet in scheduled access mode is set to the remaining $RTXOP$ duration, similar to the transmission of packets in the IEEE 802.11e TXOP. [2] This triggers the virtual carrier sensing mechanism of unmodified 802.11 clients and prevents them from transmitting during a reserved $RTXOP$.

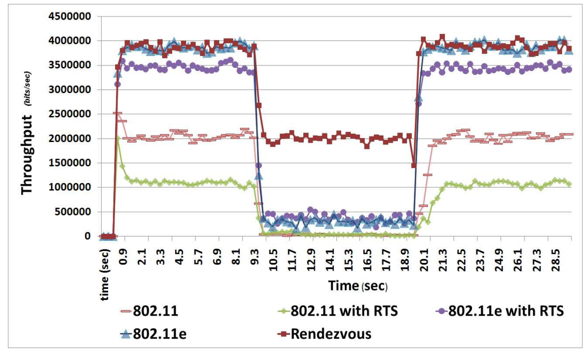


Figure 6: Flow from node A to node B.

5. PRELIMINARY EVALUATION

We evaluate the *Rendezvous* MAC protocol using the $OPNET$ simulator and a testbed implementation on a four node indoor testbed comprised of commodity 802.11 devices. Packet collisions due to hidden terminals are common in a typical network with several clients and access points. In a multi-hop network, a client may be surrounded by multiple hidden nodes. To closely examine the benefits of using a dynamically scheduled application aware channel access protocol over a contention based MAC, we consider a simple scenario with four nodes as shown in Figure 5. In the shown topology, there are two active flows. The first flow is from node A to B and the second flow from C to D. Since nodes A and C act as hidden terminals, transmissions from C cause interference at B.

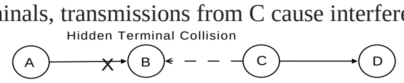


Figure 5: Topology with hidden nodes A and C.

5.1 Simulation results

We configure all the nodes to operate at a fixed transmission rate ($11Mbps$) of 802.11g to avoid the effect of rate adaptation schemes. We use backlogged UDP traffic for throughput measurements and bi-directional video streams to study the behavior of MAC parameters. The UDP packet size is varied to follow an exponential distribution with a maximum size of 1024 bytes.

In the shown topology, node A initiates a UDP transfer to B at *t=1 sec*; node C starts transmitting to D at *t=10 sec* and stops at *t=20 sec*. As shown in Figure 6, the throughput between node A and B drops at time *t=10 sec* due to the hidden terminal interference and increases to the offered load at time *t=20 sec* when the flow from C to D finishes. Table 1 shows the various MAC parameters at nodes A and C before and during the hidden terminal interference. Node A *initial* shows the values of MAC delay, random backoff slots and retransmission attempts before the flow from node C starts. The node C and A *hidden* columns show metric values between time *t=10 sec* and *t=20 sec* when flow from C to D interferes with the reception of packets at B (flow A to B).

IEEE 802.11 & 802.11e: A closer look at the node A *hidden* column in Table 1 reveals that transmissions from A to B face interference from the flow C to D. This is evident from the increased number of backoff slots, MAC delay and retransmission attempts at node A as compared to its initial state (node A *initial*) before C started transmitting. On the other hand, node D successfully receives packets from C, reflected in the low retransmission attempts at node C. Due to reduced contention parameters in 802.11e, node A chooses a small contention window despite failed transmission attempts, and hence is able to successfully transmit more packets than 802.11. Table 1 shows that with the 802.11 protocol node A spends almost 37% of the time in backoff ($11625 slots * 32\mu s/slot = 372ms$ wait period each second), as opposed to node C, which spends less than 10% of the time in backoff. Similar trends in the values of random backoff, MAC delay and retransmission attempts can be seen for the 802.11e protocol. Furthermore, interference from the hidden node C completely suffocates the flow from node A to B in both the 802.11 and 802.11e protocols, as shown in Figure 6.

Surprisingly, even the use of the RTS/CTS mechanism, which was designed to avoid hidden terminal interference, is also not able to prevent the starvation of the flow from A to B. This is because an RTS/CTS exchange is only able to guarantee the successful transmission of one packet. The reduced contention parameters of 802.11e result in increased retransmission attempts and subsequently higher throughput than 802.11. Figure 6 also shows that the 802.11 protocol recovers slowly, as compared to 802.11e, due to higher contention parameters.

[2]IEEE 802.11 clients set their *Network Allocation Vector* (NAV) to the duration value contained in overheard data packets and RTS/CTS messages. During this period, clients do not initiate any new transmissions even if their *clear channel assessment* (CCA) function does not indicate traffic on the medium.

Application aware Rendezvous: In contrast to the 802.11 protocols, *Rendezvous* leverages a simple yet powerful optimization of using each successful REQ/REP exchange to establish multiple future channel reservations and prevent the hidden nodes from transmitting during the reserved $RTXOP$s. This results in scheduled channel access and hence a lower number of retransmissions, as shown in Table 1. As can be seen Figure 6, *Rendezvous* is able to successfully prevent starvation of the flow from A to B. Also, the two flows (A–> B and C–> D) now share the channel equally (not shown here due to space constraints).

Since *Rendezvous* establishes dynamic schedules for transmissions, the medium access delay, number of backoff slots and retransmission attempts are an order of magnitude lower than with the 802.11 and 802.11e protocols, proving that *Rendezvous* is able to successfully reduce contention (refer to Node A *hidden* columns in Table 1). Most importantly, the channel access delay for node A in *Rendezvous* (0.127 secs) is almost five times lower than in the 802.11e protocol (0.589 secs) and an order of magnitude lower than 802.11 (5.792 secs). Thus *Rendezvous* allows node A to extend its sleep period significantly as compared to 802.11e, resulting in improved battery life.

5.2 Experimental results

We now present results from our four node indoor testbed. We arrange the four nodes in a linear topology (nodes A-D as in Figure 3) and measure end-to-end throughput between end nodes over a three hop connection. We use a fixed transmission rate of 54 Mbps on each link to isolate our performance results from throughput variations due to different rate adaptation schemes. The *iperf* utility is used to conduct 10 rounds of throughput measurements, where each round lasts 10 seconds. We observe an average end-to-end throughput of $16.8Mbps$ at three hops with *Rendezvous*, $12.2Mbps$ with 802.11e, and $10.1Mbps$ with 802.11. The throughput benefit comes from the fact that during a scheduled $RTXOP$ a node does not perform random backoff. All packets are separated by an SIFS interval and are bulk-acknowledged at the end of the TXOP. Scheduled access reduces idle time spent in contention and packet loss. In our testbed, we use a time-cycle of 30ms (suitable for Skype, GoogleTalk and video streaming traffic characteristics as shown in Section 2). Nodes A, B and C each transmit for an $RTXOP$ duration of 8ms with the remaining 6ms for random access. We have also successfully used this setup for real-time transfer of bi-directional high-quality video streams (data rate = 4Mbps for each stream) between the end nodes. Since *Rendezvous* allows a node to enter sleep mode during the scheduled $RTXOP$s of other nodes, the wireless interfaces of nodes A and D need to remain powered on for only one third of the time, while nodes B and C must stay powered on for only two thirds of the time, without any loss in throughput performance or latency. Thus, in our experimental setup, *Rendezvous* improves the battery life of nodes A and D by up to 300% and that of nodes B and C by up to 150%.

6. DISCUSSION AND CONCLUSION

In the above sections, we presented a case for enabling the flow of information about application specific traffic characteristics between layers of the networking stack. We believe that contrary to the conventional notion of independence between applications and the underlying networking protocols, there is a wide range of interesting research problems that can use application awareness at the lower layer protocols to improve the efficiency of mobile devices and spectrum utilization. In particular, short term proactive medium reservations, based on application awareness, can be used in a more generic setting such as:

Multi-Hop reservations: In multi-hop networks, a node can proactively reserve the medium along a path to reduce end to end delay, prevent hidden terminal interference and limit traffic at ingress nodes to avoid starvation of nodes far away from the gateway.

Multiple channel MAC extension: In multi-channel MAC protocols that require nodes to negotiate a destination channel and spectrum bandwidth to transfer data packets [12], the overhead of negotiation can be reduced by providing information about the application characteristics to the MAC layer and using a dynamic periodic schedule to reuse a single negotiation for multiple data transactions.

In summary, we have used application awareness at the MAC layer to proactively reserve the wireless medium for short durations and achieve quasi-TDM medium access over small timescales. With the help of a simple yet powerful mechanism of explicit future medium reservations based on the real-time application characteristics, *Rendezvous* is able to achieve efficient channel utilization and reduced power consumption in single and multi-hop wireless networks. By means of simulations and a testbed implementation we show that an application aware approach to medium access can yield significant performance and energy benefits in wireless devices. While preliminary results from our testbed look promising, we believe that *Rendezvous* is just the first step in this direction. This paper provides a proof of concept for the development of application aware protocols. Our aim is to conduct and motivate future research in the domain of application awareness of lower layers of the networking protocol stack.

7. REFERENCES

[1] R. Chandra, J. Padhye, and L. Ravindrananth, "Wi-Fi Neighborcast: Enabling Communication Among Nearby Clients," in *HotMobile '08*, Napa Valley, CA, USA, February 2008.

[2] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones," in *MobiSys '07*, Puerto Rico, June 2007.

[3] M. Anand, E. Nightingale, and J. Flinn, "Self-Tuning Wireless Network Power Management," in *MobiCom '03*, San Diego, CA, USA, September 2003.

[4] P. Seeling, M. Reisslein, and B. Kulapal, "Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A tutorial," *IEEE Communications Surveys*, vol. 6, no. 3, pp. 58–78, 2004.

[5] CNN IBNLive News, http://www.ibnlive.com.

[6] M. Li, M. Claypool, and B. Kinicki, "MediaPlayer versus RealPlayer - A Comparison of Network Turbulence," in *IMW '02*, Marseille, France, November 2002.

[7] P. Bahl, "ARMAP - An Energy Conserving Protocol for Wireless Multimedia Communications," in *IEEE PIMRC '98*, Boston, MA, USA, September 1998.

[8] S. Nanda, D. Goodman, and U. Timor, "Performance of PRMA: A Packet Voice Protocol for Cellular Systems," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 3, pp. 584–598, August 1991.

[9] A. Sharma and E. M. Belding, "FreeMAC: Framework for Multi-Channel MAC Development on 802.11 Hardware," in *PRESTO '08*, Seattle, WA, USA, August 2008.

[10] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," in *ACM TOCS '00*, vol. 18, August 2000, pp. 263–297.

[11] MadWifi, http://www.madwifi.org.

[12] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu, "Allocating Dynamic Time-Spectrum Blocks for Cognitive Radio Networks," in *MobiHoc '07*, Montreal, Canada, September 2007.