

Quality of Service Guarantee on 802.11 Networks

Srikant Sharma Kartik Gopalan Ningning Zhu
Pradipta De Gang Peng Tzi-cker Chiueh
Rether Networks Inc.

99 Mark Tree Road, Suite 301, Centereach, NY 11720
rni@rether.com 631-467-4381

Abstract

Rether[1] was originally developed to support guaranteed Quality of Service (QoS) for shared Ethernet LANs. With the growing popularity of wireless LANs, we modified the Rether protocol to provide QoS guarantee on wireless networks. In this paper, we present the design and implementation of the Wireless Rether protocol for 802.11 networks. Wireless Rether supports QoS for TCP, UDP and ICMP traffic in both upstream and downstream directions. The protocol can seamlessly inter-operate with any priority-based QoS mechanisms (such as Diffserv [3]) on the wired networks that connect the wireless access network to the rest of the internet. QoS requirements of real-time applications are specified as a simple configurable policy table. Legacy networking applications can benefit from QoS guarantees provided by Wireless Rether without requiring any modifications.

1 Introduction

With the growing popularity and acceptance of 802.11 wireless LANs, it is essential to look beyond connectivity and security issues and focus on Quality of Service (QoS) support for advanced multimedia applications, such as one-way video playback or two-way audio/video communication. Typically wireless LANs serve as an access network to the wired infrastructure for mobile terminals, as shown in Figure 1. Mobile hosts are connected to the rest of the network through Layer-2 base stations (or access points) and Layer-3 edge routers.

Both media streaming and audio/video conferencing are sensitive to packet latency and effective bandwidth characteristics of the underlying network. On the wired network, IETF's Integrated Services [5] and Differentiated Services [3, 4] architectures are available to support guaranteed QoS and traffic prioritization respectively above the link layer. IEEE's 802.1p is a traffic prioritization standard

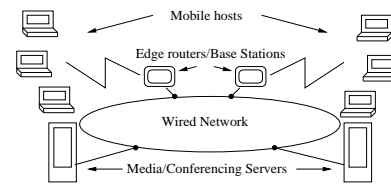


Figure 1. Wireless LANs with mobile hosts typically lie at the edges. Media playback servers and conferencing servers reside in the wired network and handle transfer of media data to/from mobile hosts.

for switched Ethernet environments. However, no commercially available QoS solution exists for wireless LANs, particularly 802.11 networks.

Since wireless LAN is a shared medium, collision arises and subsequently some form of random back-off mechanism is triggered when multiple nodes attempt to send data simultaneously. Therefore special media access control is required to avoid collision, to guarantee deterministic network access, and eventually to support differentiated QoS for different network nodes on a wireless LAN. The basic idea of Rether [1] is to repeatedly circulate a token among the nodes of a LAN within a fixed-length cycle, and to permit only the network node that currently holds the token to transmit data. The amount of data a token holder can send depends on its QoS requirement. This way there is no collision by construction and every node on the LAN can reserve a different amount of the shared wireless link bandwidth. A unique feature of Rether is that it was designed to be a software-based protocol that can run on existing Ethernet hardware, thus allowing users to leverage their existing network hardware investment.

Although Rether appears to be immediately applicable to wireless LANs, there are major technological differences between Ethernet and 802.11 networks that require significant re-thinking of the details of Rether. This paper de-

<i>Application</i>	
<i>BSD Sockets</i>	
<i>TCP</i>	<i>UDP</i>
<i>IP</i>	
<i>Wireless Rether</i>	
<i>Network Device Driver</i>	

Figure 2. Wireless Rether is implemented as a layer between the link-layer and the IP protocol layer. At this position, it can exercise QoS control over all outgoing traffic.

describes the design and implementation considerations of a wireless variant of Rether, which is the first known system that provides guaranteed QoS on 802.11 networks.

2 Design Issues in Wireless Rether

In this section we discuss important design issues in the development of the Wireless Rether protocol, the solutions we chose and their rationale.

2.1 Hardware vs Software Implementation

Wireless Rether protocol can be either implemented on the wireless LAN card or as a software layer above the network hardware’s device drivers. The advantage of the hardware approach is that it can significantly reduce the overhead involved in passing the token between hosts. The disadvantages of the hardware approach are that the solution becomes tied to individual vendor’s hardware, the cost of hardware implementation is significantly higher, and there is less flexibility in configuring QoS policies.

On the other hand, a software-based approach has the advantages of being able to work with any vendor hardware, being cost-effective and providing much more flexibility in configuring QoS policies and in modifying implementation details of the QoS mechanism. Because our previous experiences with Rether showed that with careful protocol and software design, the token passing overhead can be kept to a minimum, we decided to adopt the software approach, whose structure is shown in Figure 2.

2.2 Peer-to-peer vs. Centralized Token Passing

A central design decision in Wireless Rether is how the token is passed from one host to the next. One alternative is to maintain a logical ring among the wireless hosts and to implement a *peer-to-peer token passing* protocol. The

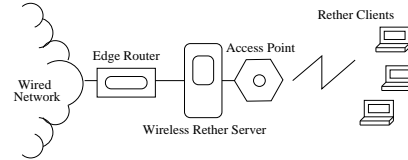


Figure 3. Wireless Rether Architecture. Each edge network is augmented with a Wireless Rether Server (WRS) that acts as a bridge between the Layer-2 access point and the Layer-3 edge router. All mobile hosts are equipped with a Wireless Rether Client (WRC) module.

host holding the token transmits the token to the next logical neighbor in the ring and transmits an ACK to the previous logical neighbor in the ring. With such a distributed token maintenance protocol there is no single point of failure in the network and hosts can join and leave the the ring at any time. This was the scheme used in [1] and it worked well in the context of shared Ethernet segments, where all hosts can communicate with each other directly. In the context of wireless LAN, since mobile hosts can move out of each other’s transmission range, direct token passing between wireless hosts is infeasible. However, since all hosts are assumed to be reachable from the access point, the access point is in a better position to relay the token.

As a result of the above consideration, we chose to implement a *centralized token-passing protocol* in which a central server, called *Wireless Rether Server* (WRS) is placed right between the access point and the wired network and is responsible for granting the token to wireless LAN hosts, called *Wireless Rether Clients* (WRC). This centralized architecture, shown in Figure 3, has several advantages. First, it is the WRS and not the token that maintains the QoS-related state. Therefore loss of the token is not fatal to the proper functioning of the Wireless Rether protocol. Secondly, because the WRS can intercept all traffic entering and leaving the wireless LAN, it can snoop on the wireless channel to determine the end of a packet that a WRC sends and reduces the token passing overhead by eliminating the need of ACKs from WRCs. A potential problem with Wireless Rether is that it now introduces a single point of failure in the WRS. This is not a major concern to us because the access points and edge routers are also in the same position. Overall, the centralized token passing protocol is much simpler, lightweight, and efficient than its distributed counterpart.

Ideally one can combine the WRS and the access point into one device. This would have lead to a cleaner hardware set-up. However, this integration is not currently possible because commercial off-the-shelf access points do not ex-

pose any programming interface to add third-party code. In retrospect, this limitation is a blessing in disguise, because the separate-WRS architecture requires the resulting Wireless Rether design to be independent of and thus able to inter-operate with 802.11 access points from multiple vendors.

2.3 Work-Conserving vs. Non-Work-Conserving

Within a cycle, the token first visits those network nodes that have made bandwidth reservation (called real-time or RT nodes), and after all the RT nodes have been visited the token visits the network nodes that potentially have best-effort traffic to send (called non-real-time or NRT nodes). Note that *every* wireless LAN node is an NRT node. If the token cannot visit all NRT nodes within a cycle, it continues to visit the NRT nodes in the next cycle starting from where it leaves off in the current cycle. Wireless Rether supports a non-work-conserving service model because even when there is no NRT traffic, the token is still passed among NRT nodes until the current cycle ends. That is, RT nodes can never send data at a higher rate than their reservation even when other RT or NRT nodes have less data to send. A non-work-conserving model reduces the extent of data burst and thus decreases the delay jitters that applications experience. In addition, this model fits well with the support for NRT or best-effort traffic.

2.4 Bandwidth Reservation

As mentioned earlier, the QoS mechanism of Wireless Rether needs to work seamlessly with the QoS mechanism in the wired wide area network in order to achieve end-to-end QoS for applications. One of the first considerations is whether or not to support end-to-end application level signaling such as RSVP [5]. Such a signaling mechanism would be ideal from perspective of new QoS aware applications which can be written from scratch to make use of signaling. However, this is impractical for two main reasons. First, end-to-end per-connection signaling requires that intermediate routers carry state about each connection. This is an impractical requirement and one of the main reasons that RSVP did not succeed. The second reason is that many legacy applications exist which require QoS guarantees, but cannot be rewritten to make use of end-to-end signaling. As a result, only Diffserv like QoS guarantees are feasible in the wired wide area network context, in which predefined traffic classes receive bandwidth reservations.

One of the primary goals of developing Wireless Rether protocol was to support QoS for legacy and third party applications without modifying them. Since such applications cannot perform explicit reservation, there is no direct way of determining their bandwidth requirements.

The solution is to use an indirect way of determining the bandwidth requirements. The end-points for such network applications are network ports. A commonly used convention is port based reservation mapping. In such a scheme the network port number used determines the bandwidth requirements. A further refinement is to use the network addresses as well. The reservation mechanism in Wireless Rether uses a *quintuple specification* of $\{\text{SrcAddress/Mask}, \text{DestAddress/Mask}, \text{SrcPortRange}, \text{DestPortRange}, \text{Bandwidth}\}$. Except the bandwidth specification all other fields can be wild cards. If a matching specification is found for any new packet stream, the reservation request for the stream is sent to the WRS. The WRS performs admission control check to admit or reject the reservation request for the new stream. If the request is admitted the network packets of the corresponding stream are queued in a special real-time (RT) packet queue and dispatched by the network scheduler according to the QoS specifications. If the request is rejected, the stream is treated as a best effort stream.

2.5 Support for TCP

Wireless Rether protocol is designed to reserve bandwidth for UDP, TCP and ICMP traffic types. TCP traffic is inherently bidirectional in nature due to presence of TCP ACKs and hence requires special consideration. In order to guarantee bandwidth reservation for TCP data traffic in one direction, it is necessary to reserve bandwidth for TCP ACKs in the reverse direction since non-arrival of ACKs in time would trigger TCP's congestion control mechanism causing it to reduce transmission rate.

Wireless Rether can transparently detect TCP streams and perform reverse bandwidth reservation for TCP ACKs. WRC module snoops on each outgoing packet. If the packet happens to be a TCP ACK, it consults the policy table to determine if the data traffic for the same TCP stream in the reverse direction has an associated reservation in the policy table. If this is the case, the WRC module makes reservation for TCP ACKs for the amount of 10% of reservation for data stream in the reverse direction.

2.6 Packet Queuing and Scheduling

Each outgoing packet in WRC is classified by Wireless Rether as an RT or best-effort packet. Each RT packet is queued in a corresponding RT queue that is created by Wireless Rether when the first packet belonging to the RT stream is encountered by the classifier. Best-effort packets are classified into three categories, namely, *control* NRT, *special* NRT and *simple* NRT. Control messages are Wireless Rether's protocol messages. Special NRT packets are the urgent messages but do not require any bandwidth reser-

vation (such as ICMP, IGMP and application-level control messages). Any other best-effort traffic is classified as NRT.

Whenever a token visits a WRC, its packet scheduler can dispatch packets from appropriate queues using either *time-based* or *byte-based* policy. Time-based policy dispatches packets based on share of token cycle time of the WRC whereas byte-based policy dispatches packets based on number of bytes reserved by a WRC during each cycle. Dispatching packets based on time share gives a fairly good control over the duration of transmission but results in a bursty dispatch of packets when the wireless bandwidth is better than expected. Dispatching packet based on number of bytes to be transmitted during each cycle avoids burstiness but results in long token holding time if the link bandwidth drops below the expected level. To overcome these problems we implemented a *hybrid* packet scheduler in which the limiting parameters are both the reserved time share and the data share per cycle. In this approach the packet are dispatched till either the time share expires or the required amount of data is transmitted. This scheme properly utilizes bandwidth in case of conservative reservations and also deals gracefully with the variable link bandwidth conditions.

Every queue has a limit on the maximum queue size in order to prevent excessive consumption of system buffers. Packet in excess of the limit can be dropped or diverted to the best-effort queue instead of the RT queue. Diverting excess RT packets to best-effort queue would result in out of order delivery of packets to the destination. These packets would ultimately be dropped at the destination if they do not reach within the buffering window, negating the very purpose of initially diverting the excess packets. For this reason, we chose the option of dropping excess packets but placed a fairly large limit on the RT queue sizes to avoid unnecessary drops.

2.7 Roaming Support

One of the principal advantages of wireless LANs is mobility of hosts using Mobile IP protocol [6]. *Foreign agents* located in the new subnet periodically broadcast *advertisements* so that any new mobile IP host moving into the subnet can register with the foreign agent and keep its connections alive.

One of the design goals in Wireless Rether was to support mobility between subnets so that protocols such as Mobile IP can operate smoothly. In the current design, the WRS periodically broadcasts *beacon* messages on the local wireless subnet such that any new host can register with the WRS by responding with a *register* message. When a mobile host moves from one subnet to another, its network card (or the card's device driver) detects the new access point and performs a link-level handoff to the new access point.

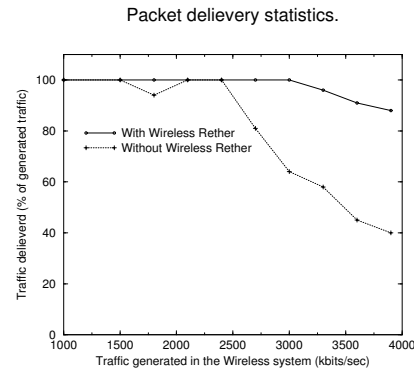


Figure 4. Packet loss with three senders transmitting. Wireless Rether reduces packet loss from 60% to a mere 10%.

(We used Lucent Orinoco wireless network cards and access point in our testbed and found this to be the behavior. It is quite probable that cards and access point from some other vendor might behave differently.) Following the low level handoff, the mobile host can receive *beacon* messages from the WRS in the new wireless subnet and register with it by sending a *registration* message. From this point onwards, the mobile host can receive *foreign agent advertisements* of mobile IP protocol, register with the new foreign agent and maintain its IP level sessions.

3 Performance Measurements

The peak link bandwidth of the hardware used is 11 Mbps. But the normal link bandwidth that can be observed in the wireless network is around 6.5 Mbps when a single sender is transmitting 1500 byte packets in the absence of collisions. When there are multiple senders the collisions result in packet loss and hence reduction in the overall throughput. Our experimental setup consisted of three hosts transmitting simultaneously - two of them upstream and one downstream. Figure 4 shows a comparison of packet loss observed in the presence and absence of Wireless Rether. Without Wireless Rether, we can observe packet losses even at traffic load of 2.5 Mbps, beyond which packet losses increase drastically to 60% when the load is around 4 Mbps. In contrast, with Wireless Rether the packet drops stay within 10% of the total network load. The primary reason for packet losses in Wireless Rether was that the access point was unable to handle heavy loads, and thus resulted in dropped packets.

Another advantage of Wireless Rether is that it reduces the bursty nature of the traffic in the network. Figure 5 shows the throughputs of a wireless network in the presence and absence of Wireless Rether. In both cases three senders

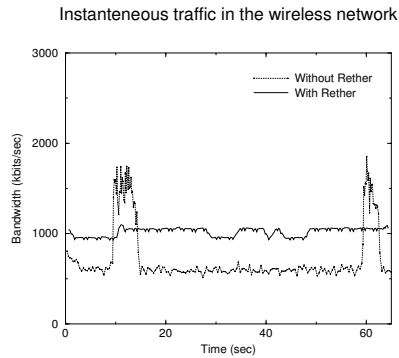


Figure 5. Wireless Rether reduces the the bursty nature of transmissions and increases the throughput.

attempt to transmit data at a rate of 1.1 Mbps. Without Wireless Rether the traffic is bursty and the actual throughput of each connection is lower than the transmission rate of 1.1 Mbps. In the presence of Wireless Rether, the required throughput is achieved and the burstiness is reduced as well.

With Wireless Rether, every transmission is accompanied by a token and an acknowledgment for the token. For small bandwidth reservations the data to be transmitted per cycle can be as low as a single packet. In this case Rether needs to pay a serious performance penalty of two small packets in each token visit. The relative overhead is highest for small payload packet sizes and decreases with increasing payload packet sizes. The total token overhead is proportional to the number of Wireless Rether clients in the network, and is around 70% when the maximum number of WRCs are active and reduces to around 7% to 11% when a single WRC is active. The maximum number of WRCs is 13 for small payload packets and 6 for large payload packets at a cycle length of 33ms. Figure 6 shows how the throughput of Wireless Rether varies with packet size and number of clients.

4 Conclusion

In this paper we presented the design and implementation of Wireless Rether protocol that provides QoS guarantees on 802.11 networks. Wireless Rether grew from attempts to adapt the original Rether [1] protocol to wireless LANs. It adopts a centralized token passing architecture and supports a non-work-conserving service model that reduces the extent of data bursts and decreases delay jitters. Sequential token-ACK message exchanges between WRS and WRCs ensure data transmission on the wireless channel is guaranteed to be collision free, thus increasing the effective throughput of the channel. Wireless Rether sup-

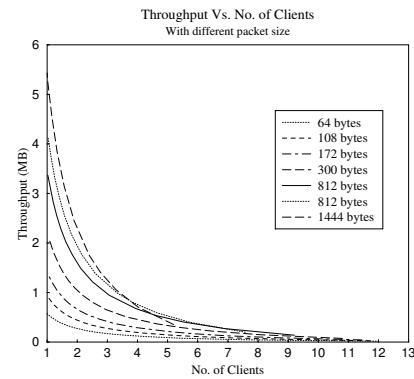


Figure 6. Maximum throughput possible in Wireless Rether for different packet sizes and different number of clients.

ports port-based reservation for real-time traffic streams. For real-time TCP streams, it transparently performs reverse bandwidth reservation for TCP ACKs. It uses a hybrid packet scheduler at WRCs that limits the transmission rate of real-time streams based on time-share as well as byte-share per cycle. Wireless Rether inter-operates smoothly with higher layer protocols such as Mobile IP to support roaming of mobile terminals. Our performance measurements validate the ability of Wireless Rether to provide QoS guarantees over wireless LANs.

References

- [1] Tzi-cker Chiueh, Chitra Venkatramani, "Supporting Real-Time Traffic on Ethernet," Proceedings of IEEE Real-Time Symposium, 1994.
- [2] Prashant Pradhan, Tzi-cker Chiueh, "Real-Time Performance Guarantees over Wired/Wireless LANs," IEEE Conference on Real-Time Applications and Systems, June 1998.
- [3] W. Feng, K. Kandlur, D. Saha, K. Shin, "Adaptive packet marking for providing differentiated services on the internet," Proc. Intl. Conf. on Network Protocols, October 1998.
- [4] P. Almquist, "Type of Service in the Internet Protocol Suite," RFC 1349, July 1992.
- [5] Lixia Zhang, Steve Deering, et. al., "RSVP: A New Resource Reservation Protocol," IEEE Network Magazine, September 1993.
- [6] C. Perkins, ed., "IP Mobility Support," RFC 2002, Oct. 1996.