END-TO-END CONGESTION CONTROL FOR BEST-EFFORT TRANSMISSION

Lei Chen and Wendi Heinzelman

Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627

ABSTRACT

UDP is the standard best-effort transport layer protocol. A well-known problem with UDP is that, unlike TCP, it does not include congestion control. Thus UDP connections have no way of detecting or reacting to network congestion. This is particularly a problem in wireless networks, where a significant loss in achievable capacity occurs when using UDP transport over a congested multi-hop network running the 802.11 MAC. Considerable energy is wasted on transmitting packets that will be dropped during transmission. Since mobile ad hoc networks lack energy and bandwidth, it is necessary to use congestion control to improve the network performance while using UDP as the transport layer protocol. In this paper, we explore the use of UDP with congestion control. Our simulations show that adding congestion control to best-effort traffic can greatly improve network capacity and dramatically reduce energy dissipation.

1. INTRODUCTION

Improving capacity and reducing energy consumption are the most important problems to be solved for mobile ad hoc networks (MANETs) in order to put MANETs into practical use. Early research on capacity issues shows that the maximum capacity of a long chain of nodes in isolation is 1/4 of the channel bandwidth, but simulation results using the IEEE 802.11 MAC show the achievable capacity is only 1/7 of the channel bandwidth because the early nodes in the chain starve the later nodes from channel access [1]. Thus, effective use of the channel and reducing packet loss are highly desirable. Also, most mobile devices operate on batteries, so power consumption is also an important issue. Under high congestion, many packets will be dropped before reaching their destination. Considerable energy is wasted on transmitting these dropped packets. Even in wired networks that are not constrained by limited energy and bandwidth, incorporating congestion control into best-effort traffic has gained attention (e.g., [2] [3]). Researchers argue that it is necessary to use end-to-end congestion control in wired networks in order to improve the overall networking performance. In this paper, we propose using feedback information from the destination to estimate the channel traffic and pre-drop packets to improve performance. We do not want to ensure reliable delivery as in TCP, only to ensure that the UDP source does not overwhelm the limited bandwidth of the wireless channel and dissipate energy for transmitting packets that will be dropped ultimately due to network congestion. This will improve the overall performance in terms of throughput and energy per packet per hop for all hosts on the network.

2. CONGESTION CONTROL

Traditionally, the best-effort service model treats every packet the same without any discrimination or explicit delivery guarantees. Users do not need to request permission before transmission, so the performance of every flow is not only affected by itself but also by other flows. This uncoordinated behavior seriously affects the performance of ad hoc networks. Incorporating congestion control into the UDP protocol would help alleviate this problem.

In this paper, we propose two end-to-end congestion control protocols: UDP with Additive Increase and Multiplicative Decrease (UDP-AIMD) congestion control, and UDP with Multiplicative Increase and Multiplicative Decrease (UDP-MIMD) congestion control. These two protocols use explicit feedback from the destination. They are built on UDP and add two features to the UDP protocol: (1) receiver feedback (ACK packets) that informs the sender of the amount of data received at the destination and the time interval in which those data were received, and (2) an adaptive sending rate algorithm at the sender to adjust the sending rate using the feedback information. We chose a rate-based scheme instead of a window-based scheme as TCP uses because window-based protocols introduce burstiness, which is not desirable in packet delivery for wireless mobile ad hoc networks. It has been proven that rate-based congestion control algorithms provide better performance than window-based algorithms in wired networks as well [4].

The most important function in both algorithms is adjusting the sending rate based on the current receiving rate.

In AIMD, the sending rate is adjusted using an additive increase and multiplicative decrease scheme, which is the same as TCP's scheme. The sending rate adjustment works as follows:

If ($\frac{SendRate}{ReceiveRate} \geq gap$)
    $SendRate = \frac{SendRate}{2}$
Else if ($SendRate \geq SSThresh$)
    $SendRate = SendRate + IncRate$
Else if ($ReceiveRate \times 2 \geq SSThresh$)
    $SendRate = SSThresh$
Else $SendRate = ReceiveRate \times 2$
where $gap > 1$ and $SSThresh$ is the slow start threshold.

In MIMD, the sending rate is adjusted using an adaptive multiplicative control scheme. A slow multiplicative increase and decrease is used when the sending rate is close to the receiving rate, and a large multiplicative decrease is used when the sending rate is far away from the receiving rate. The sending rate adjustment works as follows:

If ($\frac{SendRate}{ReceiveRate} \geq \beta$)
    $SendRate = \frac{SendRate}{2}$
Else if ($\gamma \leq \frac{SendRate}{ReceiveRate} \leq \beta$)
    $SendRate = SendRate \times \epsilon$
Else if ($\zeta \leq \frac{SendRate}{ReceiveRate} \leq \gamma$)
    $SendRate = SendRate \times \eta$
Else $SendRate = ReceiveRate \times \theta$
where $\gamma, \beta, \eta > 1$, $\zeta, \epsilon < 1$, $\zeta < \gamma < \beta$, and θ is a value around 1.

The UDP-MIMD and UDP-AIMD protocols drop packets in the transport layer. However, this is not the only choice; we use this method only to verify that congestion control is needed in mobile ad hoc networks. UDP-MIMD and UDP-AIMD can easily be modified to offer an API to the application layer so that the application can figure out how to scale back its sending rate.

3. SIMULATIONS

To determine the effects of congestion in MANETs, we ran several simulations using the ns-2 simulator [5] to experiment with UDP-AIMD and UDP-MIMD in static and mobile networks. We use the IEEE 802.11 MAC protocol in RTS/CTS/Data/ACK mode with a channel data rate of 2 Mbps. The packet size used in our simulations is 1,500 bytes. All of the simulations are run for 100 seconds, and each plot represents the average of 5 independent runs. To the best of our knowledge, there is no proposed congestion control protocol used for UDP. Therefore the comparison is only between UDP-AIMD and UDP-MIMD compared with UDP without congestion control. We are not arguing that our protocol is the best, only that it is important to use congestion control with UDP in MANETs.

3.1. UDP-AIMD

First we examine UDP-AIMD using a simple six node chain topology. Each node is in the transmission range of its neighbors, and the interference range is two hop distance. The head node is the source and the tail node is the destination. We vary the source feeding rate from 0.2 Mbps to 2 Mbps and measure the received throughput and energy used for transmitting every packet. The simulation results are shown in the Figure 1.

From Figure 1, we can see that, regardless of the additive increase rate, using congestion control provides much better performance than not using congestion control, in both end-to-end throughput and energy consumption. Also, either 12Kbps or 24Kbps is the "optimal" sending rate. However, the optimal sending rate varies depending on the network topology, because the difference between the optimal congestion-free sending rate and the congested sending rate determines the optimal additive increase rate. Therefore, it is hard to do performance optimization for a generic network. For example, table 1 shows the total capacity that can be achieved using a topology that is composed of 50 nodes with 5 source-destination pairs. In this case, the optimal additive increase rate is 9Kbps, not 12Kbps or 24Kbps as was the case for the six-node chain topology.

$IncRate(Kbps)$	1	3	5	7	9	12	18	24
$Capacity(Mbps)$	1.52	2.01	1.90	2.06	2.12	1.87	2.07	1.85

Table 1. The capacity using various additive increase rates for a 50 node network with 5 source-destination pairs.

3.2. UDP-MIMD

To determine the effect of UDP-MIMD, we ran tests varying the UDP-MIMD parameters and measured the performance for the six node chain topology. We fixed five parameters used in the protocol and varied the final parameter to see the effects on end-to-end throughput.

The simulations show that the parameters β, θ, and ζ have no effects on throughput, and choosing different γ, ϵ, and η have only a slight impact (we did not include these figures here due to space limitations). Since the performance of UDP-MIMD is not greatly affected by the choice of parameters, it is not necessary to find the "optimal" parameters for different scenarios and traffic patterns, as long as the parameters are chosen in a reasonable range.

The performance of UDP-MIMD does not depend heavily on the selected parameters, whereas the performance of UDP-AIMD does, so we use UDP-MIMD in the following simulations.

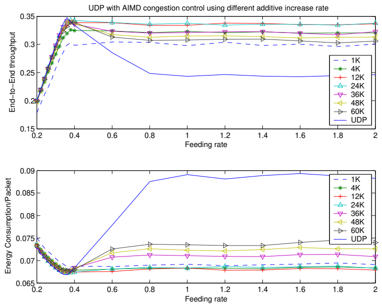

Fig. 1. The end-to-end throughput and energy/packet of the six node chain using AIMD with different additive increase rates.

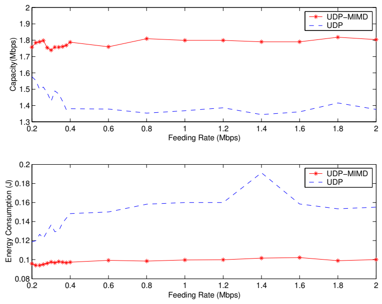

Fig. 2. The capacity and energy/packet/hop improvement averaged over 20 random static scenarios.

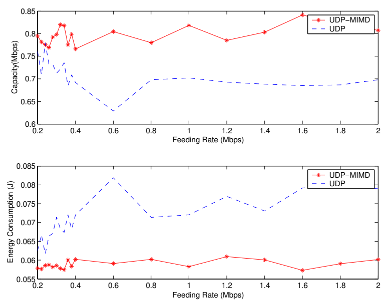

Fig. 3. The capacity and energy/packet/hop improvement averaged over 10 random mobile scenarios.

3.3. Large Scale Random Static Topologies

We investigated the performance of UDP-MIMD in more realistic random scenarios. For these simulations, there are 50 nodes located randomly in a 1000 meters by 1000 meters square. Five nodes are randomly chosen as the sources, and five nodes are randomly chosen as the destinations. The metrics used in measuring the improvement are capacity and per packet energy consumption. Figure 2 shows the capacity and energy results obtained by averaging 20 different scenarios. Average capacity is improved by 25%, and average energy consumption per packet per host is reduced by 47%.

3.4. Mobile Topologies

Our protocol is a congestion control transport layer protocol, which is based on feedback information. Therefore, if the network changes too fast, the feedback cannot reflect the current network status. Therefore we do not expect the algorithm to work well under high mobility. Thus, we choose low mobility scenarios to test our protocol. In the scenarios we choose, each node moves towards a random destination using a speed randomly chosen between 0-3 m/s. Five source-destination pairs that are randomly chosen send packets using a rate between 0.2 Mbps and 2 Mbps. Figure 3 shows the results obtained by averaging 10 different scenarios. Average capacity is improved by 13%, and average energy consumption per packet per host is reduced by 21%.

4. CONCLUSIONS AND FUTURE WORK

This paper shows that adding congestion control to best-effort traffic delivery greatly improves overall network capacity and reduces average energy/packet/hop in mobile ad hoc networks. We explored the use of UDP-AIMD and UDP-MIMD, two congestion control protocols built on top of UDP to reduce congestion in the network. UDP with congestion control improves performance in different types of topologies.

UDP-AIMD and UDP-MIMD only reduce the unnecessary bandwidth cost on a per-flow basis. Furthermore, they do not explicitly take fairness into consideration. Therefore UDP-AIMD and UDP-MIMD do not incorporate any mechanisms for network-wide optimization, nor do they incorporate any way to tradeoff total network throughput for per-flow fairness. Our future research will explore network-wide design using a cost function that combines both overall throughput and per-flow fairness to see how much improvement is possible.

5. REFERENCES

[1] J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris, “Capacity of Ad Hoc Wireless Networks,” in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom ’01)*, 2001.

[2] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet,” in *IEEE/ACM Transactions on Networking 7(4):458-472*, 1999.

[3] P. Gevros, J. Crowcroft, P. Kirstein, and S. Bhatti, “Congestion control mechanisms and the best effort service model,” in *IEEE Network 15(3):16-25*, 2001.

[4] H. Balakrishnan, H. S. Rahul, and S. Seshan, “An Integrated Congestion Management Architecture for Internet Host,” in *Proc. ACM SIGCOMM*, 1999.

[5] UC Berkeley/LNBL/ISI, “The ns-2 network simulator with the cmu mobility extensions,” http://www.isi.edu/nsnam/ns/, 2002.