DiffQ: Differential Backlog Congestion Control for Wireless Multi-hop Networks

Ajit Warrier, Sangtae Ha, Prashant Wason and Injong Rhee
Dept of Computer Science
North Carolina State University
acwarrie,sha2,pwason,rhee@ncsu.edu

Jae H. Kim
Boeing Phantom Works
jae.h.kim@boeing.com

***Abstract*—In this demo, we showcase DiffQ – a congestion control protocol inspired by theoretical cross-layer optimization approaches. DiffQ can support congestion control for network flows that use either single-path or opportunistic multi-path routing.**

Our demo will focus on the performance in single-path routing environments, where contemporary end-point congestion control algorithms like TCP face severe unfairness or even starvation. This is primarily due to the interaction of such protocols with MAC layer unfairness. We demonstrate micro (5 flows) as well as macro-evaluations (60 flows) of such cases. Our demo is conducted on WiSeNet – a 70-node wireless mesh test-bed hosted in the computer science building at NCSU. Distributed over a 100,000 sq ft building, this is one of the largest test-bed installations both in terms of number of nodes and coverage area, hence an ideal testing ground for such scenarios. Experimental results like throughput, MAC-layer statistics, delay, routing path flaps and network buffer overflows are recorded and displayed in real-time and enable a bird's eye-view of the entire network status and allow us to point out various phenomenon as they happen.

I. INTRODUCTION

Several key problem areas for TCP on multi-hop networks have been identified; to name a few – TCP's reliance on losses as a congestion signal, TCP's interaction with wireless routing protocols, TCP's aggressive probing for bandwidth and self-interference between TCP data and ACK packets. As a response to these problems, the research community has proposed several enhancements to TCP, e.g. TCP-FeW [11], TCP-Veno [5], TCP-Westwood [2], TCP-ELFN [7].

However, as the number of such TCP flows increase, we observe significant unfairness or even starvation of a large number of flows due to TCP's interaction with the MAC [6]. The core issue behind this unfairness comes from the wireless medium being a shared medium over the air. In such a network, contention occurs not only among those flows that share the same links or routers, but also among neighboring flows that do not necessarily share the links. One such case is shown in Figure 1 which shows a particular flow scenario with four one-hop flows and one five-hop flow (19 → 10). As the TCP sources of one-hop flows begin probing for bandwidth by increasing their rates, the long multi-hop flow begins to experience congestion, especially at nodes 14 and 5, leading to queue overflows at these nodes. This causes the TCP source at node 19 to *lower* its transmission rate, which further allows the one-hop flows to increase their rates. This vicious cycle continues eventually starving the long flow completely. We demonstrate this scenario on our testbed. For our demonstration, we use TCP-SACK [10] as a baseline and TFRC-ECN where we switch off TFRC's [4] response to packet losses. Instead, intermediate nodes monitor the interface queue size and set the ECN bit in the IP header whenever the queue size exceeds 75% of the maximum queue size. Marked packets within an RTT are treated as a congestion event.

As we increase the number of concurrent flows on the testbed from 5 to 60, this problem manifests itself in a very severe manner, starving almost 50-60% of the flows.

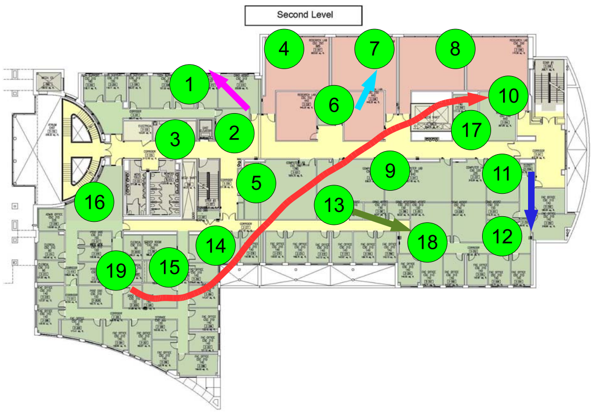

Fig. 1. A small scenario demonstrating MAC unfairness on one floor of the WiSeNet testbed. The full testbed consists of 70 nodes distributed over 3 floors.

II. DIFFQ

DiffQ is a new congestion control protocol for general purpose wireless multi-hop networks. This protocol is designed with the following informally defined goals: (1) it must support traffic carried by non-deterministic multi-path routing over possibly diverse and many paths as well as single path routing. This means congestion control must be scalable to the number of paths being used, (2) congestion control is a service for diverse applications so that it should not require any changes in application operations such as reliability (e.g., coding) and application-level routing (e.g., opportunistic multi-path or single path), and (3) it must improve efficiency in

978-1-4244-1777-3/08/$25.00 ©2008 IEEE

resource usage to achieve high throughput and fairness in resource sharing among concurrent flows.

DiffQ is heavily inspired from existing theoretical work. Our solution is adapted from differential backlog based back-pressure, conveniently called *differential backlog*. The technique was first applied to wireless multi-hop networks by Tassiulas and Ephremides [9] and later used in several follow-up studies [12], [13], [14], [15]. DiffQ implements router-assisted rate control, source rate control and MAC scheduling. We now provide a brief description of DiffQ, for more details please refer to the technical report [16].

Each node maintains a per-destination queue. The queue size for a destination is piggybacked in every transmission, hence disseminating this information to its neighbors. Intermediate nodes select the next destination to schedule based on the *queue differential* for all destinations. The queue differential for a destination at a node is the difference of the queue size for that destination at that node with the queue size for the same at the next hop. The destination with the maximum queue differential is scheduled first, and the HOL (head of line) packet from its queue is sent down to the MAC. The MAC then transmits the packet with a priority proportional to the queue differential, higher the queue differential, higher the priority. This form of scheduling creates a back-pressure starting at congested nodes moving toward the source, allowing sources to regulate their transmission just by observing their local queue size.

We now provide the rationale for the above algorithm. DiffQ looks a lot like a scheduling algorithm. But it also has a unique way of applying back-pressure. Suppose that a flow f is forwarded through a chain of nodes X, Y, Z and so on in that order, and suppose the size of the destination queue of flow f at node Z is reducing as somehow Z can forward the packets of f fast to its next hop. Then it will cause the queue differential at node Y to increase. This has effect of increasing the forwarding rate at node Y because the channel access priority increases with the queue differential. As Y gets more prioritized access, X will be waiting and its queue builds up. After Y's queue gets depleted, then again X's priority increases because its queue differential against Y's queue is rising, and then X will have a higher chance to the channel next. For the opposite case, suppose that Z's next hop is congested so Z cannot forward its packets. Then Z's queue will build up while increasing its priority. In the mean time, Y's queue differential will reduce because Z's queue is increasing and allows less chance accesses for Y. Consequently, Y's queue builds up. This back-pressure will propagate to the source if Z's congestion does not resolve soon enough.

III. DIFFQ ARCHITECTURE

DiffQ has been implemented as a kernel module for the 2.6 series of the Linux kernel (2.6.18 onwards). DiffQ performs queuing and scheduling on every packet being transmitted or received in the Linux networking stack. DiffQ is primarily implemented on top of IP except for prioritized channel access (link layer) and source rate control (transport layer). DiffQ uses the routing support of IP for single path routing while using source routing for multi-path routing in which case, DiffQ bypasses IP routing.

To implement the functions of DiffQ over IP, we need various mechanisms (1) to capture, process and re-inject all IP packets before or after routing in the kernel, (2) to control the MAC priority of each packet transmitted by MAC, and (3) to provide DiffQ information to the transport layer (e.g., queue size information for source rate control). DiffQ uses the Linux Netfilter module to implement the first mechanism, and uses the TOS field of IP header to specify the priority of each packet. The MadWiFi driver is modified to read this field of each packet to get the priority information of the packet and put the packet into appropriate priority queues. The driver outputs packets into the air interface in the order of priority using the queues.

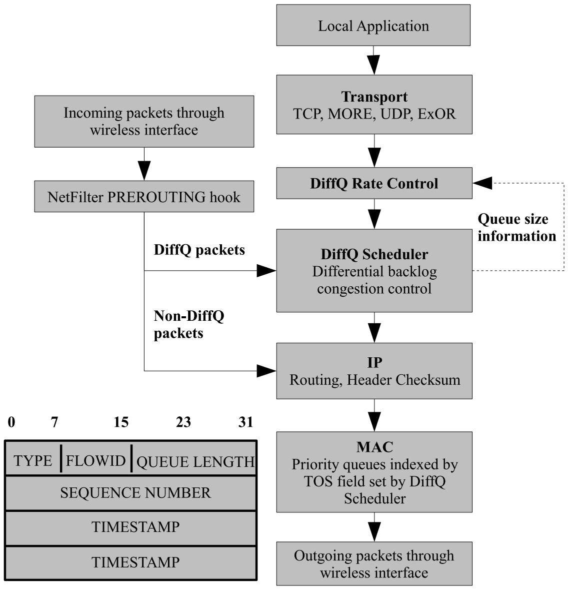

Fig. 2. DiffQ Architecture and Packet Header Format - DiffQ sits on top of IP and provides congestion control services to upper layer transport modules. It also controls the MAC priority of packets for scheduling and performs source rate controls for the transport flows (for support of TCP, it disables TCP's congestion control.

IV. WISENET TESTBED

The WiSeNet testbed consists of 70 wireless nodes distributed over three floors in a building of about 100,000 sq ft space. This is one of the largest testbeds both in terms of number of nodes and coverage area. One floor of this testbed with 20 nodes is shown in Figure 1. Nodes are PCs with 128MB RAM and 266 MHz processor. Figure 3 show one such node. Each node is equipped with two Atheros-based 802.11 a/b/g wireless interfaces (AR 5213/5112) connected to omni-directional antennas. All wireless interfaces are configured to operate in the ad-hoc mode with RTS/CTS disabled, transmission power set to 19 dBm and PHY rate set to 11

Mbps. We use an implementation [1] of the OLSR [8] routing protocol which uses ETX [3] to generate routes.

Fig. 3. Soekris mesh nodes used on WiSeNet.

V. DEMONSTRATION

Our demonstration setup is shown in Figure 4. We connect to our remote testbed through a laptop and execute the experiments described above. Experiment statistics are captured remotely and relayed to this laptop which then displays them in real-time on a console.

REFERENCES

[1] http://olsr.org/.

[2] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. Tcp westwood: end-to-end congestion control for wired/wireless networks. *Wirel. Netw.*, 8(5):467–479, 2002.

[3] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM.

[4] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. *SIGCOMM Comput. Commun. Rev.*, 30(4):43–56, 2000.

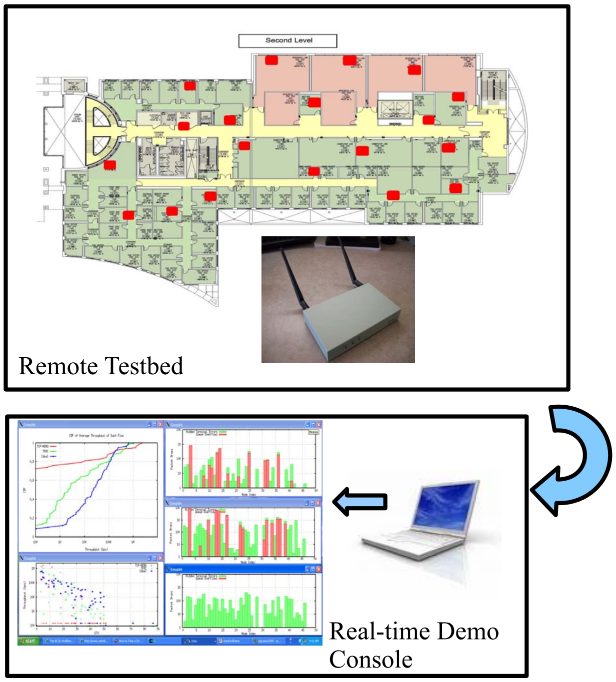

Fig. 4. Above figure shows the overall demo setup. Experimental data is transferred from the remote testbed to the demo site and displayed in real-time on the console.

[5] C. Fu and S. Liew. Tcp veno: Tcp enhancement for transmission over wireless access, 2003.

[6] V. Gambiroza, B. Sadeghi, and E. W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 287–301, New York, NY, USA, 2004. ACM.

[7] G. Holland and N. Vaidya. Analysis of tcp performance over mobile ad hoc networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 219–230, New York, NY, USA, 1999. ACM.

[8] P. Jacquet, P. Mhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol. In *INMIC'01*.

[9] A. E. L. Tassiulas. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1949, 1992.

[10] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. Tcp selective acknowledgement options.

[11] K. Nahm, A. Helmy, and C.-C. J. Kuo. Tcp over multihop 802.11 networks: issues and performance enhancement. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 277–287, New York, NY, USA, 2005. ACM.

[12] M. J. Neely. Energy optimal control for time varying wireless networks. In *INFOCOM'05*.

[13] M. J. Neely. Joint congestion control and media access control design for ad hoc wireless networks. In *INFOCOM'05*.

[14] M. J. Neely. Optimal backpressure routing for wireless networks with multi-receiver diversity. In *CISS'06 (invited paper)*.

[15] B. Radunovic, C. Gkantsidis, P. Key, P. Rodriguez, and W. Hu. An optimization framework for practical multipath routing in wireless mesh networks. In *MSR-TR-2007-81, June 2007*.

[16] A. Warrier, S. Ha, P. Wason, and I. Rhee. Diffq: Differential backlog congestion control for wireless multi-hop networks. In *Tech Report, Computer Science Department, North Carolina State University*.