Noname manuscript No.
(will be inserted by the editor)

The Efficacy of Centralized Flow Rate Control in 802.11-based Wireless Mesh Networks

Kamran Jamshaid · Paul Ward · Martin Karsten · Basem Shihada

Received: date / Accepted: date

Abstract Commodity WiFi-based Wireless Mesh Networks (WMNs) can be used to provide last mile Internet access. These networks exhibit extreme unfairness with backlogged traffic sources. Current solutions propose distributed source rate control algorithms requiring MAC or transport-layer changes on all mesh nodes. This is often infeasible in large practical deployments.

In wireline networks, router-assisted rate control techniques have been proposed for use alongside end-to-end mechanisms. We wish to evaluate the feasibility of establishing similar centralized control via gateways in WMNs. In this paper we focus on the efficacy of this control, rather than the specifics of the controller design mechanism. We answer the question: Given sources that react predictably to congestion notification, can we enforce a desired rate allocation through a single centralized controller? The answer is not obvious because flows experience varying contention levels, and transmissions are scheduled by CSMA/CA MAC using imperfect local knowledge. We find that common router-assisted flow control schemes used in wired networks fail in WMNs because they assume: (1) links are independent, and (2) router queue buildups are sufficient for detecting congestion. We show that non-work-conserving rate-based centralized scheduling can effectively enforce rate allocation. It can achieve results comparable to source rate limiting, without requiring any modifications to mesh routers or client devices.

Keywords Wireless Mesh Networks, 802.11, DCF, CSMA/CA, Max-min Fairness, Congestion Control

K. Jamshaid and B. Shihada
MCSE Division, KAUST, Saudi Arabia
E-mail: kamjam@acm.org

P. Ward and M. Karsten
University of Waterloo, Canada

1 Introduction

Wireless Mesh Networks (WMNs) based on the commodity IEEE 802.11 radios are a low-cost alternative for last mile broadband access. Such networks consist of static mesh routers powered by utility electricity. The mesh routers communicate with each other over multihop wireless links. Client devices connect to their preferred mesh router either via wire or over a (possibly orthogonal) wireless channel. Communication is typically to/from clients through the mesh routers over multiple wireless hops, to a gateway mesh router that has a wired connection to the wider world, typically the public Internet.

The 802.11 chipset is the preferred radio platform in both commercial WMN products and research testbeds. However, these networks often exhibit poor performance characteristics. Multihop flows experience unfairness, including starvation, when competing with nodes closer to the gateway [10, 11, 33]. This is primarily due to the inherent limitations of CSMA/CA MAC in a multihop environment, as well as its operational specifications in the IEEE 802.11 Distributed Coordination Function (DCF) access mechanism. We describe these below:

1. CSMA/CA transmitters located outside mutual carrier sense range may produce misaligned transmissions that result in excessive collisions at a receiver or deprive some nodes of transmission opportunities. As a result, nodes sharing the same wireless channel develop an inconsistent, location-dependent view of the channel state.
2. DCF provides all nodes in a single contention area with equal transmission opportunities (TXOPs). This MAC-level fairness does not translate to end-to-end fairness in multihop networks where nodes closer to

the gateway relay traffic for nodes that are further away.

The impact of these problems can be severe in networks with backlogged traffic; it has been shown that flows closer to the gateway may completely capture the wireless channel at the cost of starving the distant, disadvantaged flows [10]. Without any explicit rate feedback to data sources, this unfairness persists for the duration that the contending flows are active. Existing congestion control protocols such as TCP fail to provide this rate feedback in CSMA/CA-based systems [10, 33]. These problems remain inherent in DCF extensions such as Enhanced Distributed Channel Access (EDCA), which schedules elastic TCP streams using the 'Background' or 'Best Effort' class.

A number of research publications (*e.g.*, [10, 30, 2]) have proposed distributed algorithms that allow traffic sources to compute and enforce flow rate limits based on current contention levels in the network. These algorithms require periodic network-wide flooding of time-varying state information. This requires MAC-layer changes to mesh nodes or transport-layer changes to client devices; these are both often infeasible in large, practical deployments.

In wired networks, router-assisted flow control mechanisms (*e.g.*, [8] have been proposed for use alongside end-host based congestion control protocols. Pure end-to-end flow control schemes cannot provide isolation between flows or ensure rate or delay guarantees; they instead depend on these router-assisted mechanisms for support. We are interested in evaluating the feasibility of establishing similar controls at gateway mesh routers in WMNs providing last mile access. Traffic flows in these networks are primarily directed towards or away from the gateway. This allows the gateway to develop a unified view of the end-to-end flow rates of flows through this gateway, making it a suitable choice for enforcing various resource allocation policy objectives. In particular, we wish to use gateway-enforced control to address flow rate unfairness in WMNs.

In this paper, we focus on the efficacy of such a centralized control, rather than specifics of the controller mechanism design itself. Given a desired rate allocation policy objective (*e.g.*, max-min allocation [3]), we evaluate the effectiveness of gateway rate control in enforcing this objective in a 802.11-based WMN. This evaluation is necessary because multihop wireless network characteristics are distinct from wired networks or even one-hop wireless LANs (WLANs): competing flows in a WMN traverse different number of hops, each flow experiencing varying levels of link contention along its path; further, transmissions along individual links are scheduled based only on the localized view of the CSMA/CA transmitters. We discover that these characteristics render some common router-assisted wired network mechanisms ineffective as gateway-enforceable solutions in WMNs. Work-conserving scheduling techniques, such as Fair Queueing (FQ) or Weighted Fair Queueing (WFQ) [5] are inadequate on their own as they assume independence of links. Similarly, router-assisted probabilistic packet drop techniques including Active Queue Management (AQM) [8] are ineffective because packet losses in a multihop network are spatially distributed and cannot be accurately predicted using the queue size at the gateway router. We describe these fundamental differences in Sect. 5.1 and 5.2.

We show that simple non-work-conserving rate-based centralized scheduling techniques can enforce fairness in 802.11-based WMNs. Link layer retransmissions allow a 802.11 node to recover from wireless losses. When combined with rate-based scheduling, this allows all nodes to obtain their share of the network capacity. We show that even course-grained rate control on net-aggregate traffic passing through the gateway is effective in eliminating unfairness. Further improvements are obtained when we isolate flows using FQ alongside aggregate rate-based scheduling. Finally, rate-based scheduling can be enforced on a per-flow basis, allowing fine-grained control over the resource allocation process. We evaluate and establish the efficacy of these gateway-enforced control techniques in both single-channel and multi-radio, multi-channel WMNs.

The remainder of this paper is organized as follows. In Sect. 2, we explain how DCF leads to flow unfairness and starvation in WMNs with backlogged traffic. We discuss related work, contrasting it with our approach in Sect. 3. In Sect. 4 and 5, we describe various techniques for enforcing centralized rate control in WMNs and evaluate their effectiveness using simulations. These simulations use network capacity models to determine fair share rate information. While such models are extraneous to this work (we are interested in evaluating the efficacy of centralized control *given* a desired rate allocation), for completeness we describe the models used in this paper in A.

2 Flow Unfairness and Starvation in DCF-based multihop networks

A core function of any MAC protocol is to provide *fair* and *efficient* contention resolution mechanism. Here we describe the behavior of 802.11 DCF in multihop networks when the contending nodes are (a) within, and (b) outside, mutual carrier sensing range.

2.1 Nodes within mutual carrier sense range

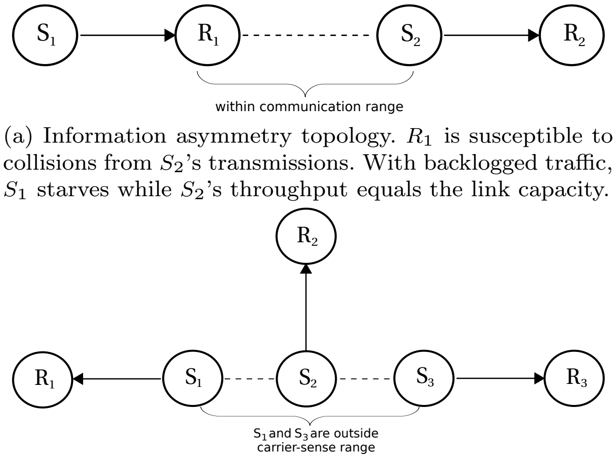
On average, DCF provides equal TXOPs to nodes within carrier sense range. This provides per-station fairness in WLANs where stations communicate directly with the Access Point (AP). However, it does not translate to flow-level or end-to-end fairness in WMNs where nodes closer to the gateway relay an increasing amount of aggregate traffic. Without a proportionate increase in the number of TXOPs, these nodes will experience higher queue drops. This results in capacity loss when the dropped packets originated from other nodes and had already consumed a portion of the shared spectrum. For example, consider a 2-hop parking lot topology with two flows originating from the 1-hop and the 2-hop node destined to a common gateway. Assume uniform wireless link rates with a nominal MAC-layer capacity W. The max-min fair share for each flow is $\frac{W}{3}$, for an aggregate network capacity of $\frac{2W}{3}$. However with 802.11 MAC and continuously backlogged sources, the aggregate network capacity reduces to $\frac{W}{2}$ with the 2-hop flow starving [17].

2.2 Nodes outside mutual carrier sense range

When two transmitters are outside carrier sense range, DCF's distributed scheduling driven by local carrier sensing may produce misaligned transmissions [11]. We use two illustrative topologies to show its impact on flow rate fairness: *information asymmetry* topology in Fig. 1a where S_1 experiences excessive packet loss due to collisions at R_1, and *flow-in-the-middle* topology in Fig. 1b where S_2 starves for TXOPs. In both cases, nodes develop a location-dependent, inconsistent view of the state of the shared wireless channel.

2.2.1 Starvation from collisions

Consider the topology in Fig. 1a where both senders S_1 and S_2 have backlogged traffic for their respective receivers R_1 and R_2. The two senders are outside mutual carrier sense range. Assume that both transmitters are in the first backoff stage, *i.e.*, they choose a random backoff between 0–31 time slots. A collision at R_1 is inevitable as the two transmissions can be at most 32 time slots (640 μs for 802.11b) apart, while it takes upwards of 1500 μs to transmit a 1500-byte Ethernet-friendly MTU and its subsequent link-level ACK using 802.11b physical layer parameters [33]. This collision only impacts S_1's packet to R_1. S_1 now doubles its MAC contention window, choosing a backoff between 0–63 time slots, while S_2 remains in the first backoff stage. S_2 is now twice likely to start transmitting before S_1; even if S_1 waits a maximum of its 64 time slots, the probability of collision is still 1. S_1 doubles its contention window yet again, but even in this third backoff stage, the probability of collision is 0.6. Thus, DCF steadily builds up the contention window for the disadvantaged node S_1, while allowing S_2 to contend for the channel with a minimum window following every successful transmission; the two transmitters share an inconsistent, asymmetric view of the channel state [11].



(a) Information asymmetry topology. R_1 is susceptible to collisions from S_2's transmissions. With backlogged traffic, S_1 starves while S_2's throughput equals the link capacity.

(b) Flow-in-the-middle topology. S_2 is in the carrier sense range of S_1 and S_3. When both S_1 and S_3 have backlogged traffic, S_2 starves for transmission opportunities.

Fig. 1: Topologies illustrating DCF performance limitations in multihop networks.

We note that the information asymmetry topology in Fig. 1a is an extension of the hidden terminal problem. However, floor-acquisition mechanisms such as RTS/CTS fail in this scenario. First, even the RTS frames are susceptible to a collision probability of 0.55 when both transmitters are in the first backoff. Second, when the RTS frames do not collide, R_1 will not respond to S_1's RTS if it has already been silenced by a prior RTS from S_2 to R_2. From S_1's perspective, this is no different from when its RTS frame collided at R_1 because of S_2's transmission.

2.2.2 Starvation from lack of transmission opportunities

Collisions are not the only reason for the nodes to develop an inconsistent view of the channel state; this may occur even in an ideal CSMA/CA protocol with no collisions. Consider the flow-in-the-middle [11] topology in Fig. 1b where S_2 is in carrier sense range of both S_1 and S_3, but S_1 and S_3 are outside carrier sense range of each other. With backlogged traffic sources, the throughput for S_1 and S_3 equals the channel capacity with S_2 starving. This is because S_2 is always deferring its transmissions to one of the other senders.

2.3 Cross-layer interaction with TCP

The DCF behavior described above may lead to cross-layer interaction with higher-layer protocols. In particular, TCP's congestion control mechanism further exacerbates the fairness problem. First, TCP allocates bandwidth as a function of a flow's round-trip time, thus penalizing flows with a large hop count. Second, TCP interprets delays in receiving an ACK as a sign of packet loss due to network congestion. In CSMA/CA networks, delays may occur due to transient medium access errors inherent in topologies similar to those described in Fig. 1. While wireless link-layer retransmissions may potentially recover from collisions, TCP retransmission timeouts may still occur in the interim. This results in TCP invoking slow start and dropping its congestion window to one. On the other hand, contending nodes that successfully transmitted a packet gradually increase their TCP congestion window under backlogged traffic, eventually capturing the wireless channel. Thus with backlogged TCP, the short-term MAC unfairness degenerates to long-term flow rate unfairness and starvation for disadvantaged flows.

3 Related Work

The challenges associated with using a CSMA/CA-based MAC in multihop networks have been discussed previously [19, 33]. In general, a flow not only contends with other flows sharing the spectrum (*i.e.*, inter-flow contention), but may even interfere with its own transmissions along the path to the destination (*i.e.*, intra-flow contention) [39]. Flows can be routed over non-interfering high-throughput paths [32, 24] when they exist; however, in many WMNs, the traffic is predominantly directed towards and away from the gateway, creating a network bottleneck. The degree of contention increases with increasing traffic loads. Related work in the literature addresses it from different perspectives: MAC-layer enhancements, transport layer enhancements, and higher-layer rate control algorithms.

By far the largest body of literature specifically devoted to wireless network fairness is that of the MAC-layer solutions (See [4, 26, 34], among others). Such approaches tend to assume that contending flows span a single hop and fairness may be achieved by converging the MAC contention windows to a common value. Schemes for reducing collisions can also help improve fairness, *e.g.*, Virtual Backoff Algorithm [18] use sequencing techniques to minimize the number of collisions in a single hop wireless network. However, optimal end-to-end fair allocation for multihop flows cannot be achieved by MAC scheduling based only on the local information. For multihop networks, solutions include prioritizing transmissions based on timestamps [27], using EDCA TXOPs differentiation [21], or adjusting the minimum contention window parameter at each relay node [2]. However, these solutions are not backwards-compatible across all variants of 802.11a/b/g/n networks, or may have limited utility in multi-radio, multi-channel WMNs. In this work we show that centralized flow rate control techniques are not constrained by these limitations.

A number of studies have associated the inter-flow contention experienced by a single TCP flow to its TCP congestion window exceeding its optimum size. For a chain topology, the optimum window size that maximizes spatial reuse is $1/4^{th}$ the number of hops between a source and destination [9]. Note that this does not resolve any inter-flow contention between multiple TCP flows, and subsequent unfairness and starvation may still ensue. Modifications or alternatives to TCP for multihop networks have also been proposed (*e.g.*, [6, 29]), though these require modifying the transport stack on the client devices and may present integration challenges when communicating with a wired host running the standard TCP stack. In this work, we show that strong fairness characteristics can be enforced through a centralized rate-based scheduling mechanism without modifying individual client devices or mesh routers.

Rate control algorithms operating outside the transport layer have also been shown to improve fairness between flows. Given a network topology and traffic demands, conflict graph models such as the clique model [26], its time-fairness extension [10], as well as Jun and Sichitiu's nominal capacity model [17], may be used to compute optimal bounds on network capacity. We defer description of these models to Appendix A. Raniwala *et al.* [30] proposed a distributed algorithm based on the conflict graph approach [13] for modeling constraints on simultaneous transmissions. Rangwala *et al.* [28] have proposed an AIMD-based rate control alongside a congestion sharing mechanism. The proposed mechanism is designed for many-to-one communication paradigm of sensor networks, but fails in one-to-many (typical downloads in mesh networks) or many-to-many scenarios, or a mix of these. This is because the congestion sharing mechanism as defined in IFRC, fails to propagate the congestion information to all potential interferers.

Finally, we note that several works deal with congestion control in wireless networks with relayed traffic [35,38,25]. However, congestion control does not aim for fairness, and the solutions do not guarantee any fairness scheme. Furthermore, their system models either focus on (very) bursty traffic and power-consumption for the sensor-net environment [35], or are incompatible with IEEE 802.11, assuming independence of links [38].

4 Centralized flow rate control in WMNs

Centralized rate control implemented at gateway routers offers many advantages over distributed rate control schemes. First, since the gateway bridges all traffic between WMN and the wired Internet, it can formulate a unified, up-to-date view of the traffic state without additional signaling. Second, the gateway rate control does not require any changes to mesh routers. This is advantageous when the mesh routers are commodity CPE (customer peripheral equipment) owned and managed by subscribers with the Internet Service Provider (ISP) having little control over them. Third, centralized rate control is effective even when the nodes in the network cannot be trusted to correctly enforce the desired rate control mechanisms. Finally, the notion of centralized rate control also lends itself naturally for providing an auditing and a billing framework that can be essential to the operations of an ISP.

Router-assisted congestion control mechanisms have been extensively studied for wired networks. Congestion in Internet routers occurs due to statistical multiplexing or link speed mismatch across different network interfaces. Gateway nodes in WMNs interface the high-speed wired backhaul link with the shared-spectrum wireless resource that is often the system bottleneck, creating opportunities for reusing existing wired solutions in this new problem domain. In the following sections, we consider three categories of algorithms: work-conserving scheduling-based algorithms (Sect. 4.1), preferential packet-drop algorithms (Sect. 4.2), and traffic-shaping algorithms (Sect. 4.3).

4.1 Work-conserving scheduling-based algorithms

Work-conserving packet scheduling algorithms like FQ and WFQ are approximations of the Generalized Processor Sharing (GPS) scheduler that is the theoretically ideal mechanism for providing fair bandwidth allocation [5]. Their work-conserving nature maintains a high network utilization. While distributed FQ protocols have earlier been proposed for ad hoc networks [23], we are interested in evaluating their impact on fairness when enforced at the gateway. To the best of our knowledge, this has not been evaluated in prior work.

4.2 Packet-drop/marking algorithms

Packet loss in wired networks primarily occurs at the router queue interface across the bottleneck link. Selective packet drop and/or marking techniques (*e.g.*, AQM variants such as Random Early Detection (RED) [8]) allow these routers to signal incipient network congestion to traffic sources. Since gateway mesh router bridges traffic between the high-speed wired network and the shared-spectrum wireless network, it appears that these algorithms may also be effective as gateway-enforced solutions in WMNs.

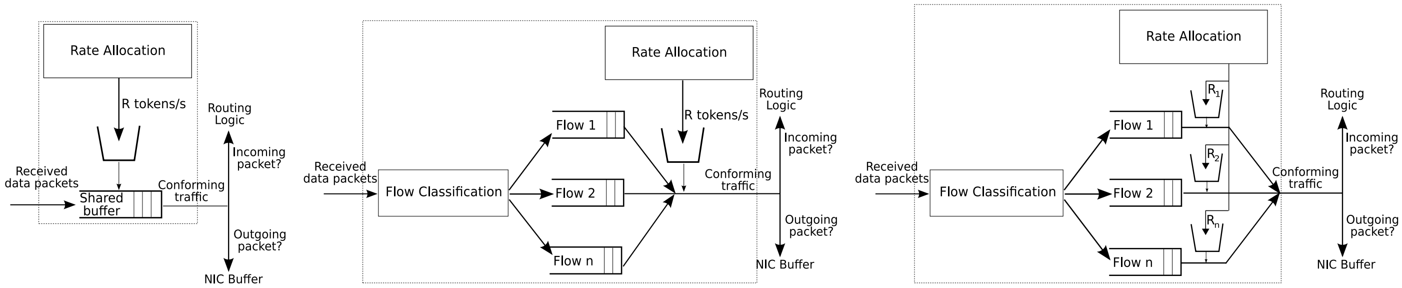
Fair Random Early Drop (FRED) [22] extends the RED algorithm to improve flow rate fairness. While RED effectively avoids network congestion, it may not improve fairness since it does not differentiate between connections. Thus when incipient congestion is detected, all packets (irrespective of the flow) are marked with the same drop probability. In contrast, FRED uses per-flow accounting to ensures that the drop rate for a flow depends on its buffer usage.

A brief overview of FRED is as follows: A FRED gateway classifies flows into logically separate buffers. For each flow i, it maintains the corresponding queue length $qlen_i$. It defines min_q and max_q, which respectively are the minimum and the maximum number of packets individual flows are allowed to queue. Similarly, it also maintains min_{th}, max_{th}, and avg for the overall queue. All new packet arrivals are accepted as long as avg is below the min_{th}. When avg lies between min_{th} and max_{th}, a new packet arrival is deterministically accepted only if the corresponding $qlen_i$ is less than min_q. Otherwise, as in RED, the packet is dropped with a probability that increases with increasing queue size.

We note that Xu *et al.* [37] have proposed the use of RED over a virtual distributed ‘neighborhood’ queue comprising nodes that contend for channel access. This was in the context of wireless ad hoc networks in which flows do not necessarily share traffic aggregation points. In our work we explore the traditional use of AQM as a router-assisted (gateway-enforced) mechanism.

4.3 Traffic policing/shaping algorithms

Traffic policing and shaping algorithms are commonly used when traffic limits are known or pre-determined in advance (*e.g.*, for enforcing compliance with a contract). The difference between policing and shaping is minor yet subtle: policing does not implement any queueing and excess packets are immediately dropped. Shaping, on the other hand, can absorb short bursts of packet, where the burst size is determined by the allocated buffer. When the buffer is full, all incoming packets



(a) Interface aggregate rate limit.

(b) Interface aggregate rate limit with FQ.

(c) Per-flow rate limit.

Fig. 2: Traffic shaping at gateway router. Fig. 2a limits the aggregate traffic allowed through the interface to rate R. All flows share a single FIFO buffer. Fig. 2b provides isolation between flows using per-flow queues and limits the aggregate traffic through the interface. Fig. 2c enforces per-flow rate limiting, with rate R_1 for Flow 1, R_2 for Flow 2, etc.

are immediately dropped and traffic shaping effectively acts as traffic policing. Both policing and shaping are examples of non-work-conserving scheduling methods.

Traffic shaping can be enforced at different levels of resource abstraction; it can be applied to aggregate traffic allowed to pass through a network interface, or it may be enforced on individual flows in a traffic stream. We describe some of these control configurations below.

4.3.1 Interface aggregate rate limit

The fundamental trade-off between total network capacity and flow-level fairness has been identified in prior work [10]. Specifically, aggregate network throughput is highest when all resources are allocated to the least cost flow while starving all others. Since the gateway router injects TCP packets or the subsequent ACKs into the wireless network, it can be used to control the aggregate throughput of a network. We are interested in enforcing a *fair-aggregate* rate limit at the gateway wireless interface. This is the fair-aggregate network capacity, and is simply the sum of max-min fair rate allocation of all flows in the network. This rate is then enforced on the net aggregate data traffic allowed through the gateway using the token bucket mechanism shown in Fig. 2a.

4.3.2 Interface aggregate rate limit with FQ

TCP flows sharing a single queue are susceptible to synchronization due to bursty and correlated packet losses. To prevent this, we introduce per-flow queues with fair scheduling between them. By separating flows, we can provide isolation between flows experiencing different levels of contention for network access, *e.g.*, we can separate locally generated traffic at a node from its relayed traffic. This new architecture is shown in Fig. 2b. Note that while flows are queued separately, rate limits are still enforced for the net aggregate traffic traversing the gateway.

Separating traffic into flows requires a flow classifier. For WMNs providing last mile access, this classification can be based on source or destination mesh routers. Thus a flow f_i represents the aggregate of all micro-flows originating from, or destined to, mesh router n_i in the network. In this context, we use nodes and flows interchangeably in our discussion. We note that this classification is consistent with the common practices employed by ISPs on wired networks, where capacity is managed on a per-subscriber basis.

4.3.3 Per-flow rate limit

While the architecture in Figs. 2a and 2b manage aggregate traffic through an interface, there may be a requirement for more fine-grained control over resource allocation between individual flows. This may be necessitated by QoS-enabled mesh networks where the provider wishes to support differentiated services, or provide weighted max-min or proportional fairness. We extend the system architecture to provide per-flow rate limiting at the gateway router as shown in Fig. 2c. Data traffic through the gateway can be classified into queues, which are then drained out at their specific rate. Note that we are proposing rate-limiting data traffic only; system housekeeping messages like routing updates are not rate limited.

5 Performance Analysis

We evaluate the efficacy of gateway-enforced control in WMNs using simulations in ns-2 [1]. We implement and evaluate each of the control actions described in Sect. 4 on the gateway. Our implementation works between the 802.11 MAC layer and the network layer, and operates transparently without requiring changes to either layer. We do not modify the regular mesh routers. We model the wireless channel propagation using the the two-ray ground reflection model [31]; it considers the direct path as well as the ground reflected path between the source and destination. We assume a static noise floor and a uniform static link rate of 1 Mb/s. We simulate the DCF channel access mechanism. Our TCP experiments simulate an infinite file transfer using TCP NewReno [7]. Our *upstream* flows originate from a mesh router and terminate at a host on the wired network; *downstream* flows take the other direction. We use Jain's Fairness Index (JFI) [14] as a quantitative measure of fairness for the resulting allocation.

5.1 Work-conserving scheduling-based algorithms

We simulate a TCP source on a wired network sending data to three mesh routers arranged in a 3-hop parking lot topology. Nodes are indexed such that nodes n_1, n_2, and n_3 are, respectively, $1, 2$, and 3-hops away from the gateway node n_0. Let the corresponding flows be f_1, f_2, and f_3. Nodes up to 2-hops may interfere per the default carrier sense and interference range values in our simulator. The wireless interface on the gateway n_0 implements FQ for downstream traffic. We benchmark these results against experiments with a shared Drop Tail FIFO queue at n_0.

Fig. 3 shows that FQ has little impact on flow rate fairness. TCP ACKs sent by n_3 are susceptible to collisions at receiver n_2 because of concurrent transmissions from n_0 which is outside n_3's carrier sense range. These collisions produce an inconsistent view of the channel state between the nodes; while n_3 backs off after repeated collisions, the TCP congestion window for flow f_1 builds up to fill the channel capacity. Smaller buffer size at n_0 limits the growth of this window, but when n_3 is backed up, any leftover capacity is consumed by the flow f_2.

FQ, WFQ, and similar router-assisted scheduling techniques assume independence between links and were designed as work-conserving schedulers; they do not allow the output link to remain idle if any of the flows have packets to send. While this maintains high efficiency in wired networks, it creates problems in wireless networks where the contending links are not independent, *i.e.*, transmission on a link precludes successful delivery of data on contending links. In topologies where mesh nodes share an inconsistent view of the channel state, work-conserving scheduler would schedule packets for advantaged node when it has nothing to send for distant, disadvantaged flows, while ideally it should defer any transmissions and keep the medium idle to allow for successful transmissions by disadvantaged nodes. The situation deteriorates when work-conserving schedulers are used with backlogged traffic sources using elastic TCP streams due to the cross-layer interaction described in Sect. 2.

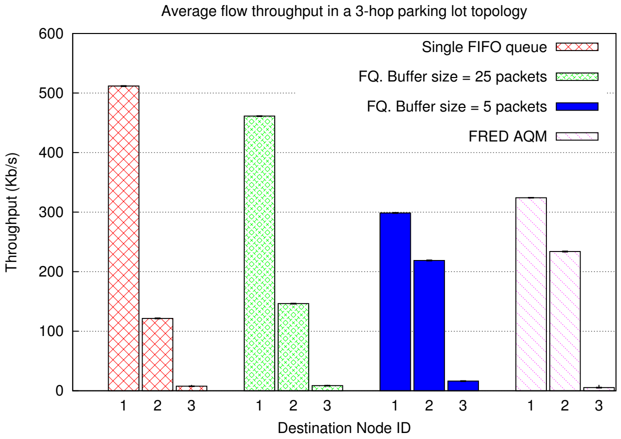


Fig. 3: Performance comparison of a shared FIFO queue, Single FIFO queue vs. per-flow queue at the gateway router for a 3-hop chain with download traffic.

5.2 Packet-drop/marking algorithms

We simulate a FRED gateway router on the 3-hop parking lot topology used above. We use downstream flows because a queue build-up (for detecting incipient congestion) only occurs when packets traverse from a high-speed wired link to a shared-medium WMN. The gateway queue size and various FRED parameters are consistent with the default values in ns-2.

Our results with the FRED queue at the gateway are shown in Fig. 3. It fails to prevent starvation for TCP flow to node n_3. By monitoring queue drops at the gateway, we found that FRED queue did register some proactive packet drops for f_1 and f_2, though it was insufficient to preclude the starvation of f_3.

Fig. 4 shows the per-flow data arrival rate (not ACKs) in the FRED queue at the gateway during our simulation. The queue space is evenly shared amongst the flows at the start, but continues deteriorating through the simulation execution. New data packets are not seen for f_3 because TCP ACKs for the previously transmitted ones are never received. This is because TCP ACKs transmitted by n_3 experience a high loss rate due to collisions from concurrent transmissions by n_0.

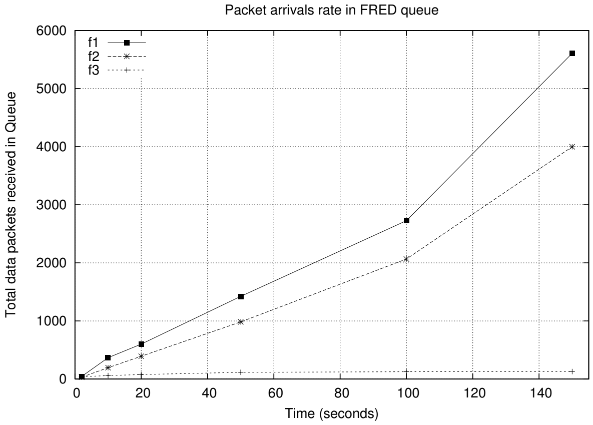


Fig. 4: New data packet arrival rate in FRED queue.

As discussed in Sect. 2, this hidden terminal cannot be resolved using RTS/CTS control frames. Because of frequent collisions, n_3 repeatedly increases its contention window to a point where TCP timeouts occur, and the packets have to be retransmitted by the gateway. Though f_1 transmits fewer packets with FRED, the extra available bandwidth is acquired by f_2 because there is very little traffic to be sent out for f_3 because of the combined effect of the 802.11 contention window and the TCP congestion window.

We conclude that AQM is ineffective as a gateway-enforced technique for improving flow rate fairness in WMNs. This is due to fundamental differences in packet loss characteristics between wired networks and WMNs [16]. In wired networks, packet loss occurs primarily at the queue interface into the bottleneck link. In WMNs, however, these packet losses are spatially distributed over various intermediate routers (see Sect. 2) and cannot be accurately predicted by simply monitoring the queue size at the gateway router.

5.3 Traffic policing/shaping algorithms

We evaluate the various traffic shaping alternatives described in Sect. 4.3. Our simulations include a number of chains, grids, and random multihop network topologies, including both upstream and downstream flows, with up to a maximum of 35 simultaneously active nodes transmitting via a single gateway. Experiments for a given topology are repeated 25 times with different random seeds and random flow activation sequences, and the results averaged. For each topology, the traffic shaping rate is computed off-hand using a *collision domain* network capacity model [17]. Other capacity models such as *clique-based* models [13] may similarly be used. We note that the mechanisms for computing fair flow rates is orthogonal to this work; in this paper, we focus on evaluating the efficacy of gateway-enforced control *given a desired rate allocation*, rather than the mechanics of accurately estimating the network capacity and inferring flow rates. Nonetheless, for completeness, in A we provide an overview of these two capacity models.

The fair rate allocation computed by the model is enforced at the gateway via the traffic shaping architectures described in Sect. 4.3. The collision domain capacity model allows us to compute per-flow rate. The interface aggregate rate limit is then simply the sum of the fair rates of constituent flows. This rate limit is the fair-aggregate capacity of the network.

5.3.1 Long-lived elastic TCP flows

We first evaluate the performance characteristics of long-lived TCP flows whose congestion control phase is significant longer than their slow start phase such that the impact of the slow start phase can be ignored. Our results are summarized in Tables 1 and 2 for downstream and upstream flows respectively. In addition to JFI, we also list $\frac{avg.min.flow\,rate}{fair\,rate}$ and $\frac{avg.max.flow\,rate}{fair\,rate}$ to illustrate the imbalance between the minimum and maximum throughput flows. To quantify spatial reuse, we define *effective network utilization* [40] $U = \sum_{i \in N} r_i \times l_i$, where r_i is the measured throughput for flow f_i, and l_i is the number of hops between the source and destination on the routing path of f_i. We list the value of $\frac{U}{U_{opt}}$, where U_{opt} is the network utilization achieved by the computational model described in A.

We benchmark our results as follows:

1. We perform the same set of experiments using a single, FIFO Drop Tail queue at the gateway router.
2. We repeat these experiments using FQ at the gateway router with a per-flow buffer size of 5 packets. Our prior work [15] shows that this buffer size maintains low queueing delays at the gateway with little loss in end-to-end flow rate.
3. For upstream flows, we perform additional experiments that source rate limit the flows to their computed fair share rate without any modifications on the gateway router. For downstream flows, this source rate limit is akin to per-flow gateway rate limit as the gateway is now injecting packets in the wireless medium.

Our results in Table 1 and 2 show that simply enforcing rate-based scheduling, even on the granularity of aggregate amount of traffic allowed through the network interface, provides upwards of two-fold improvements in JFI compared to the base case with a shared

Scheme	**JFI**		$\frac{min.\,rate}{fair\,rate}$		$\frac{max.\,rate}{fair\,rate}$		$\frac{U}{U_{opt}}$	
	Avg.	Std.Dev.	Avg.	Std.Dev.	Avg.	Std.Dev.	Avg.	Std.Dev.
Single FIFO queue	0.41	0.24	0.10	0.10	10.86	7.08	0.95	0.09
FQ	0.55	0.20	0.20	0.06	5.36	2.95	1.00	0.07
Aggregate rate limit	0.81	0.15	0.43	0.30	2.02	0.86	0.91	0.11
Aggregate rate limit & FQ	0.98	0.02	0.72	0.31	1.08	0.08	0.98	0.03
Per-flow rate limit	0.99	0.02	0.75	0.26	1.01	0.01	0.95	0.07

Table 1: Fairness indices for downstream TCP flows.

Scheme	**JFI**		$\frac{min.\,rate}{fair\,rate}$		$\frac{max.\,rate}{fair\,rate}$		$\frac{U}{U_{opt}}$	
	Avg.	Std.Dev.	Avg.	Std.Dev.	Avg.	Std.Dev.	Avg.	Std.Dev.
Single FIFO queue	0.31	0.13	0.17	0.16	15.2	11.6	0.95	0.09
FQ	0.32	0.13	0.16	0.15	14.04	0.61	1.00	0.06
Aggregate rate limit	0.79	0.16	0.43	0.34	2.23	0.91	0.99	0.07
Aggregate rate limit & FQ	0.91	0.11	0.59	0.40	1.38	0.37	1.00	0.03
Per-flow rate limit	0.99	0.02	0.76	0.30	1.00	0.06	0.90	0.06
Source rate limit	0.99	0.01	0.77	0.21	1.00	0.01	0.91	0.07

Table 2: Fairness indices for upstream TCP flows.

FIFO queue. We note that rate-based scheduling enforced via traffic shaping is, by nature, non-work-conserving. Thus while underlying topologies may still be susceptible to 802.11 MAC limitations described in Sect. 2, link-layer retransmissions can provide reliable packet delivery as long as non-work-conserving, rate-based scheduling can shield individual flows from effects of cross-layer interaction with TCP.

FQ by itself only provides a marginal improvement in fairness over FIFO Drop Tail queues. However, when FQ is combined with non-work-conserving rate-based scheduling, we see a further improvement of about 15%–20% over interface rate limiting alone. FQ introduces isolation between flows, protecting one flow's traffic from that of another. This leads to better short-term fairness that translates to improved long-term fairness calculated over average flow rates. We highlight this for a 5-hop, 4-flow parking lot topology in Fig. 5. The buffer size at the gateway was 5 packets in experiments with per-flow buffering, 25 packets otherwise. With 1 Mb/s wireless links, max-min fair share per-flow is approximately 65 Kb/s. Simply providing flow isolation using FQ without any rate limiting does not solve the fairness problem. In Fig. 5b, the work-conserving FQ allows TCP congestion window size to grow to large values even with small buffer sizes at the gateway. Interface aggregate rate limiting improves fairness in Fig. 5c, though some flows still experience short-term unfairness at instances when other aggressive flows have built up a large TCP congestion window. This happens because all flows share the same buffer at the gateway. It is the combination of FQ and aggregate rate limiting that improves short-term fairness between flows. TCP congestion window sizes are now bounded as shown in Fig. 5f, thus considerably cutting down the jitter between packets from different flows. Per-flow rate limiting provides similar qualitative results as it also allocates separate buffers at the gateway.

The quantitative analysis of per-flow rate control in Tables 1 and 2 show a further improvement in fairness index of about 1%–8% over FQ with interface aggregate rate limiting. We note that these fairness characteristics of per-flow rate-limiting are very similar to those achieved with source rate limiting. Incidentally, perfect fairness cannot be achieved even with source rate limiting. Some network topologies may exhibit inherent *structural unfairness* [20], requiring control action beyond simple rate limiting. Addressing this is beyond the scope of this work.

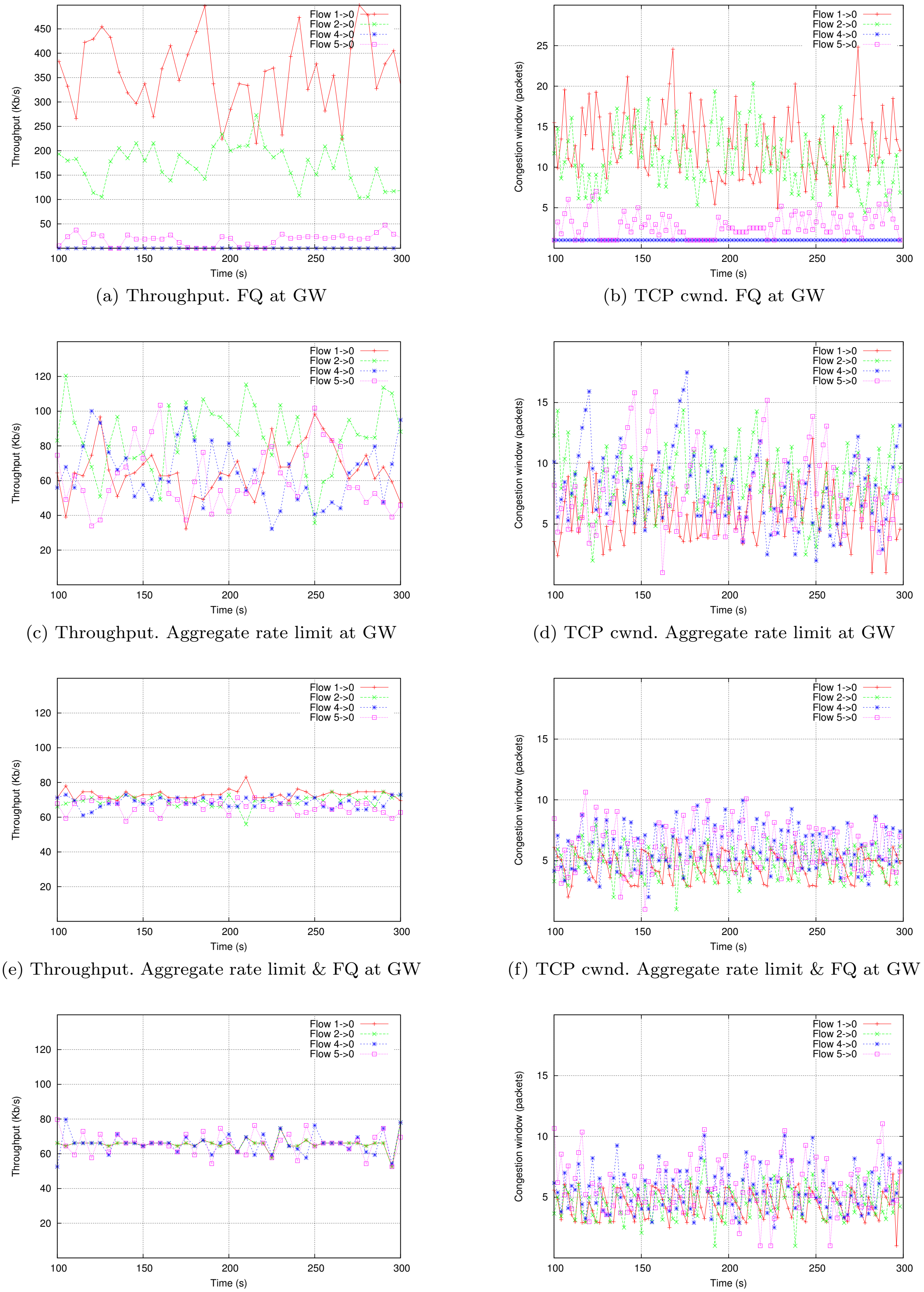
Finally, we note that normalized effective network utilization is upwards of 90% for all scheduling techniques for both downstream and upstream flows; backlogged TCP flows saturate the spectrum around the gateway in all cases, irrespective of fairness in rate allocation between individual flows.

In summary, our experiments show that centralized rate control cannot be exercised in WMNs using work-conserving scheduling techniques. Using non-work-conserving, rate-based scheduling is equally effective as source rate limiting techniques that require changing the MAC or transport layer on end-hosts.

5.3.2 Multiple long-lived flows per node

We now evaluate the efficacy of gateway rate control when multiple flows originate from a mesh router. Consider a 20-node topology with a random node placement. We randomly select 10 of these nodes as traffic sources. Each source generates between 1 to 3 upload flows. The fairness criterion we target is per-subscriber fairness irrespective of their flow count, where each subscriber corresponds to a mesh node. As discussed earlier, this resource allocation policy is consistent with the practices employed by ISPs on wired networks.

Our results are shown in Fig. 6. Flows from a given node are grouped together and can be distinguished by the node ID. We normalize the measured flow throughput to the fair share rate computed with the collision domain capacity model. Node 11 has the highest standard deviation amongst its three flow rates. The sum of the flow rates for each node, however, remains bounded within the fair allocation constraints for this network. Equal allocation of a node's share of network capacity between its sub-flows needs to be managed in a wireless network insofar as it needs to be enforced in a wired network. Studying this aspect is beyond the scope of this current work.



(a) Throughput. FQ at GW

(b) TCP cwnd. FQ at GW

(c) Throughput. Aggregate rate limit at GW

(d) TCP cwnd. Aggregate rate limit at GW

(e) Throughput. Aggregate rate limit & FQ at GW

(f) TCP cwnd. Aggregate rate limit & FQ at GW

(g) Throughput. Per-flow rate limit at GW

(h) TCP cwnd. Per-flow rate limit at GW

Fig. 5: Flow throughput and TCP cwnd for a 5-hop, 4-flow chain. Max-min rate per flow is approx. 65 Kb/s. In Fig. 5c and 5d, the GW has a single FIFO buffer of 25 packets. In all other cases, it has a per-flow buffer of size 5 packets.

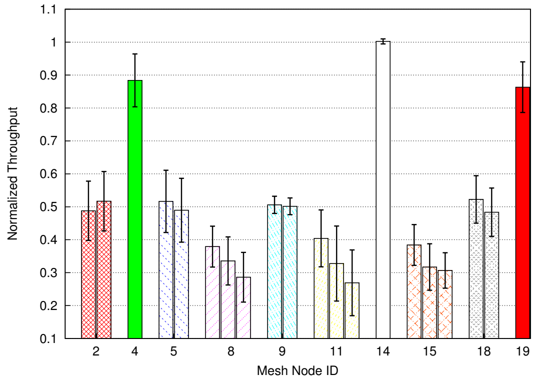


Fig. 6: Gateway-enforced rate control for multiple upload streams from a node. 10 randomly selected nodes from a 20-node random topology generate between 1 to 3 flows each. Flow throughput is normalized to the fair share rate of the node. Error bars are the 95% confidence intervals.

5.3.3 Short-lived elastic TCP flows

We next consider the performance characteristics of short-lived dynamic flows where we do not ignore the impact of slow start but use it to evaluate how quickly new flows converge to their fair share allocation. Similarly, when an existing flow terminates, we are interested in evaluating how quickly the freed resources can be utilized by other flows.

Flow activation/termination can be detected in multiple ways. TCP stream activation and tear-down can be detected by the exchange of the TCP-specific three-way handshake messages. In our case where a flow bundle constitutes multiple TCP streams, we simply use the presence or absence of packets to determine the current state of stream activity, thus obviating any overhead associated with the distribution of stream activity information. On detecting a new stream, the centralized controller simply computes a new rate per active stream and starts enforcing it. Detecting stream deactivation can be a little tricky; our controller waits for a time interval during which no packet is received from a flow. This time interval should be a function of the average delay and jitter experienced by a flow.

We consider the results of a 7-hop chain with nodes indexed $n_0, n_1, ...n_7$, with n_0 being the gateway router. Only the neighboring nodes can directly communicate while nodes up to 2 hops away may interfere. Initially, five flows are active. Flows 1→0 and 0→5 are terminated at time 150 s, while flow 0→7 is terminated at 200 s. Finally, flows 1→0 and 0→7 are reactivated at 250 s. Measured flow rates with per-flow rate limiting at the gateway are shown in Fig. 7. We are particularly interested in the convergence time required for flows to converge around their new fair rates. We note that this convergence time is a function of the TCP state. A TCP agent starts up in slow start, where its congestion window builds up exponentially over time. This allows flows 1→0 and 0→7 to rapidly approach their fair rate within the 5 s resolution of our plot. However, rate increases for flows in congestion avoidance mode takes longer as the congestion window can only increase linearly in time. Consequently, flows 3→0 and 0→6 take up to 15 s. to stabilize at their new fair rates at 215 s.

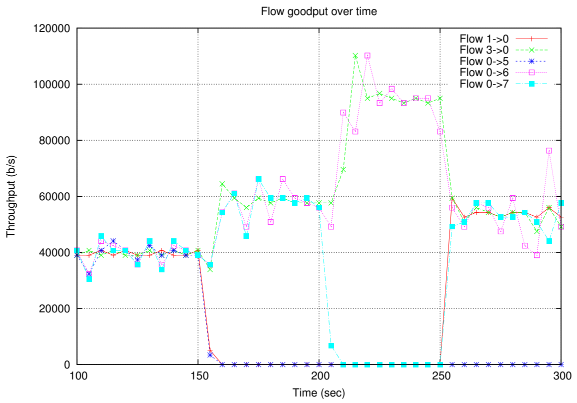


Fig. 7: Throughput vs. time for a 7-hop chain with per-flow rate limiting at the gateway. 1→0 and 0→5 are turned off at 150 s, while flow 0→7 is turned off at 200 s. Flows 1→0 and 0→7 come back at 250 s.

5.3.4 Non-adaptive flows

Router-assisted rate control mechanisms are targeted at adaptive transport protocols that can react to congestion notification. TCP is the canonical example of such an adaptive protocol and constitutes the bulk of Internet traffic [36]. However, UDP-based communications are increasingly being used for real-time delivery of audio and video data. We now evaluate the performance of gateway-assisted centralized rate control for such non-adaptive flows.

We simulate a 3-hop parking lot topology, with nodes indexed n_0, n_1, n_2, n_3 with n_0 being the gateway router. Our UDP constant bit rate (CBR) application generates 500-byte payload at 5 ms. intervals, for a total load of 800 Kb/s. We considered both upstream and downstream flows, with a UDP stream originating/terminating per mesh node respectively. With 1 Mb/s wireless links, the max-min fair rate is approximately 125 Kb/s. Thus our UDP sources generate traffic load that is higher than the fair share per-flow, but is still low enough to prevent complete channel capture by any single flow. Our results with per-flow rate limiting at the gateway are shown in Table 3.

Traffic Type	Flow	$\frac{avg.rate}{fair.rate}$	JFI
Upstream UDP	1→0	1.00	0.54
	2→0	0.31	
	3→0	0.02	
Downstream UDP	0→1	1.00	1.00
	0→2	1.00	
	0→3	1.01	

Table 3: Per-flow rate control for upstream and downstream UDP flows in a 3-hop parking lot topology with gateway node n_0.

We observe that gateway-assisted rate control in WMNs can successfully contain downstream UDP flows only. In this case, it effectively acts as source rate control, limiting each stream to its fair share on the wireless network. However, upstream flows continue experiencing unfairness; while we can limit the goodput of f_1 to its fair share by dropping its excess traffic at the gateway, its non-adaptive transport protocol still sources traffic at 800 Kb/s. The locally generated traffic at n_1 shares the same transmit buffer as the relayed traffic from n_2 and n_3. With a probability that increases with the offered load, the relayed packets are likely to find this buffer full and will be dropped in the Drop Tail buffers [17]. Thus the relayed traffic from f_2 and f_3 experiences a high loss rate and the resulting flow rate unfairness.

Additional mechanisms beyond gateway-enforced traffic policing/shaping algorithms are required to adapt the rate of non-congestion controlled upstream flows, *e.g.*, rate information calculated at the gateway may be communicated back to the source nodes for enforcement at the traffic ingress points. Mesh routers need to correctly interpret and enforce these rate limits. We defer this study to future work.

5.3.5 Multi-radio, multi-channel WMNs

We have further validated the efficacy of gateway control in multi-radio, multi-channel WMNs. We extended ns-2 to support two radio interfaces per mobile node. Each interface is assigned a static, non-overlapping channel so as to maintain connectivity between neighboring nodes. Channel assignment for optimal network performance is beyond the scope of our current work.

We consider the topology with three non-overlapping channels in Figure 8. Node 0 is the gateway mesh router. Flows 1 → 0, 2 → 0, 5 → 0, and 6 → 0 are activated at time 100 s., while flows 3 → 0 and 4 → 0 are activated at time 200 s. and 300 s., respectively. Finally at time 400 s. flows 2 → 0, 4 → 0, and 5 → 0 are terminated. Figure 9 shows the measured throughput averaged over 5 s. intervals. Table 4 shows the measured throughput of a flow normalized to its computed fair share over various intervals.

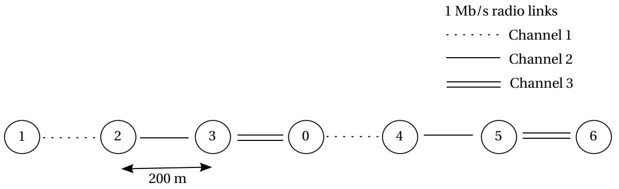


Fig. 8: Multi-radio, multi-channel network. Node 0 is the mesh gateway.

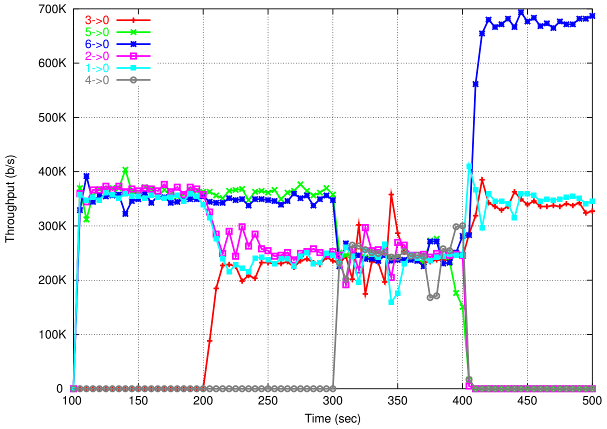


Fig. 9: Throughput vs. time for the topology in Fig. 8 with per-flow rate limiting at the gateway.

Table 4: Flow rate averages for the results in Fig. 9.

Interval (s)	$\frac{throughput}{computed\ fair\ share}$					
	1→0	2→0	3→0	4→0	5→0	6→0
100-200	0.96	0.99			0.98	0.95
200-300	0.98	1.06	0.88		0.97	0.94
300-400	0.95	0.99	0.99	0.99	0.96	0.99
400-500	0.95		0.91			0.88

We observe that fairness improves considerably such that overlapping flows rates over various intervals are often indistinguisable. Of particular interest are the 200-300 s. and the 400-500 s. intervals where the active flows sourced from nodes on either side of the gateway do not share a common bottleneck, leading to max-min fairness with unequal flow rates. Using per-flow rate limiting at the gateway, we can correctly converge the flows to their fair share of network capacity.

6 Conclusions and Future Work

WMNs, particularly those based on the 802.11 radios, exhibit extreme fairness problems, requiring existing deployments to limit the maximum number of hops to the gateway to prevent distant nodes from starving. In this paper we explore the feasibility of using centralized rate control that can be enforced at traffic aggregation points such as gateway routers. We show that router-assisted techniques in wired networks, including work-conserving packet scheduling (such as FQ and its variants) and probabilistic packet-drop techniques (such as AQM and its variants) are inadequate as centralized rate control techniques in WMNs. This is because of fundamental differences in the abstraction of wired and wireless networks: (1) transmissions on wired links can be scheduled independently, and (2) packet losses in wired networks occur only as queue drops at bottleneck routers. Our experiments indicate that non-work-conserving rate-based centralized scheduling can be used effectively in WMNs. Even rate-limiting the aggregate traffic passing through the gateway router improves the fairness index two-folds over the base case with a shared FIFO queue. Further granularity in rate allocation control can be obtained by isolating flows using per-flow buffering and by exercising per-flow rate limiting. The fairness indices achieved with these modifications are comparable to source rate limiting techniques that require changing the MAC or transport layer on the end-hosts.

Having established the feasibility of gateway-assisted rate control in WMNs, we are now working on extending this work along multiple dimensions. First, we are developing practical heuristics and mechanisms to estimate flow rates using the information available locally at the gateway. We are pursuing a feedback-based approach in which the centralized controller adapts its behavior in response to changing network and flow conditions. Second, we are considering the impact of multiple gateway nodes in large WMN deployments. Some gateways in these networks may need to exchange signaling information to reconcile their views of the available network capacity. This requires identifying flows that use one gateway but interfere with flows using other gateway(s). The signaling between these gateways, however, may use the wired backbone without consuming wireless capacity. We hope to address these challenges in the future.

References

1. The network simulator - ns-2. `http://www.isi.edu/nsnam/ns`.
2. A. Aziz, D. Starobinski, P. Thiran, and A. E. Fawal. EZ-Flow: removing turbulence in IEEE 802.11 wireless mesh networks without message passing. In *Proc. of the ACM CoNEXT '09*, pages 73–84, 2009.
3. D. P. Bertsekas and R. G. Gallager. *Data networks*. Prentice-Hall, 1992.
4. V. Bharhgavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LANs. In *Proc. of the ACM SIGCOMM '94*, pages 249–256, September 1994.
5. A. J. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. of the ACM SIGCOMM '89*, pages 1–12, September 1989.
6. S. M. ElRakabawy and C. Lindemann. A practical adaptive pacing scheme for TCP in multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(4):975–988, Aug. 2011.
7. S. Floyd, T. Henderson, and A. Gurtov. The NewReno modification to TCP's fast recovery algorithm. RFC 3782, Internet Engineering Task Force, Apr. 2004.
8. S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
9. Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP performance. *IEEE Transactions on Mobile Computing*, 4(2):209–221, March/April 2005.
10. V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *Proc. of the ACM MobiCom '04*, pages 287–301, September 2004.
11. M. Garetto, T. Salonidis, and E. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. In *Proc. of the IEEE INFOCOM '06*, April 2006.
12. R. Gupta and J. Walrand. Approximating maximal cliques in ad-hoc networks. In *Proc. of the IEEE PIMRC '04*, pages 365–369, September 2004.
13. K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. of the ACM MobiCom '03*, pages 66–80, September 2003.
14. R. Jain, D.-M. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer system. Technical report, DEC Research Report TR-301, September 1984.
15. K. Jamshaid, L. Li, and P. A. Ward. Gateway rate control of wireless mesh networks. In *Proc. of the WiMeshNets*, August 2006.
16. K. Jamshaid and P. A. Ward. Experiences using gateway-enforced rate-limiting techniques in wireless mesh networks. In *Proc. of the IEEE WCNC '07*, pages 3725–3730, March 2007.
17. J. Jun and M. L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, pages 8–14, October 2003.
18. P. Krishna, S. Misra, M. S. Obaidat, and V. Saritha. Virtual backoff algorithm: An enhancement to 802.11 medium-access control to improve the performance of wireless networks. *IEEE Transactions on Vehicular Technology*, 59(3):1068–1075, 2010.
19. J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. of the ACM MobiCom '01*, pages 61–69, July 2001.
20. L. Li and P. A. Ward. Structural unfairness in 802.11-based wireless mesh networks. In *Proc. of the IEEE CNSR '07*, pages 213–220, May 2007.

21. T. Li, D. J. Leith, V. Badarla, D. Malone, and Q. Cao. Achieving end-to-end fairness in 802.11e based wireless multi-hop mesh networks without coordination. *Mobile Networks and Applications*, 16(1):17–34, Feb. 2011.
22. D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proc. of the ACM SIGCOMM '97*, October 1997.
23. H. Luo, J. Cheng, and S. Lu. Self-coordinating localized fair queueing in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 3(1):86–98, January 2004.
24. S. Misra, T. I. Ghosh, and M. S. Obaidat. Routing bandwidth guaranteed paths for traffic engineering in WiMAX mesh networks. *Wiley International Journal of Communication Systems*, 2013.
25. S. Misra, V. Tiwari, and M. S. Obaidat. Lacas: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 27(4):466–479, 2009.
26. T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC layer fairness in wireless packet networks. In *Proc. of the ACM MobiCom '00*, pages 87–98, August 2000.
27. F. Nawab, K. Jamshaid, B. Shihada, and P.-H. Ho. TMAC: Timestamp-ordered MAC for CSMA/CA Wireless Mesh Networks. In *Proc. of the IEEE ICCCN '11*, August 2011.
28. S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *Proc. of the ACM SIGCOMM '06*, pages 63–74, September 2006.
29. S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan. Neighborhood-centric congestion control for multi-hop wireless mesh networks. *IEEE/ACM Transactions on Networking*, 19(6):1797–1810, Dec. 2011.
30. A. Raniwala, P. De, S. Sharma, R. Krishnan, and T. Chiueh. End-to-end flow fairness over IEEE 802.11-based wireless mesh networks. In *Proc. of the IEEE INFOCOM Mini-Conference*, pages 2361–2365, May 2007.
31. T. S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 2nd edition, 2002.
32. T. Salonidis, M. Garetto, A. Saha, and E. Knightly. Identifying high throughput paths in 802.11 mesh networks: a model-based approach. In *Proc. of the ICNP '07*, pages 21–30, Oct. 2007.
33. J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. Knightly. Measurement and modeling of the origins of starvation in congestion controlled mesh networks. In *Proc. of the IEEE INFOCOM '08*, 2008.
34. L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless networks. In *Proc. of the IEEE INFOCOM '02*, June 2002.
35. C.-Y. Wan, S. Eisenman, and A. Campbell. CODA: Congestion Detection and Avoidance in sensor networks. In *Proc. of the ACM SenSys '03*, pages 266–279, 2003.
36. C. Williamson. Internet traffic measurement. *IEEE Internet Computing*, 5(6):70–74, Nov/Dec. 2001.
37. K. Xu, M. Gerla, L. Qi, and Y. Shu. TCP unfairness in ad hoc wireless networks and a neighborhood RED solution. *Wireless Networks*, 11(4):383 – 399, July 2005.
38. Y. Yi and S. Shakkottai. Hop-by-hop congestion control over a wireless multi-hop network. In *Proc. of the IEEE INFOCOM '04*, 2004.
39. H. Zhai and Y. Fang. Distributed flow control and medium access in multihop ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(11):1503–1514, November 2006.
40. L. Zhang, S. Chen, and Y. Jian. Achieving global end-to-end maxmin in multihop wireless networks. In *Proc. of the IEEE ICDCS '08*, pages 225–232, August 2008.

A Model for estimating per-flow fair-share

Flow rates used in our analysis in Sect. 5 were computed offline using a network capacity model. For completeness, in this appendix we briefly describe the two computational models that we considered and provide an analysis of the capacity achievable with these models to the model implemented in ns-2.

We first state the assumptions necessary to our approach. We presume that routing is relatively static, based on the fact that the WMN nodes are stationary, and likely quite reliable. By "relatively static" we mean that changes in routing will be significantly fewer than the changes in stream activity. This assumption implies a few things, including that network membership changes (such as node additions or hardware failures) are few and far between, and that load balancing is not used in the network. While the first assumption is certainly valid, the second assumption is a simplification that we hope to address in the near future.

We also assume that the WMN has a single gateway. Though this is generally not true in large deployments, given static routing, for each node there will be a single gateway. We thus partition a multi-gateway WMN into disjoint WMNs, each with a single gateway. While there may be interference between the resulting set of WMNs, this is a problem that must already be dealt with insofar as there may be interference from any number of other sources.

Given these assumptions, we consider a WMN with N nodes that are arbitrarily located in a plane. Let d_{ij} denote the distance between nodes n_i and n_j. Let T_i be the transmission range of node n_i. We model this network as a labeled graph, where the mesh nodes are the vertices, and a labeled edge exists between two vertices n_i and n_j iff

$$(d_{ij} \leq T_i) \wedge (d_{ij} \leq T_j)$$

In other words, the nodes must be within transmission range of each other. An edge in this connectivity graph is also referred to as a *link*. A *stream* is defined by an exchange of data packets between a mesh node and its corresponding gateway. An *active stream* is one for which data is currently being exchanged.

A.1 Computational Model

The fair-share computation model is an optimization problem subject to the feasibility model for the network, the network state, and the fairness criterion adopted.

The feasibility model reflects the throughput constraints imposed by the network. It consists of a set of constraints determined by how streams use the links, and then how these links contend for the wireless channel. The former is a function of the routing protocol; for the latter, we describe two variations (bottleneck clique vs. collision domain) in the following section below.

This feasibility model is extended by the network state, which is simply the desired rate, $G(s)$, for each stream, s. For this paper we consider only binary activity: the stream is either silent ($G(s) = 0$) or never satisfied ($G(s) = \infty$). This corresponds to TCP behavior, which either is not transmitting or will increase its transmission rate to the available bandwidth. We are incorporating flows with fixed bandwidth requirements in the future.

Finally, the fairness criterion implements the selected fairness model. In this paper we deliberately restrict our analysis to max-min fairness (*i.e.*, active streams receive as much throughput as the network can offer without causing other active streams with a lesser throughput to suffer), so as to focus on the accuracy of the model for 802.11-based WMNs and the efficacy of the gateway as a control point. However, we note that the computation model can be extended to any feasible, mathematically-tractable fairness criterion that can be expressed as a set of rate allocation constraints.

A.2 Network Feasibility Models

We now describe the details of the two network feasibility models. Both models start by dividing the problem into one of link constraints (*i.e.*, usage of links by streams) and medium constraints (*i.e.*, usage of the medium by links). The former is the same for both models, as it is a function of the routing together with the demands placed on the network. The latter is where the two models differ.

A.2.1 Link-Resource Constraints

Let $R(s)$ be the rate of stream s, and $C(l)$ be the maximum allowed aggregate throughput that link l can carry. For each link l the link resource constraint is specified as:

$$\sum_{i:s_i \text{ uses } l} R(s_i) \;\le\; C(l) \tag{1}$$

Since a stream uses all the links on its route, the above usage information can be inferred directly from the routing information. This usage information can be encoded in a 0-1 link-usage matrix $\mathbf{L}$ as follows:

$$\mathbf{L}[i,j] = \begin{cases} 1 & \text{when stream } s_j \text{ uses link } l_i \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathbf{C}$ be the link-capacity vector, where $\mathbf{C}[j] = C(l_j)$. Also let $\mathbf{R}$ be the stream throughput vector, where $\mathbf{R}[i] = R(s_i)$. Then the stream-link usage constraint can be expressed as:

$$\mathbf{LR} \le \mathbf{C} \tag{2}$$

$$\mathbf{R} \ge 0 \tag{3}$$

A.2.2 Medium-Resource Constraints

The basic problem in developing medium-resource constraints is that contention is location-dependent, with the medium conceptually divided into overlapping resources of limited capacity. The clique model computes mutually incompatible sets of links, all but one of which must be silent at any given time for collision-free transmission. The collision-domain model considers the medium-resource unit to be the link, and determines the set of links that must be silent for a given link to be used. We first formalize the clique model.

Clique Model of Medium-Resource Constraints In the clique model, two links *contend* if they cannot be used simultaneously for transmission of packets. Link contention is captured by a set of link-contention graphs $G = (V, E)$, where V is the set of all links, and $\{u, v\} \in E$ iff links u and v contend. Define $B(u)$ to be the available bandwidth in each such distinct region u (*i.e.*, in each clique). Since all links in a clique contend with each other, only one link in the clique can be active at any instant. We can thus define the medium-resource constraints of the clique model as:

$$\sum_{i:i \text{ in clique } u} C(l_i) \;\le\; B(u) \tag{4}$$

Note that if each wireless router transmits at the same rate, the value of $B(u)$ can be reasonably approximated as the throughput that can be achieved at the MAC layer in a one-hop network with infrastructure. If routers transmit at different rates, a weighted contention graph may be used.

The resulting set of medium-resource constraints can be written down as matrix equation. First, define the 0-1 medium-usage matrix $\mathbf{M}$ as:

$$\forall i, j \; \mathbf{M}[i,j] = \begin{cases} 1 & \text{when link } l_j \in \text{clique } u_i \\ 0 & \text{otherwise} \end{cases}$$

Let the medium-capacity vector be $\mathbf{B}$, where $\mathbf{B}[i] = B(u_i)$. The medium-resource constraint is then:

$$\mathbf{MC} \le \mathbf{B} \tag{5}$$

The clique model requires the (NP-complete) computation of cliques within the contention graph and, as a more practical matter, the determination of which links contend. While the former problem is, to some degree, amenable to careful analysis potentially enabling more-efficient computation [12], the latter problem is extremely difficult to deal with. Specifically, determining which links interfere with which other links in a wireless mesh network is not, in general, feasible, in part because interference is experienced by a receiver, not by a link, and thus depends on traffic direction.

Collision-Domain Model of Medium-Resource Constraints We therefore examine the efficacy of a simpler model of collision domains [17]. This model both reduces the computation requirements as well as being practically determinable. In this model, two links contend if one endpoint of a link is within transmission range of an endpoint of the other link. The collision domain of link l_i is defined as the set of all links that contend with link l_i. Note that this is equivalent to the set of all vertices adjacent to vertex l_i in the link-contention graph, modulo the definition of “contend”. In this case we define $B(u)$ as the available bandwidth in each collision domain. In single-rate routers this will be the same value as that in the clique model. The medium-resource constraints for the collision-domain model are then:

$$\sum_{i:l_i \text{ in } u} C(l_i) \;\le\; B(u) \tag{6}$$

Note that since transmission range is often much less than interference range, this model underestimates the level of contention. However, each collision domain will, in general, contain links that do not contend with each other, thus overestimating the number of contending links compared to the more-accurate cliques. As a result the combined model has the potential of offering acceptable accuracy, with computational simplicity and practical feasibility. We must emphasize that in this model it is possible for nodes within the WMN to identify the set of contending links, which is difficult, if not infeasible, with the clique model.

As with the clique model, we can define a 0-1 medium-usage matrix $\mathbf{M}$ as follows:

$$\forall i, j \; \mathbf{M}[i,j] = \begin{cases} 1 & \text{when link } l_j \in \text{collision domain } u_i \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the medium-capacity vector **B** can be redefined as $\mathbf{B}[i] = B(u_i)$, where $B(u_i)$ is the available bandwidth of collision domain u_i. Equation 5 then remains unaltered, though using the collision-domain definitions of **M** and **B**.

In both cases, the network feasibility model is the combination of the link (2) and medium (5) resource constraints, and can be represented in the following manner:

$$\mathbf{MLR} \leq \mathbf{B} \quad (7)$$

$$\mathbf{R} \geq 0 \quad (8)$$

A.3 Network State Constraints and Fairness Criterion

Any rate allocation specified in **R** has to satisfy the linear constraints in equations 7 and 8, together with those imposed by the network state constraints and the fairness model.

The network state constraints require that no flow be allocated a rate higher than its desired rate. Thus, if the bandwidth requested by a stream s is $G(s)$, then $\forall_s R(s) \leq G(s)$. As previously discussed, in this model we only consider either inactive streams ($G(s) = 0$) or TCP streams with infinite backlog ($G(s) = \infty$).

Finally, the fairness criterion that we consider is max-min fairness. This imposes an additional constraint that no rate $R(s_i)$ can increase at the expense of $R(s_j)$ if $R(s_i) > R(s_j)$.

The resulting computational problem is to maximize the bandwidth allocation vector **R**, while satisfying the set of constraints described in the above sections.

A.4 Model Comparison with ns-2

Having described the two computation models, we now provide an analysis of the capacity achievable with these models to the model implemented in ns-2. To achieve this we have devised experiments that would determine, for a given topology and a set of active streams, the max-min fair-share points. We compare these experimentally-determined values to those computed using the two models to determine the accuracy of the computation. Given enough topologies and stream variations, we can then determine the statistical accuracy of the models.

The experiment we created is as follows. For a given set of streams in a given network topology, we simulate, using *ns-2* [1], source-rate limiting those streams over a range of rates from 50% of the computed fair-share rate to 150% of the computed fair-share rate. To avoid TCP complications, UDP data is used, with 1500 byte packets. This simulation is executed 5 times, with a different random seed each time. The 5 results are averaged to give an expected throughput for each stream for any given input rate.

Plotting these results yields graphs such as that shown in Figure 10. This particular figure is a 36-node network arranged in a 6x6-grid topology with 15 streams. The vertical line labeled "o+cl" represents the computed value for the clique model, where "o" is for omniscient, since it requires omniscient knowledge to know which links interfere with which other links. This is feasible in the simulator, though not in practice. Similarly, the vertical line "r+cd" represents the computed value for the collision-domain model, where "r" is for "realistic" as it is computable within a physical network.

To determine the accuracy of the computational models we define the fair-share points as follows. The fair-share point for bottleneck i is that point at which the throughput of more than one-third of the streams that are constrained by bottleneck i is less than the input source rate by more than 5%. All streams that are constrained by a lesser bottleneck must be capped when running the relevant simulation. We determine a drop of more than 5% by requiring this to be true for four successive data points, and then taking the first of those four points as the point of loss for that stream. While this definition may seem somewhat arbitrary, when trying several variations (*e.g.*, 8% loss by 20% of the stream, *etc.*), all pointed to approximately the same fair-share point, and visual inspection of plots has suggested that this definition is fairly reasonable.

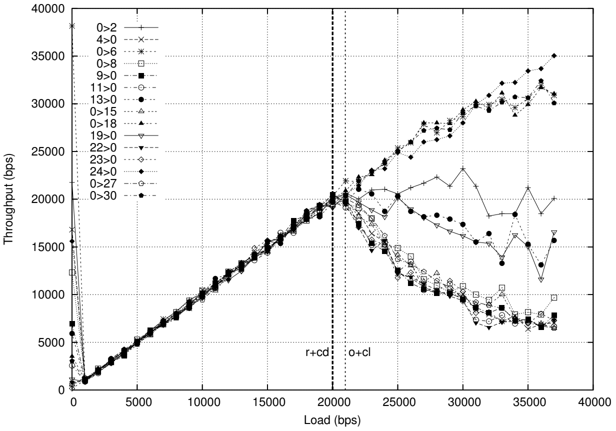


Fig. 10: Plot of Input Rate vs. Throughput for a Sample Topology, without RTS/CTS

Measured Entity	No RTS/CTS	With RTS/CTS
Avg of (Ocl - fp)/fp	0.000077	0.183
Std. Dev. of (Ocl - fp)/fp	0.11	0.357
Avg of (Rcd - fp)/fp	0.027	0.212
Std. Dev. of (Rcd - fp)/fp	0.12	0.361

Table 5: Random Topology Accuracy Results

Given this definition, we executed this experiment over 50 random topologies in a 1000x1000 m area with between 25 and 40 streams for each topology, both with and without the RTS/CTS protocol. We then compute the average error in each computation model, together with the standard deviation.

Results are shown in Table A.4. The value "Ocl" is the computed value of the clique model, "Rcd" is the computed value of the collision domain model, and "fp" is the experimentally-determined fair-share point.

As is apparent, both models are reasonably accurate at predicting the first fair-share point, generally slightly underestimating the capacity, and within about 10% deviation. The simpler collision domain model is only marginally less accurate than the more-complex clique model, and thus is quiet sufficient for our purpose in estimating the fair rates for different topologies in Sect. 5. Finally, we note that while this experiment was performed on a 36-node network, we have corroborated this observation and found it consistent across a large number of random, chain, and grid topologies.

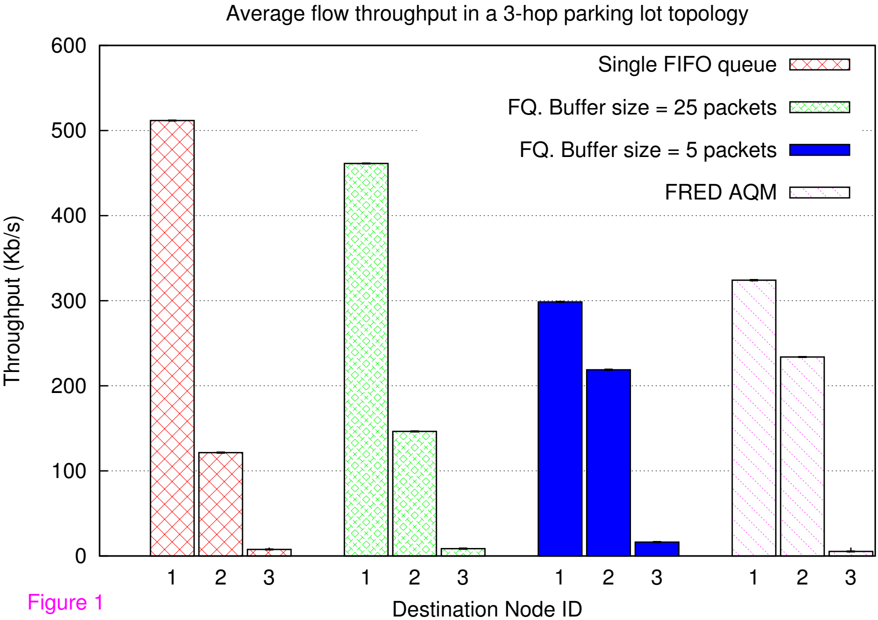
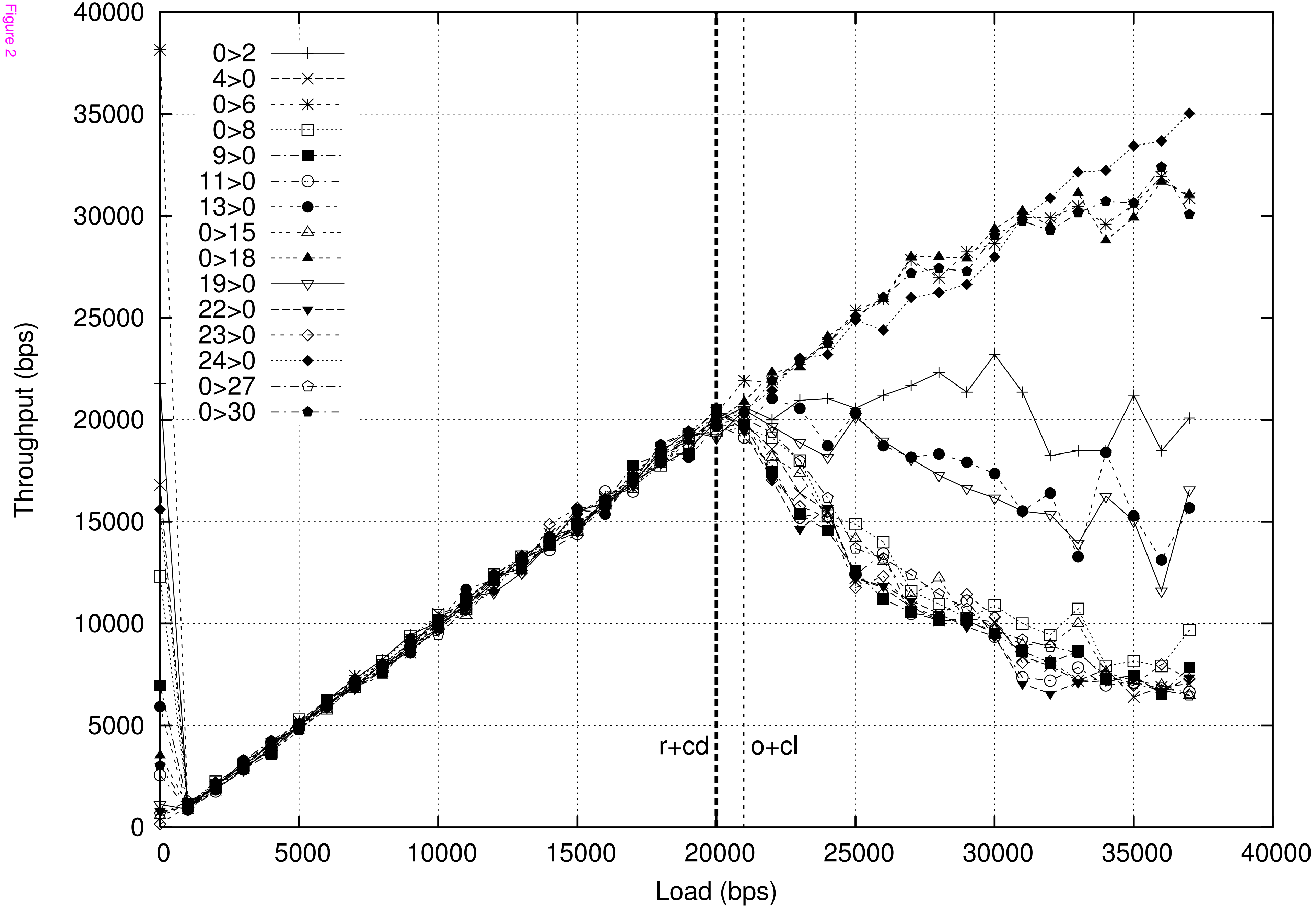

Average flow throughput in a 3-hop parking lot topology
Single FIFO queue
FQ. Buffer size = 25 packets
FQ. Buffer size = 5 packets
FRED AQM
Throughput (Kb/s)
600
500
400
300
200
100
0
1
2
3
1
2
3
1
2
3
1
2
3
Destination Node ID

Figure 1

Figure 2



Throughput (bps)
Load (bps)
0
5000
10000
15000
20000
25000
30000
35000
40000
0>2
4>0
0>6
0>8
9>0
11>0
13>0
0>15
0>18
19>0
22>0
23>0
24>0
0>27
0>30
r+cd
o+cl

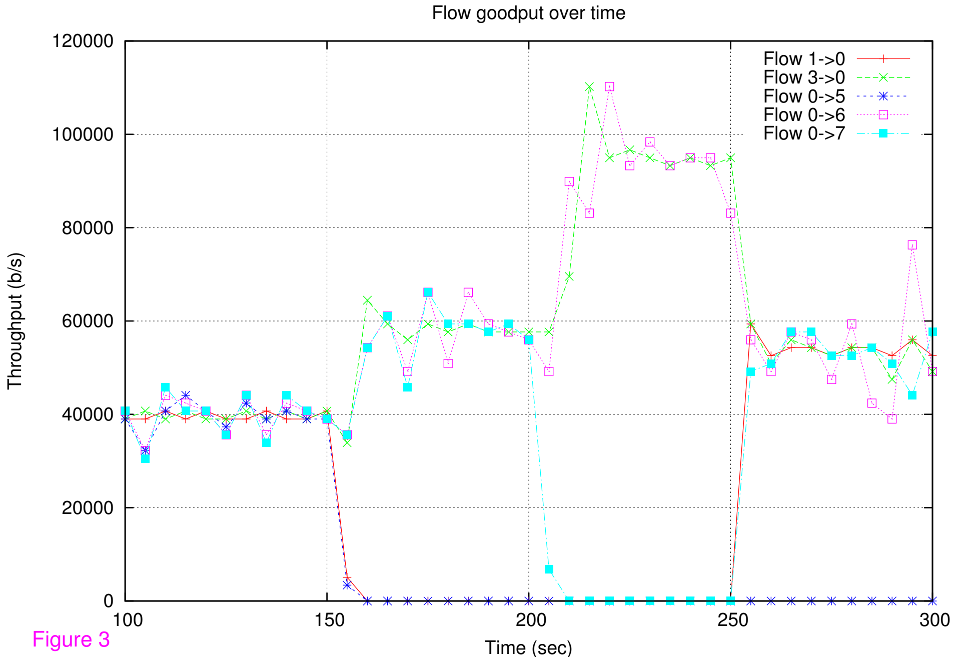

Flow goodput over time
Throughput (b/s)
Time (sec)
0
20000
40000
60000
80000
100000
120000
100
150
200
250
300
Flow 1->0
Flow 3->0
Flow 0->5
Flow 0->6
Flow 0->7

Figure 3

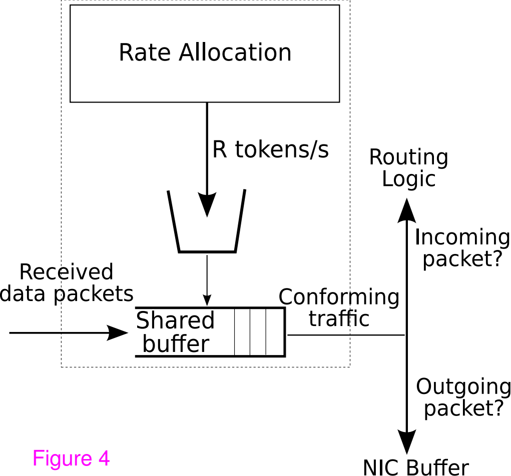

Rate Allocation
R tokens/s
Routing
Logic
Incoming
packet?
Received
data packets
Conforming
traffic
Shared
buffer
Outgoing
packet?
NIC Buffer

Figure 4

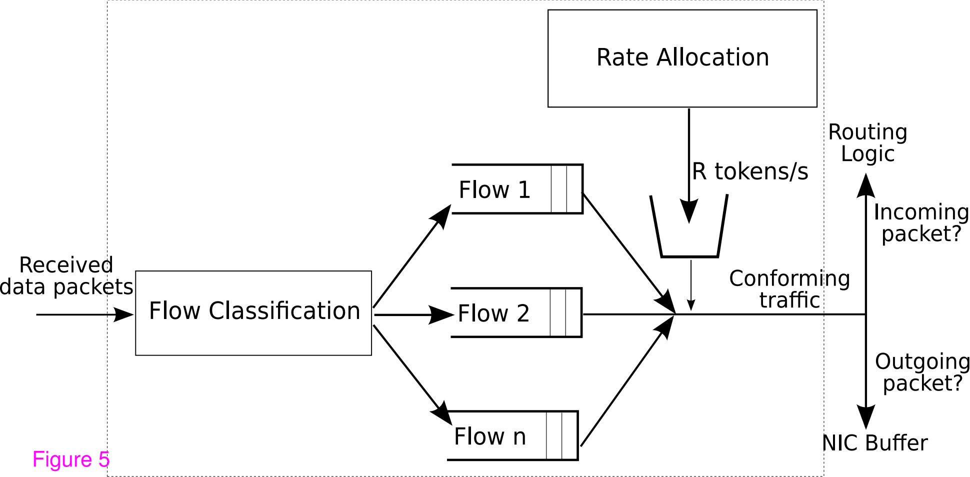

Rate Allocation
R tokens/s
Routing
Logic
Incoming
packet?
Received
data packets
Flow Classification
Flow 1
Flow 2
Flow n
Conforming
traffic
Outgoing
packet?
NIC Buffer

Figure 5

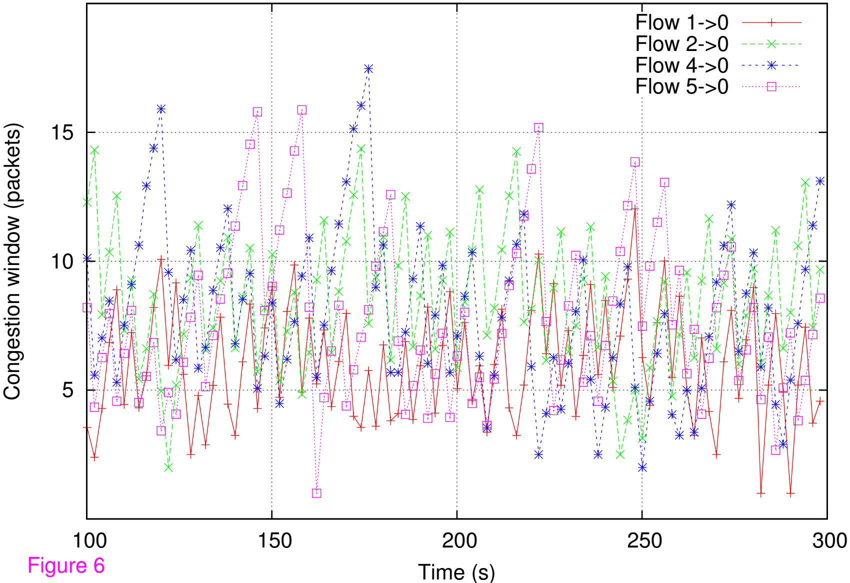

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Congestion window (packets)
15
10
5
100
150
200
250
300
Time (s)

Figure 6

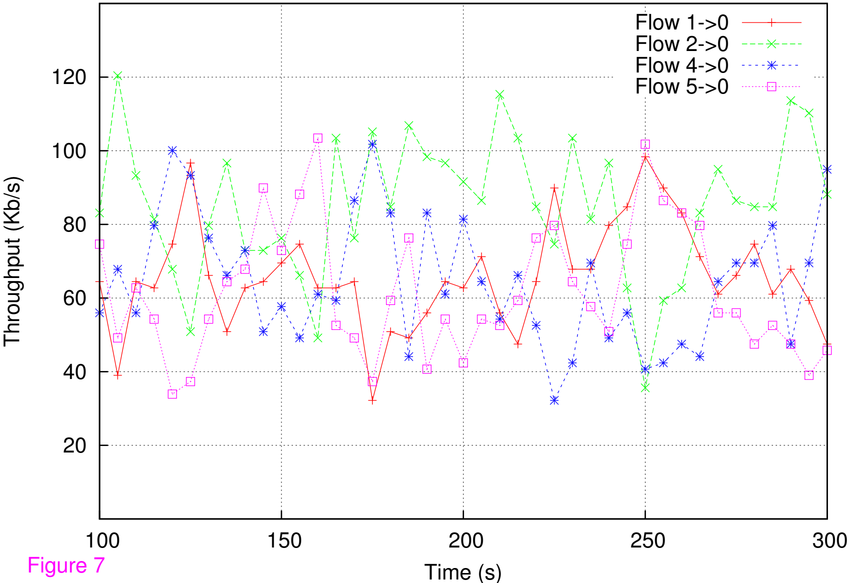

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
120
100
80
60
40
20
Throughput (Kb/s)
100
150
200
250
300
Time (s)

Figure 7

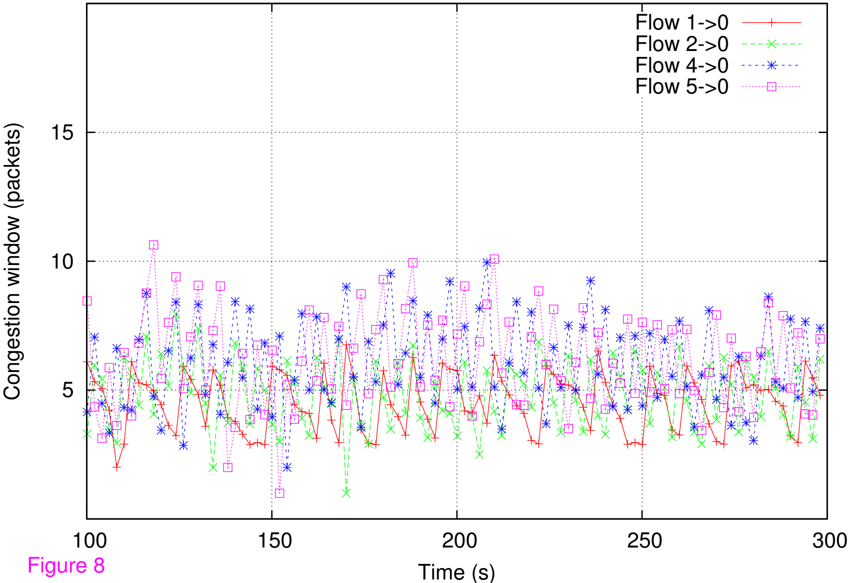

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Congestion window (packets)
15
10
5
100
150
200
250
300
Time (s)

Figure 8

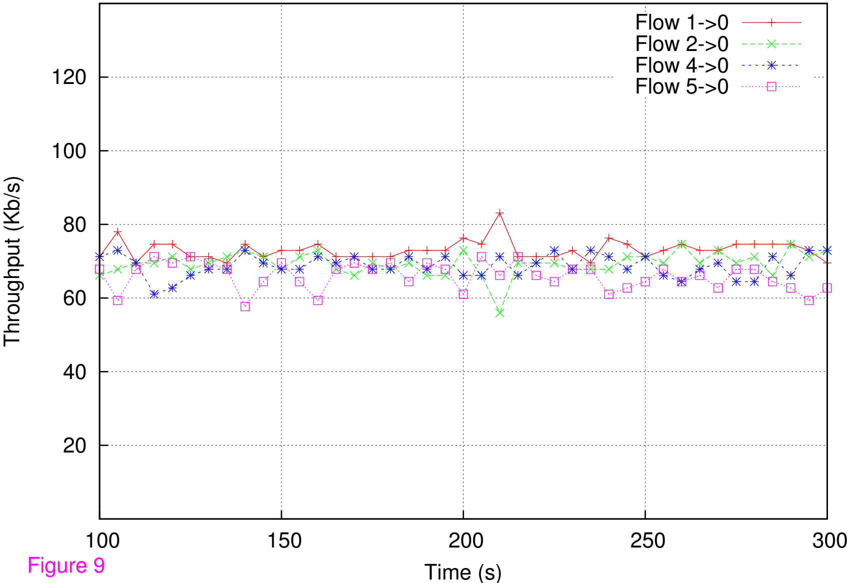
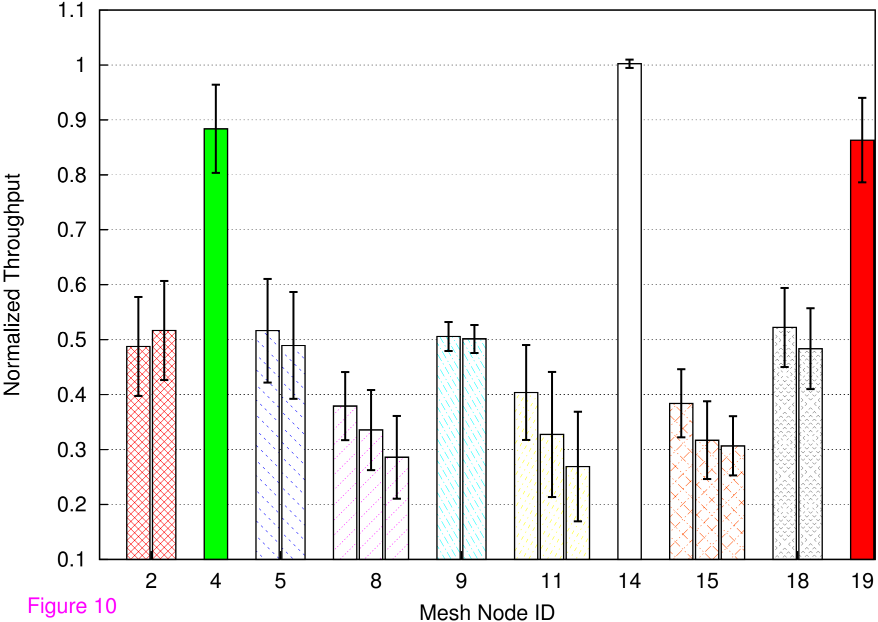

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Throughput (Kb/s)
120
100
80
60
40
20
100
150
200
250
300
Time (s)

Figure 9



1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Normalized Throughput
2
4
5
8
9
11
14
15
18
19
Mesh Node ID

Figure 10

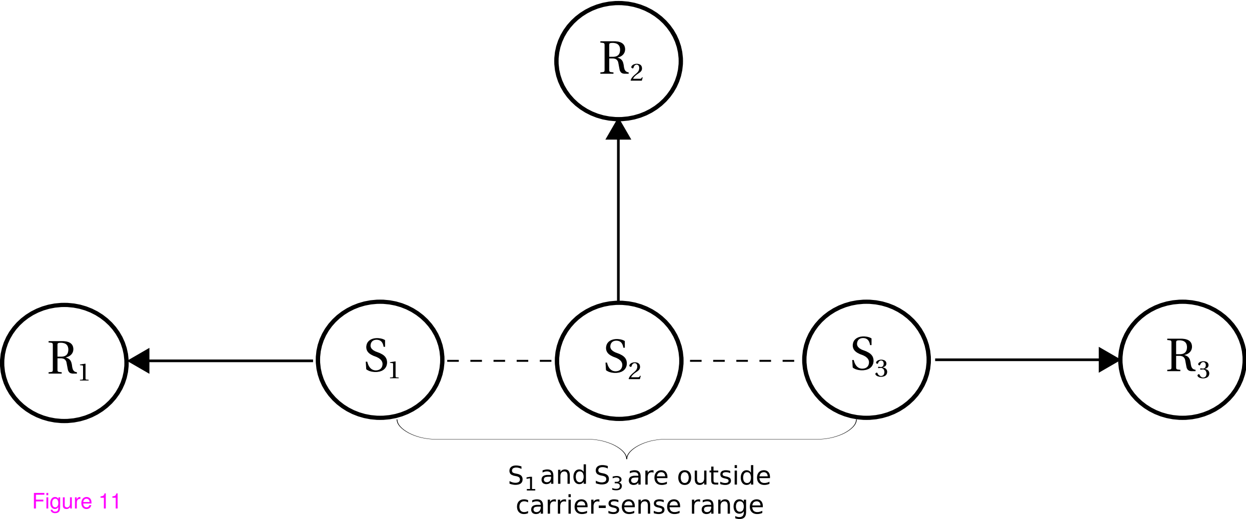


Figure 11

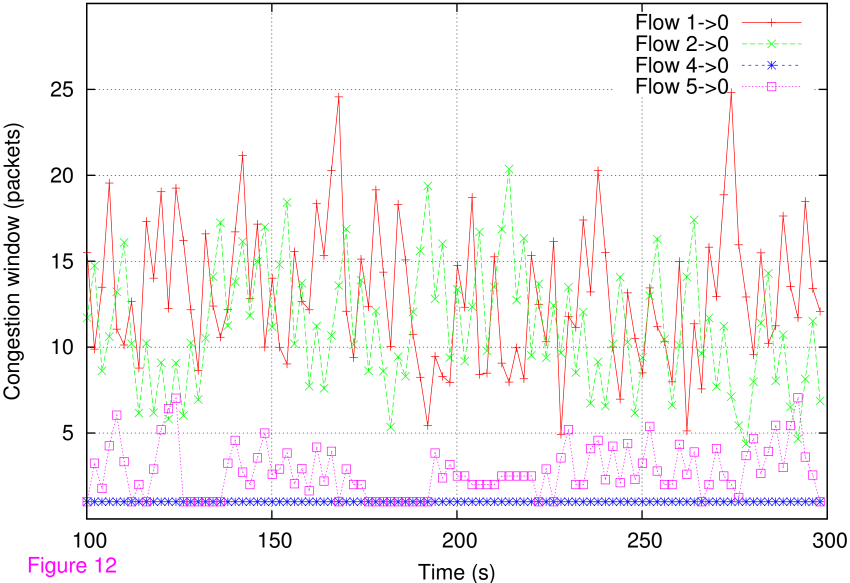

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Congestion window (packets)
25
20
15
10
5
100
150
200
250
300
Time (s)

Figure 12

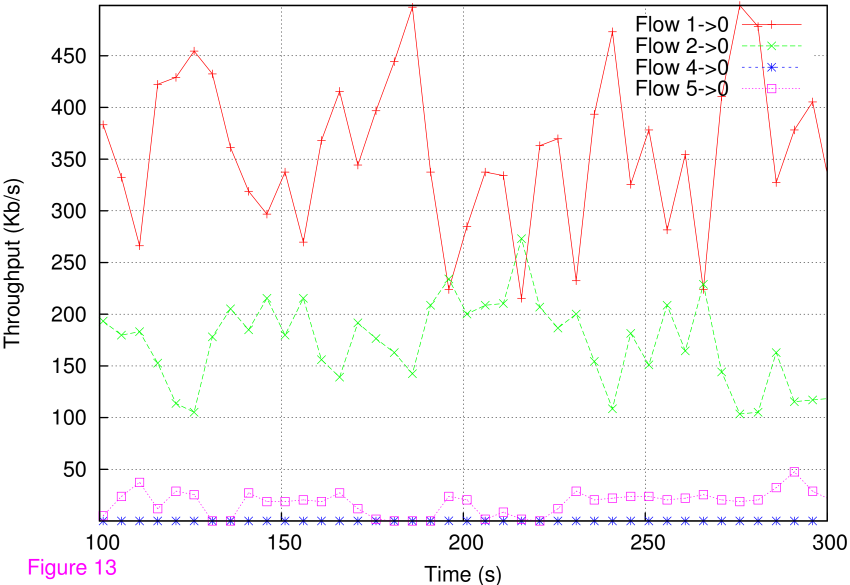

Throughput (Kb/s)
450
400
350
300
250
200
150
100
50
100
150
200
250
300
Time (s)
Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0

Figure 13

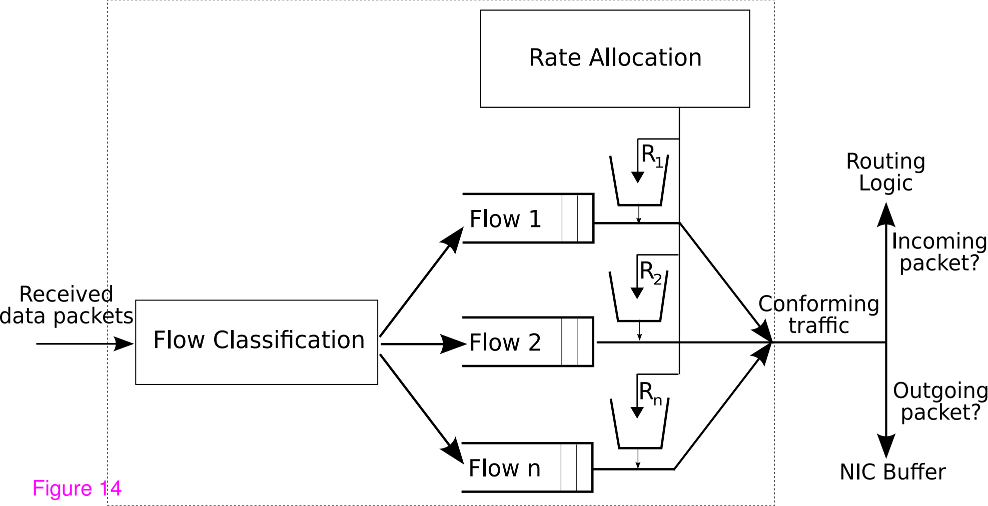

Rate Allocation
R_1
R_2
R_n
Routing Logic
Incoming packet?
Received data packets
Flow Classification
Flow 1
Flow 2
Flow n
Conforming traffic
Outgoing packet?
NIC Buffer

Figure 14

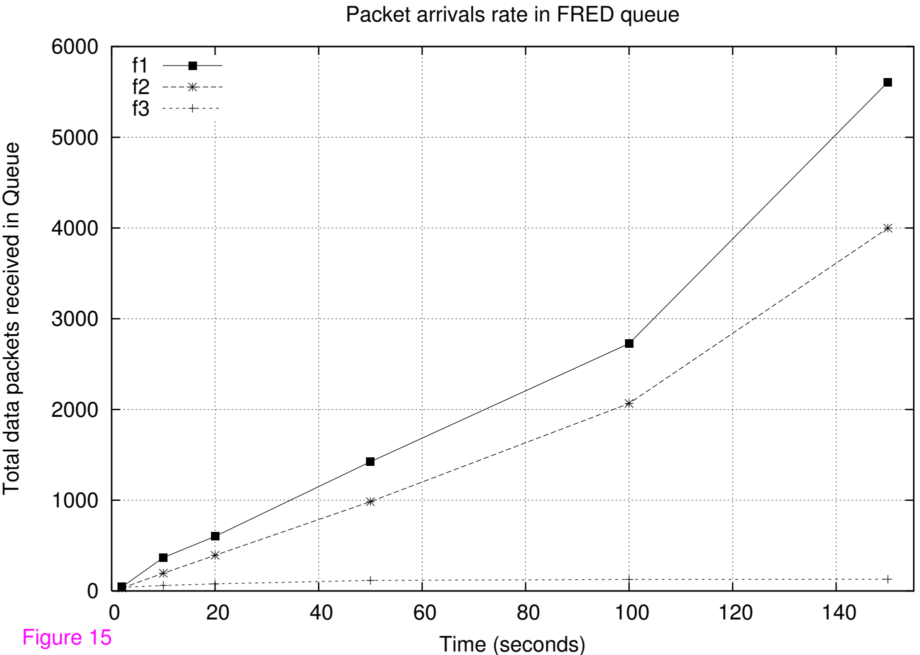
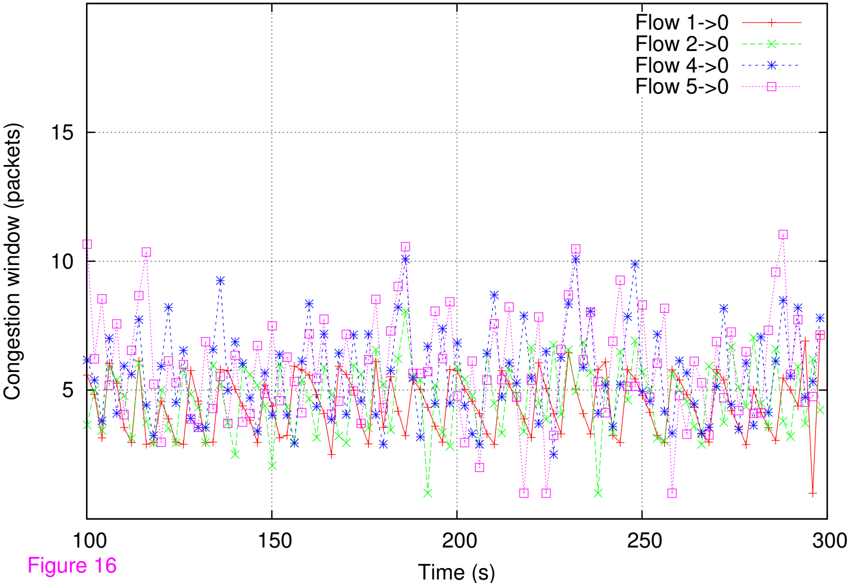

Packet arrivals rate in FRED queue
Total data packets received in Queue
Time (seconds)
f1
f2
f3
0
1000
2000
3000
4000
5000
6000
0
20
40
60
80
100
120
140

Figure 15



Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Congestion window (packets)
15
10
5
100
150
200
250
300
Time (s)

Figure 16

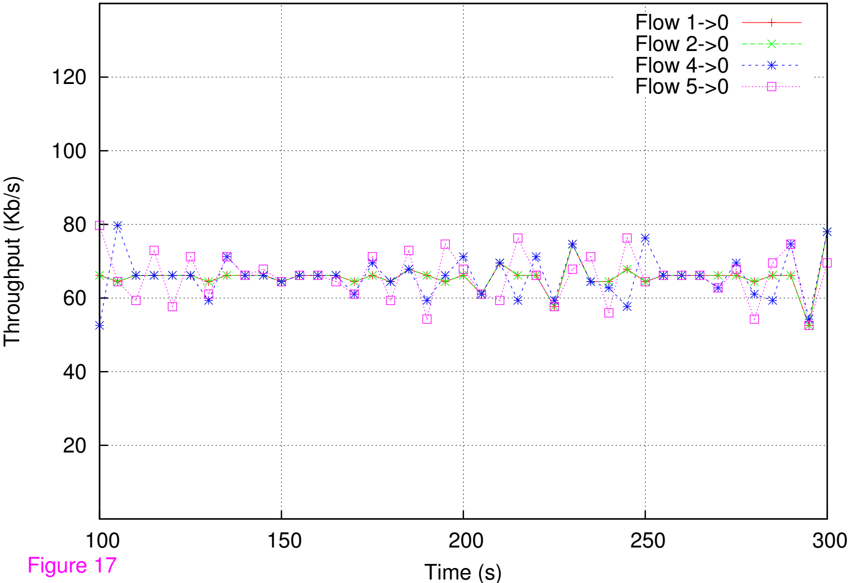

Flow 1->0
Flow 2->0
Flow 4->0
Flow 5->0
Throughput (Kb/s)
120
100
80
60
40
20
100
150
200
250
300
Time (s)

Figure 17

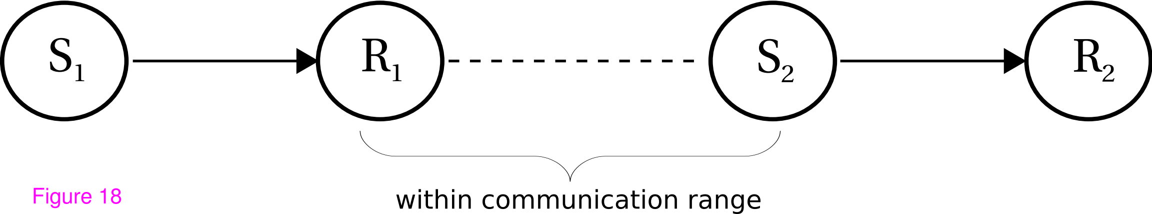


Figure 18

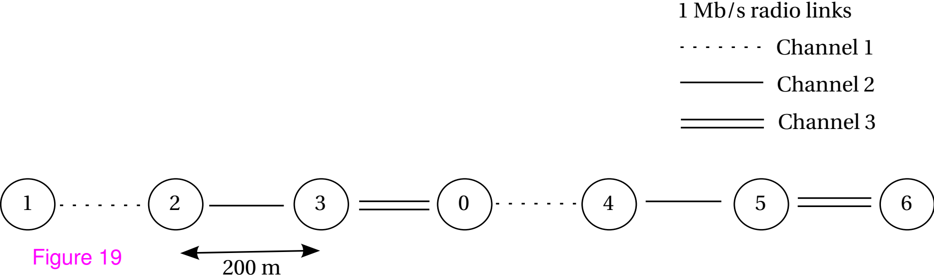
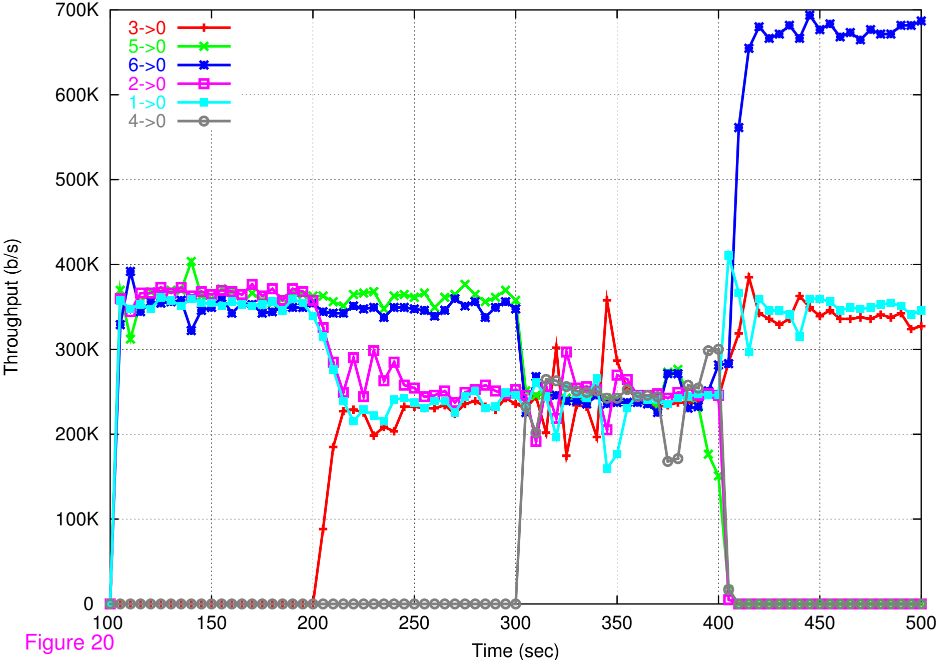


Figure 19



3->0
5->0
6->0
2->0
1->0
4->0
700K
600K
500K
400K
300K
200K
100K
0
Throughput (b/s)
100
150
200
250
300
350
400
450
500
Time (sec)

Figure 20

Additional files provided with this submission:

Additional file 1: ms.tex, 84K
http://jwcn.eurasipjournals.com/imedia/7209739291006511/supp1.tex