Achieving Edge-Based Fairness in a Multi-Hop Environment

Mustafa Arisoylu
University of California San Diego (UCSD)
Electrical & Computer Engineering Department
La Jolla, CA 92093
arisoylu@gmail.com

Wu-chun Feng
Virginia Tech
Department of Computer Science
The Synergy Laboratory
Blacksburg, VA 24060
feng@cs.vt.edu

***Abstract*— We propose efficient buffer-accounting algorithms that achieve edge-based max-min and proportional fairness in a multi- hop (MH), multi-bottleneck network environment by extending and generalizing an existing proactive queue-management scheme called GREEN. We call our scheme GREEN-MH. We envision deploying GREEN-MH at an institutional gateway in the context of a larger multi-hop and multi-bottleneck network environment. GREEN-MH uses a dynamic buffer-accounting algorithm on a per-flow basis such that certain edge-based fairness policies (e.g., max-min and proportional) are enforced among the competing TCP flows.**

Keywords: TCP, edge-based fairness, GREEN, active queue management.

I. Introduction and Related Work

Rate assignment is a fundamental design problem in multi-hop network environments. Algorithms for rate assignment can be deployed at the end host (e.g., TCP), within the network (e.g., AQM), or a combination of both. To evaluate the performance of such algorithms, we use metrics such as fairness among flows, aggregate throughput and packet loss, utilization of network resources, algorithmic complexity, and so on. And in general, because certain metrics may be antithetical to each other, constructing a rate-assignment algorithm requires design tradeoffs to be made.

For example, with the numerous definitions of fairness proposed in the literature, e.g., max-min fairness [5] and proportional fairness [3], many rate-assignment algorithms have been correspondingly proposed and studied to achieve fairness amongst flows [6], [12], [24], [25], [27], albeit at the potential expense of other metrics.

On the other hand, the rate-assignment algorithm in the ubiquitous TCP was designed prior to notions of fairness and instead ensures reliable data transfer (via flow control) and stability (via congestion control) [1], [11].

And from within the network, many of the rate assignment algorithms in active queue management (AQM), e.g., RED [20], FRED [21], SFB [22], seek to improve network performance by dropping (or marking) packets inside the network with respect to some congestion indicator (e.g., queue length).

Our AQM scheme GREEN-MH extends upon GREEN AQM [2] by leveraging the simple TCP throughput equation [9] and coupling it with per-flow buffer-accounting algorithms in a multi-hop (MH), multi-bottleneck environment in order to ensure max-min and proportional fairness.

Given that GREEN-MH requires (minimal) per-flow information, we typically expect GREEN-MH to be deployed at an institutional boundary, i.e., at the institutional edge or gateway, where the traffic should abide by a certain fairness property. We refer to such fairness as edge-based fairness such that fairness is ensured only among the flows passing through the gateway. Since the number of flows traversing an edge router is expected to be much smaller than that of a core router, making certain per-flow computations and measurements is feasible, as noted in [16], [17], [18].

Such a scheme can be considered as a compromise between fairness and overall complexity. The edge router (gateway) performs measurements and processing on the flows whereas the core routers are not modified at all. Although such a scheme may sacrifice the global fairness, it is still interesting for an institution to ensure fairness among its outgoing TCP flows. The institution in question can be a software company where large amount of files are to be downloaded by their customers.

In [2], authors discuss the effectiveness of GREEN in the case where the outgoing link from GREEN is the only bottleneck link for all the flows passing through, in other words it is assumed that application layer, advertised window size and the link capacities outside do not limit the throughput of any TCP flow. However, as found in recent research papers [17], the sender throughput may be limited by a lack of data to send. In addition to that, some of the flows may have other bottleneck links outside or simply the advertised window size may be the bottleneck.

In this work, we focus on a multi-hop and multi-bottleneck extension of the GREEN gateways, such that the outgoing link from the gateway is not necessarily the bottleneck link for all the flows passing through. We discuss a per-flow buffer accounting approach to achieve edge based proportional and max-min fairness among the flows. To put it another way, we simply transform the rate-assignment problem into the buffer domain and make all the measurements and computations in the buffer domain to achieve fairness in the rate domain.

The algorithm achieving max-min fair case requires round-trip time estimation for each flow and certain aggregate buffer size; however, the algorithm achieving proportional fairness does not require round-trip time estimations for each flow and requires relatively much smaller total buffer size. Performance of GREEN-MH is compared with SFB, FRED and Droptail which are not able to expose similar fairness performance. All these results are supported by both analysis and simulations performed in NS-2 [23].

The rest of the paper is as follows. In section II, GREEN is briefly reviewed. In section III, round-trip time estimation techniques are discussed. The concept of edge-based fairness is available in section IV. The algorithms achieving edge-based max-min and proportional fairness are presented in section V. Section VI includes simulations and discussions. Finally section VII concludes the paper.

II. GREEN

In this section, we will shortly discuss the basics of GREEN gateway that is described in more detail in [2].

It is based on the simple throughput formula of TCP [8] [9], as in equation 1 where T_i is the throughput of TCP flow i, a is the constant defined in [8], [9] and RTT_i, p_i are the corresponding round trip time and the packet drop rate respectively.

$$T_i = \frac{a \times MSS}{RTT_i \times \sqrt{p_i}}, \quad p_i = (\frac{N \times a \times MSS}{RTT_i \times C})^2 \qquad (1)$$

The GREEN gateway is assumed to be located as a gateway of an institution, enforcing fairness among the flows passing through. GREEN gateway computes a packet dropping probability for each flow and drops the packets accordingly. The packet dropping probability p_i is computed as in equation 1 where N is the number of flows and C is the capacity of the outgoing link. In [2], the outgoing link is assumed to be the only bottleneck for all the flows.

III. Round-Trip Time Estimation

In the paper on GREEN [2], the round-trip time (RTT) is assumed to either appear in the packet headers or estimated by a service called IDMAPS [19]. Passive RTT estimation techniques have recently been studied in various papers [13], [14], [15]. Detailed discussions on these techniques can be found in [26].

In this section, we will discuss a way of estimating the round-trip time of a TCP flow passing through our GREEN-MH gateway.

Considering GREEN-MH as a router located at the boundary of an institution, we can safely assume that the first router that the flows pass through is the GREEN-MH router. Since GREEN-MH router is so close to the TCP sources the interarrival time behavior of the packets transmitted from the source can be assumed not to be altered significantly.

One assumption we have for any flow i is that the maximum congestion window size $maxcwnd_i$ and the capacity of the virtual link from the source to GREEN-MH, (c_i), satisfies the following condition.

$$maxcwnd_i / c_i << RTT_i \qquad (2)$$

where RTT_i is the round-trip time of the flow i. The other assumption for this estimation technique is that each TCP source always has data to send throughout the connection life time.

If this is the case, then the simple round-trip time estimation algorithm can be implemented using only unidirectional interarrival time information of the flows.

Due to the self-clocked and bursty nature of TCP, packets transmitted from the source are assumed to be either in burst or interleaved by a certain amount of time T_{inter}. For instance, in the phase of congestion avoidance, the receiver sends cumulative acknowledgement for each b packets and the sender sends more packets with modified congestion window size. Assuming equation 2, the amount of time between two consecutive bursts, T_{inter}, is expected to be almost equal to the round-trip time of the flow.

In order to validate our argument above we have performed simulations using Network Simulator, NS-2. The related results are available in the simulations and discussions section.

In the next section, we discuss the fairness policy to be enforced among the competing TCP flows passing through GREEN-MH.

IV. Fairness Issues

In this section, we briefly discuss the concept of edge-based fairness.

A. Edge-Based Fairness

Basically, whatever the fairness policy to be enforced, edge-based fairness on a link is achieved when only all the flows passing through that link satisfies that fairness policy.

As can be seen in the Figure 1, the gateway at the boundary of the institution enforces some fairness policy on the flows passing through; however, nothing is known about the outside network (e.g., topology, queueing management in routers).

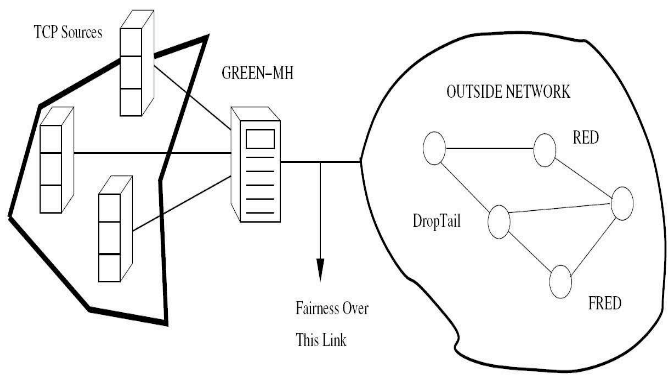

Fig. 1. Green-MH, Located at the boundary of an institution enforcing fairness over the outgoing link

The related edge router (i.e., GREEN-MH) ensures fairness among the flows which are bottlenecked at the outgoing link. We call these flows as the "In Bottlenecked" flows and the corresponding set of flows as IB. However, some other flows may have other bottleneck links at the outside network or their advertised window size $awnd$ may be the bottleneck. We call these flows "Out Bottlenecked" flows and the corresponding set of flows as OB. The edge router does not take any action regarding the fairness of the flows in OB.

In the next two sections, we briefly review the notion of proportional and max-min fairness.

1) Proportional Fairness: Proportional fairness [3], maximizes the sum of the utility of the individual rates.

More formally, let $U(X_i)$ be the utility function of X_i which is the rate of flow i. Thus the corresponding optimization problem can be written as follows.

$$Maximize \quad \sum_i U(X_i), \quad S.t. \quad Feasibility \tag{3}$$

In the proportional fairness case, the utility function U(X) is simply the logarithm function ln(X).

2) Max-Min Fairness: Given a max-min fair rate assignment [5], it is not possible to increase the rate of any flow without decreasing rate of any other flow that has already smaller or equal rate. A well-known centralized algorithm [5] as well as many distributed algorithms [12] are studied to achieve the max-min fair rate assignment.

In the next section, related algorithms for GREEN-MH are discussed in the context of the above fairness policies.

V. Achieving Edge-Based Fairness in GREEN-MH

In this section, we discuss algorithms to be deployed in GREEN-MH to achieve edge-based max-min and proportional fairness.

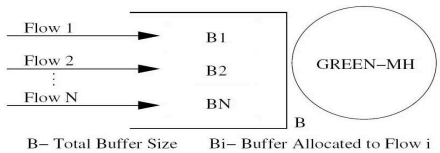

Fig. 2. Buffer Accounting in Green-MH

As can be seen in Figure 2, each flow i is allowed at most B_i number of packets inside the queue. In order to ensure this, we have a $counter_i$ for each flow i the number of packets of flow i in the queue (i.e., number of packets buffered). If the $counter$ value reaches B_i, no more incoming packets of flow i are allowed in the queue. The queue is itself a FIFO queue and no other mechanism is required.

The choice of each B_i mainly depends on the steady state TCP model described in [8], [9]. Intuitively, each TCP flow in a congestion avoidance phase indicate roughly a periodic behavior [8], such that the congestion window opens up to a certain maximum level and then is halved and then grow again by one in every round trip time. This behavior also can be seen in Figure 3.

When we limit the buffer share of a TCP flow to B_i then the maximum congestion window W is also limited roughly to B_i.

Using the model in [8], [9], the throughput of a TCP flow can be modeled as

$$T_i = \frac{3W_i}{4RTT_i} \approx \frac{3B_i}{4RTT_i} \tag{4}$$

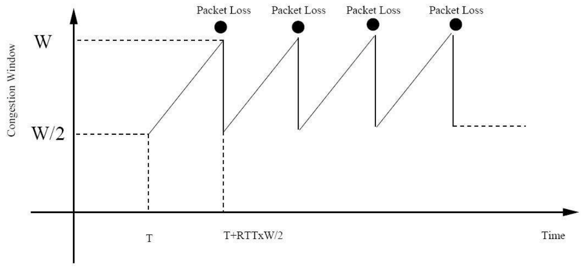

Fig. 3. Steady State TCP Congestion Window Characteristics

where W_i is the maximum congestion window size of flow i. Thus, in case of a single bottleneck link, a static buffer management scheme in agreement with [10] can be used to have equal rates such that $B_i = \frac{RTT_i}{\sum_j RTT_j} \times B$ where B is the total buffer size and RTT_i is the round trip time of flow i.

The model described in [8], [9] is valid as long as the packet loss percentage is sufficiently small. In [7], a new model is proposed including the effect of timeouts such that when the packet loss rate reaches a certain value, the throughput of TCP deviates from the model in [8], [9] substantially.

So, we need to discuss the minimum buffer space allocated to a flow in a queue in order to realize the model in [8], [9]. Using model described in [8] the probability of packet loss (or packet loss fraction) can be roughly calculated as $P = \frac{1}{3/4W(W/2+1)}$.

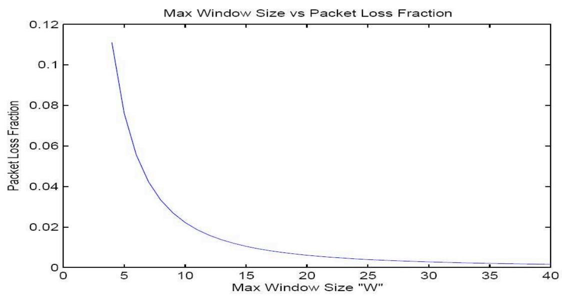

Fig. 4. Packet Loss Rate vs. Maximum Window Size

The plot for P vs W can be seen in Figure 4. For instance, using the above model W should be at least 30 in order to have P around 0.003. Assuming 0.003 is small enough, we need to provide at least 30 packets of buffer share, B_i, to the flow with the minimum round trip time. This is the main disadvantage for this scheme since it requires some total buffer space.

Next, we propose a buffer accounting algorithm that allocates the buffer shares, B_is, to the individual flows in case of multi-bottleneck case where the outgoing link of the GREEN-MH is not necessarily the only bottleneck link for all the flows passing through.

In this case, we assume that if a flow is bottlenecked somewhere else (a link outside or advertised window awnd, in other words $\in OB$), it is not able to reach the maximum allowable buffer size in the queue. So the unused part of the buffer allocated to such flows is shared by the flows that are bottlenecked in GREEN-MH outgoing link, (i.e., $\in IB$).

Next, we discuss achieving edge-based max-min fairness in a multi-bottleneck environment.

A. Edge-Based Max-Min Fairness

In this section, we propose a buffer accounting algorithm to ensure edge-based max-min fairness among the TCP flows. We adapt the explicit rate algorithm used in [12]. We call the algorithms as GREEN-MH-rtt or GREEN-MH-est where the former uses the real round trip time values (assumed that round trip times are available on the header of each packet) and the latter estimates the round trip time values using the simple technique described in section III.

1) Algorithm: GREEN-MH-rtt/GREEN-MH-est: Step by step algorithm is as follows

STEP1: Initialization

$$B_i = \frac{B}{\sum_j RTT_j} \times RTT_i \quad \forall i \in TF \tag{5}$$

where B_i is the buffer share for flow i, and B is the overall buffer space available such that $B = \sum_j MinB \times rtt_j/rtt_{min}$. $rtt_{min} = min_{i\in TF}(rtt_i)$ where TF is the set of all flows and $MinB$ is the coefficient that should be set according to the discussions on Figure 4.

STEP2: Measurement

For each flow i, find out (the maximum buffer usage), $MaxBuf_i = max(counter_i) : n \times Tm_i < time < (n+1) \times Tm_i, n = 0, 1, 2, ...,$ maximum value of $Counter_i$ within a time interval Tm_i

STEP3: Marking Flows

$$If \quad MaxBuf_i < \beta \times B_i \quad then \quad Have \quad Flow \quad i \in OB \tag{6}$$
$$Else \quad Have \quad Flow \quad i \quad \in IB \tag{7}$$

where $\beta \leq 1$ is the buffer utilization coefficien. β is set close to 1 by the system admin as a threshold value to determine whether a flow is fully utilizing its buffer share or not.

STEP4: Buffer Allocation

$$B_{rem} = B - \sum_{j\in OB} MaxBuf_j \tag{8}$$

where B_{rem} is the total unused buffer by the outbottlenecked flows.

$$B_i = \frac{RTT_i}{\sum_{j\in IB} RTT_j} \times B_{rem} \quad \forall i \in IB \tag{9}$$

For only one iteration, there may be inconsistency with the new calculated buffer share and the sets IB and OB. In [12], it is shown that two iterations are sufficient to reach to a consistent state.

Additional discussion on the the flows that are limited by their advertised window sizes and their interaction with the above algorithm is available in [26].

In order to measure $MaxBuf_i$ precisely, the value of the time interval that keeps the maximum buffer usage, Tm_i should satisfy the following condition $Tm_i > \frac{W_i \times RTT_i}{2}$, where W_i and RTT_i are the maximum congestion window size and the round trip time of flow i.

B. Edge-Based Proportional Fairness

The rate of a flow i, X_i as in equation 3, corresponds to the throughput of each TCP flow, T_i, defined in equation 4. So the optimization problem defined in the section of proportional fairness becomes as follows

$$Maximize \quad M(B_v) = \sum_i ln(T_i) = \sum_i ln(\frac{\alpha_i B_i}{RTT_i}) \tag{10}$$
$$Subject \quad to: \quad g_0(B_v) = \sum_i B_i \leq B \tag{11}$$
$$g_i(B_v) = B_i \leq bwnd_i, \quad \forall i \in TF \tag{12}$$

where B is the total computed buffer space, TF is the set of all flows, and B_v is the buffer allocation vector whose elements are B_is. On the other hand, $bwnd_i = min(awnd_i, ownd_i)$ where $awnd_i$ is the advertised window size of flow i and $ownd_i$ is assumed to be the maximum buffer space that flow i can utilize throughout the outside network.

The objective function, M, to be maximized is a concave function and the feasible set defined by the constraints is a convex set. Therefore the above problem can be solved using the Kuhn-Tucker conditions. If the vector $B_v^* = B_1^*, B_2^*, ..., B_{|TF|}^*$ is in the interior of the feasible set and satisfies the Kuhn-Tucker conditions, B_v^* solves the maximization problems. α_i above is equal to 1 if $B_i^* = awnd_i$ and is equal to $3/4$ if $B_i^* < awnd_i$. Consider equation 4 and note that when a tcp flow is limited by the advertised window size the throughput becomes $T_i = awnd_i/RTT_i$. The discussion on the effect of α_i on the optimization problem can be found in [26]. We also assume that $\sum_i bwnd_i > B$.

In our case, the Kuhn-Tucker conditions are as follows:

$$\nabla M(B_v) - \sum_i \lambda_i \times \nabla g_i(B_v) = 0 \tag{13}$$
$$g_i(B_v) - bwnd_i \leq 0 \quad \forall i = 1....|TF| \tag{14}$$
$$g_0(B_v) \leq B, \quad \lambda_i \times (bwnd_i - g_i(B_v)) = 0 \tag{15}$$
$$\lambda_0 \times (B - g_0(B_v)) = 0, \quad \lambda_i \geq 0 \tag{16}$$

Solving the above equations simultaneously, the following equations are obtained.

$$\sum_j B_j^* = B, \quad B_j^* = \frac{1}{\lambda_0 + \lambda_j}, \quad B_j^* > 0 \tag{17}$$
$$If \quad B_j^* < bwnd_j \quad then \quad \lambda_j = 0 \quad and \quad B_j^* = \frac{1}{\lambda_0} \tag{18}$$
$$Else \quad B_j^* = bwnd_j = \frac{1}{\lambda_0 + \lambda_j} \tag{19}$$

From the above results, one can easily see that the optimum point gives equal buffer shares to all flows which are not able to reach their $bwnd$s. That is, proportional fairness in our case results in throughputs that are exactly inversely proportional to the corresponding round-trip times, (see equation 4). Interestingly, the buffer share for a flow i does not depend on the round-trip time of the flow.

We call the algorithm in GREEN-MH achieving proportional fair rates as GREEN-MH-prop.

1) Algorithm: GREEN-MH-prop: The algorithm for GREEN-MH-prop is the same as GREEN-MH-rtt with all the RTTs are equal to each other. This algorithm is also exactly the buffer allocation version of the explicit rate allocation in [12] as well as the classical water filling algorithm.

The algorithm assigns the same buffer allocation to the flows that are not able to reach their $bwnd_i$s, subject to the constraint that the sum of all buffer allocations, B_is, are equal to B. The other flows are assigned buffer spaces which are equal to their $bwnd_i$s. The above buffer allocation is exactly the same allocation satisfying the Kuhn-Tucker conditions described previously.

It is interesting that the algorithm leads to max-min fair buffer allocation which yields proportional fair rate allocation for each flow.

VI. Simulations and Discussions

In this section, we present simulation results for our algorithms achieving edge-based max-min, GREEN-MH-(est/rtt) , and proportional fairness, GREEN-MH-prop.

We perform several simulations for different scenarios on single and multi bottleneck networks to examine the performance of the algorithms that we propose for the edge-based fair allocation. For all the scenarios, the packet sizes are set to 1KB, the simulation time is set to 360 seconds, and TCP Reno is used as the transport protocol. We generate 50 to 200 TCP flows with different round-trip times from node $S(i)$ to node $R(i)$, and each TCP flow is assumed to have always some packet to send. Due to the space limitations, the simulation results for single bottleneck case is not presented here, they are available at [26]. The performance of our algorithm on both scenarios are similar.

We use the Jain's Fairness Measure [4] for the performance comparison of our algorithm with the other well-known mechanisms like DropTail, FRED, SFB. As can be seen in equation 20, Jain's fairness measure stays between 0 and 1. The closer the value to 1, the more fair the rate allocation is.

$$Jain'sFairness(x_1, ..., x_i, ...x_N) = \frac{(\sum_N^{i=1} x_i)^2}{N \sum_N^{i=1} x_i^2} \qquad (20)$$

where x_i is the throughput of flow i. In calculation of this fairness index, real throughput values are used for all the queue management schemes except GREEN-MH-prop. As found before, GREEN-MH-prop results in throughput for each flow that is exactly inverse proportional to the corresponding round trip time. So normalized throughputs are used in the computation of GREEN-MH-props fairness index, where normalized throughout is the real throughput times the corresponding round trip time. The closer the index to 1, the closer the normalized throughput values to each other, implying that the throughputs are distributed as inversely proportional to the round trip times.

Due to the space limitations, we plotted all the fairness indices in the same plot, GREEN-MH-(rtt,est), FRED, SFB and Droptail are compared with each other in terms of max-min fairness, whereas GREEN-MH-prop fairness index individually indicates the fairness for proportional case.

In order to have a fair comparison in all simulations, the total buffer size for GREEN-MH is set to equal to those of other schemes (SFB, FRED, DropTail). Moreover, the minimum buffer, MinB, for both GREEN-MH-(rtt/est) and GREEN-MH-prop are set to 30.

As in the case of fairness comparison, in order to save space we plotted average buffer usages of all the schemes in single plots. On one hand, average buffer sizes of GREEN-MH-(rtt/est) and FRED, SFB, and Droptail are compared. On the other hand, we would like to compare the average queue sizes of GREEN-MH-(rtt/est) and GREEN-MH-prop.

We use the topology in Figure 5 where half of the TCP flows (even numbered) are destined to node N_1 and the other half (odd numbered) are destined to node N_3. The capacity of the link between nodes N_2 and N_3 is kept much smaller than that of between GREEN-MH and N_1.

The even numbered flows are bottlenecked in the GREEN-MH router and the odd numbered flows are bottlenecked at the link between N_2 and N_3.

We examine the fairness among the flows in IB (even numbered flows in this scenario), and compare it with the fairness of the other queue mechanisms.

Figure 6 indicates the Jains fairness among the flows that are bottlenecked at the outgoing link from GREEN-MH, (i.e. even numbered flows). The results are pretty similar to the single bottleneck case such that GREEN-MH (est/rtt/prop) most of the time has fairness measure close to 1, whereas SFB, FRED, and DropTail have fairness measures around 0.7 and 0.8. The normalized throughputs are used in the computation of fairness measure for GREEN-MH-prop.

Figure 7, on the other hand, indicates the number of packets sent by each 100 flow with different round-trip times. Again GREEN-MH-rtt/est generate throughputs for each flow which are very close to each other. The link utilizations are always around 100% so it can easily be assumed that the assignment is edge-based max-min fair. However, other schemes are not able to enforce similar fairness and the flows with smaller round trip times have naturally higher throughputs than the ones with higher round trip times.

As can be seen in Figure 8, FRED has the smaller average queue size than GREEN-MH-(rtt/est), SFB and droptail. On the other hand, again GREEN-MH-prop has much smaller average buffer space than the other GREEN-MH variations. The analysis of the average buffer occupancy under such isolated buffer accounting mechanisms is under investigation. Computed total buffer size, B, is again much larger than the average values in all variations of GREEN-MH. As a result, in this case much smaller total buffer sizes may be used while making all the related computations with respect to B. The analysis of buffer occupancy is left as a future work.

In our last scenario for max-min fair case, we consider the single bottleneck link case; however, this time some of the flows are bottlenecked by their advertised window sizes. The advertised window sizes are randomly chosen. The detailed results of the simulation can be found at [26]. GREEN-MH-(est/rtt) both generate fairness measures greater than 0.95.

Moreover, in all scenarios, GREEN-MH-est performs as well as GREEN-MH-rtt indicating the accuracy of the round trip time estimation technique described in section III.

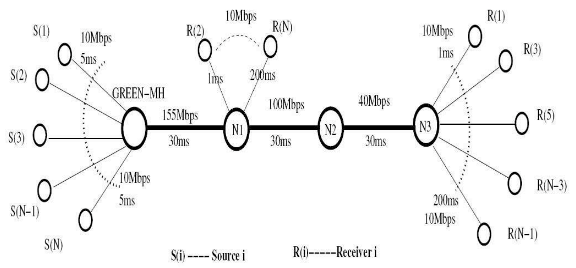

Fig. 5. Two Bottleneck Link Case

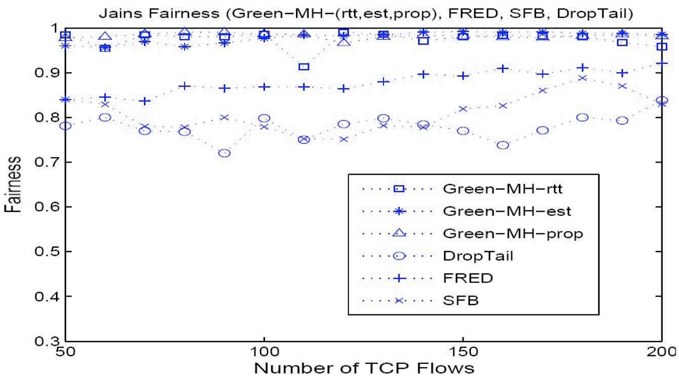

Fig. 6. Jain's Fairness, Two Bottleneck Link Case

VII. Conclusions and Future Work

In this paper, we examined per-flow queue accounting algorithms that are designed to achieve certain fairness policies explicitly.

GREEN-MH-(rtt/est) and GREEN-MH-(prop) are found to achieve edge-based max-min and proportional fairness respectively in both analysis and simulations. On the other hand, unlike GREEN-MH-prop, GREEN-MH-(rtt/est) requires round trip time information of each flow. It is also found that both GREEN-MH-(rtt/est) and GREEN-MH-(prop) requires certain aggregate buffer space.

It can also easily be seen from the simulations that other existing queue management schemes are not able to remove the round-trip time bias on the TCP throughput as well as the effect of other bottleneck links.

As future work, the algorithm is to be extended for the case of small buffers and with no round trip time estimation requirements. Moreover, the analysis of the buffer occupancy in our model is also left as a future work.

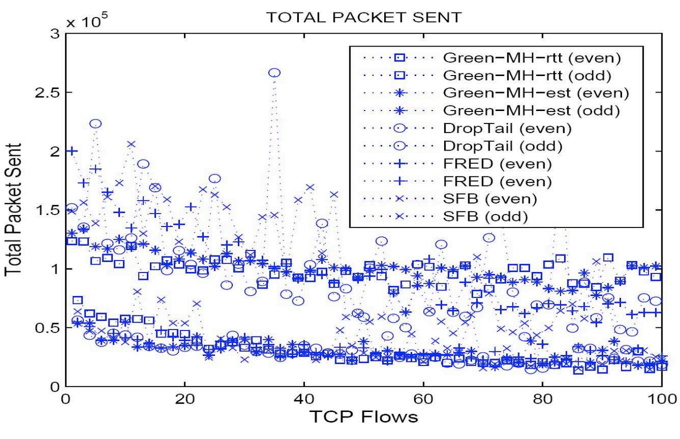

Fig. 7. Total Packets Sent, Two Bottleneck Link Case

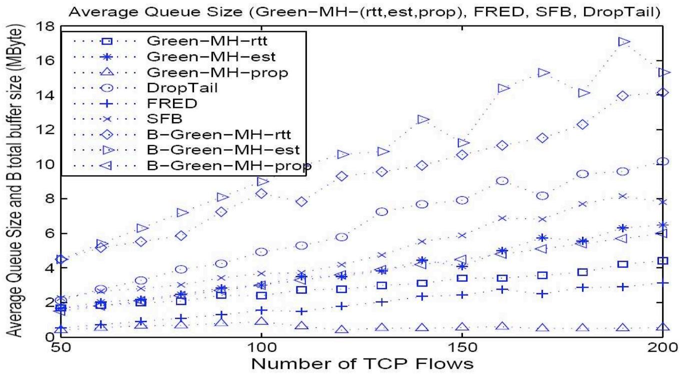

Fig. 8. Avg and Total Buffer Sizes; B-GREEN-MH indicates total buffer size, others are avg. sizes

References

[1] V. Jacobson, " Congestion Avoidance and Control," in SIGCOMM, Aug. 1988.
[2] W. Feng, et al., " GREEN: Proac- tive Queue Management over a Best-Effort Network," in IEEE GLOBECOM, Nov. 2002.
[3] F. P. Kelly, "Charging and Rate Control for Elastic Traffic," in European Trans. on Telecom., 8(1):135-155, Jan. 1987.
[4] D. Chiu, R. Jain, " Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks," in Computer Networks and ISDN Systems, 17:1-14, 1989.
[5] D. Bertsekas, R. Gallager, " Data Networks," in 2nd Edition, Prentice Hall, 1992.
[6] F. P. Kelly, et al., "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," in Journal of the Oper. Res. Society,, 1998.
[7] J. Padhye, et al., " Modelling TCP Throughput: A Simple Model and its Empirical Validation," in SIGCOMM,, 1998.
[8] S. Floyd, K. Fall, " Promoting the Use of End-to-End Con- gestion Control in the Internet," in IEEE Transactions on Networking,, Aug. 1999.
[9] M. Mathis, et al., " The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," in Computer Communication Review,, 27(3), Jul. 1997.
[10] B. Suter, et al., " Design Considerations for Supporting TCP with Per-Flow Queueing," in INFOCOM, 1998.
[11] G. Vinnicombe, " On the Stability of Networks Operating TCP-like Congestion Control," in IFAC World Congress, Spain, 2002.
[12] A. Charny, et al., " Congestion Control with Explicit Rate Indication, " in IEEE ICC,, 1995.
[13] S. Jaiswal, et al., " Inferring TCP Connection Characteristics Through Passive Measurements, " in INFOCOM, 2004.
[14] H. Jiang, C. Dovrolis, " Passive Estimation of TCP Round Trip Times, " in (SIGCOMM CCR), 32(2), 2002.
[15] Y. Zhang, et al., " On the Characteristics and Origins of Internet Flow Rates, " in (SIGCOMM), 2002.
[16] M. Yuksel, et al., " Congestion Pricing on Edge-to-Edge Congestion Control, " ICC, 2003.
[17] S. Karandikar, et al., "TCP Rate Control," in ACM CCR,,30(1):45-58, Jan. 2000.
[18] I. Stoica, et al., "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocations In High Speed Networks," SIGCOMM,, Aug. 1998.
[19] P. Francos, et al., " IDMaps: A Global Internet Host Distance Estimation Service," IEEE Transactions on Networking,,Oct. 2001.
[20] S. Floyd, V. Jacobson, " Random Early Detection Gateways for Congestion Avoidance, " in IEEE Transactions on Networking, 1(4):397-413, Aug. 1993.
[21] D. Lin, R. Morris, " Dynamics of Random Early Detection," in SIGCOMM, 1997.
[22] W. Feng, et al., "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fair- ness," in INFOCOM, 2001.
[23] NS-2, Network Simulator, http://www.isi.edu/nsnam/ns.
[24] M. Arisoylu, T. Javidi, and R. L.Cruz, " End-to-end and MAC-layer fair rate assignment in interference-limited wireless access networks," in IEEE ICC, 2006.
[25] M. Arisoylu, R.L Cruz, and T. Javidi "Rate Assignment in Micro-Buffered High Speed Networks," in Allerton, 2005.
[26] M. Arisoylu, W. Feng "Achieving Edge Based Fairness in a Multi-Hop Environment," in Los Alamos Unclassified Report LA-UR-04-5989. ,2004
[27] M. Arisoylu, S. Ergut, R. L. Cruz, R. Rao "Packet Size Aware Path Setup For Wireless Networks," in IEEE CCNC ,2008