



**HAL**  
open science

## MaxNet and TCP Reno/RED on Mice Traffic

Truong Khoa Phan, Tuan Thanh Tran, Duc Doan Nguyen, Nam Thoai

► **To cite this version:**

Truong Khoa Phan, Tuan Thanh Tran, Duc Doan Nguyen, Nam Thoai. MaxNet and TCP Reno/RED on Mice Traffic. Modeling, Simulation and Optimization of Complex Processes, Mar 2009, Ha Noi, Vietnam. pp.247-255, 10.1007/978-3-642-25707-0\_20 . hal-00721882

**HAL Id: hal-00721882**

**<https://inria.hal.science/hal-00721882v1>**

Submitted on 7 Nov 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# MaxNet and TCP Reno/RED on Mice Traffic

Khoa T. Phan, Tuan T. Tran, Duc D. Nguyen, and Nam Thoai

**Abstract** Congestion control is a distributed algorithm to share network bandwidth among competing users on the Internet. In the common case, quick response time for mice traffic (HTTP traffic) is desired when mixed with elephant traffic (FTP traffic). The current approach using loss-based with Additive Increase, Multiplicative Decrease (AIMD) is too greedy and eventually, most of the network bandwidth would be consumed by elephant traffic. As a result, it causes longer response time for mice traffic because there is no room left at the routers. MaxNet is a new TCP congestion control architecture using an explicit signal to control transmission rate at the source node. In this paper, we show that MaxNet can control well the queue length at routers and therefore the response time to HTTP traffic is several times faster than with TCP Reno/RED.

**Keywords** Active queue management • TCP congestion control • MaxNet

## 1 Introduction

TCP Reno [Jac90] uses AIMD mechanism [Jac88] in which the sending rate is increased until packet loss happens. To avoid buffer overflows at router, AQM RED (Active Queue Management Random Early Detection) [FJ93] can be used in conjunction with TCP Reno. The weakness of RED is that it does not take into account the number of incoming flows arrived at a bottleneck link to perform appropriate treatments to lighten down the heavy load. When there are a large number of sharing flows at a bottleneck link, the offered load will not be decreased,

---

K.T. Phan · T.T. Tran · D.D.Nguyen · N. Thoai  
Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology,  
Vietnam  
e-mail: [khoaphan@cse.hcmut.edu.vn](mailto:khoaphan@cse.hcmut.edu.vn); [nam@cse.hcmut.edu.vn](mailto:nam@cse.hcmut.edu.vn)

also the queue length at router does not change. This results in longer response time for the mice traffic.

MaxNet [SAW08, MRB05, BLM03] is a new congestion control mechanism using multi-bit signal instead of packet loss to control the sending rate. Besides, MaxNet router can control the magnitude of transient queues well regardless the number of new arrival flows. In other words, MaxNet can always keep a free space at routers for mice traffic to fly though. As a result, the response time to HTTP requests is much shorter with MaxNet than with TCP Reno/RED.

The rest of this paper is structured as follows. Section 2 is the theoretical analysis of queueing delay of RED and MaxNet routers. Section 3 shows the efficiency of MaxNet's quick control of the transfer rate of mice flows. We have some experiments and evaluations in Sect. 4. Finally, we present the conclusions and future work in Sect. 5.

## 2 Equilibrium Queueing Delay at RED and MaxNet Router

### 2.1 Queueing Delay at RED Router

RED routers calculate and compare the average queue length based on the two parameters: maximum threshold and minimum threshold. Base on this comparison, RED router operates in three modes [FJ93]:

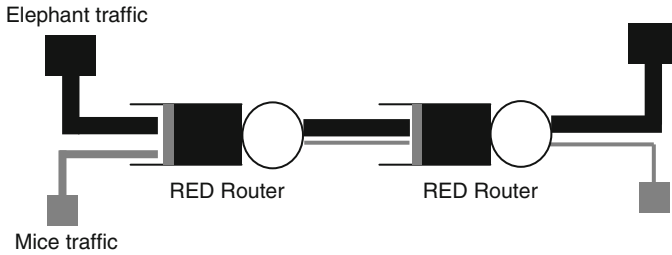
- “No dropped”: when the average queue length is less than the value of the minimum threshold, router assumes that its link is under-utilized. As a result, all packets are allowed to go through without marking or dropping.
- “Probabilistic dropped”: when the average queue length is between the minimum and maximum thresholds, router assumes that the network can be saturated, it then marks/drops packets with a probability corresponding to the traffic load.
- “Forced dropped”: when the average queue length is greater than the maximum threshold, all packets that go through the router will be marked/dropped to reduce heavy load on the link.

From the flow-level model of AIMD, the window size of TCP Reno is updated using the following equation [WJL06]:

$$w_i'(t) = \frac{1}{T_i(t)} - \frac{2}{3}x_i(t)q_i(t)w_i(t) \quad (1)$$

where  $T_i(t)$  is the round-trip-time;  $x_i(t) = w_i(t)/T_i(t)$  (packets/s);  $w_i(t)$  is the current window size and  $q_i(t)$  is the end-to-end loss probability. At the equilibrium point, window size adjustment  $w_i'(t) = 0$ , then from (1), the end-to-end equilibrium mark/drop probability feedback to source  $i$  is derived as following:

$$q_i^* = \frac{3}{2.w_i^{*2}} > 0 \quad (2)$$



**Fig. 1** Queuing delay of RED routers

Equation (2) implies that, at the equilibrium point, end-to-end mark/drop probability of source must be greater than zero. As a result, from the marking scheme of RED, it can be asserted that each router on the end-to-end always keeps a backlog. This consequently causes inevitable queuing delay for mice traffic such as HTTP requests to fly through. Figure 1 illustrates two RED routers which always maintain backlog at equilibrium point.

### 2.2 Queueing Delay at MaxNet Router

The marking mechanism of MaxNet router uses an explicit multi-bit signal instead of marking/dropping packet as RED router’s. Congestion price  $p_l$  at MaxNet router is defined in [SAW08]:

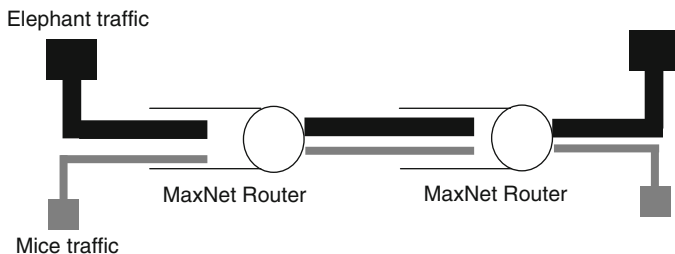
$$p_l(t + dt) = p_l(t) + dt \frac{y_l(t) - \mu_l C_l}{C_l} \tag{3}$$

where  $y_l(t)$  is the aggregated rate at link  $l$ ;  $C_l$  is the link capacity and  $\mu_l$  is the target link utilization. In MaxNet router, at the equilibrium point, the link price adjustment (3) tries to match the aggregated input rate  $y_l(t)$  with  $\mu_l C_l$ , leaving spare  $(1 - \mu_l)C_l$  capacity to absorb mice traffic and reduce the queuing delay.

Figure 2 illustrates the queuing delay of two MaxNet bottleneck links at the equilibrium point. In contrast to RED routers, there is no backlog in both two MaxNet routers when the target link utilization is set to  $\mu_l$  where  $0 < \mu_l < 1$ , hence, mice traffic can fly through links without being blocked.

## 3 Magnitude of Transient Queue of RED and MaxNet Routers

As pointed out in [FKS99, LAJ03], the weakness of RED is that it does not take into account the number of flows arriving a bottleneck link to have proper treatments to avoid heavy load. Assuming there are  $n$  flows sharing a bottleneck link. If a packet is



**Fig. 2** Queuing delay of MaxNet routers

marked or dropped then the offered load is reduced by a factor of  $(1 - 0.5n^{-1})$ . When  $n$  is large then  $(1 - 0.5n^{-1}) \rightarrow 1$ , which means the offered load will not be decreased and the queue length doesn't change, either. That means if RED is not configured aggressively then marking a single packet could result in simple “droptail” packet drops. The packet loss then severely declines the throughput of Reno sources due to their AIMD mechanism.

MaxNet router well controls the magnitude transient queue regardless the number of new arrival flows. At the equilibrium point, when the number of new arrival flows is small, transient queues exist, but the magnitude of these queues decreases rapidly as the number of flows increases [SAW08]. This can be explained with the following simple case: assuming that there are  $N$  flows sharing a bottleneck link. At the equilibrium point, each flow transmits at the rate of  $\frac{\mu_i C_i}{N}$ . Thus, when a new flow joins, its advertised rate is at most  $\frac{\mu_i C_i}{N}$ . The aggregated arrivals at router are at most  $\mu_l C_l + \frac{\mu_l C_l}{N}$ . Thus, this causes the overload:

$$0 \leq \textit{overload} \leq \left(1 + \frac{1}{N}\right)\mu_l - 1)C_l \tag{4}$$

Obviously, the larger the  $N$  is, the smaller the magnitude of transient queue becomes and eventually when  $N > \frac{\mu}{1-\mu}$ , the transient queue size drops to zero.

As mice traffic is short-lived flows, an effective congestion control should quickly controls the rate of such flows to avoid uncontrolled overshoot and transient queue. Unlike SlowStart mechanism of Reno [Jac88], MaxNet source employs MaxStart mechanism [SAW08] to seek for the target rate at initial state Fig. 3.

By adopting the multi-bit explicit signaling mechanism, MaxStart enables source to seek for its target rate within a significant short duration. New MaxNet flow is initiated at the minimum spare capacity of all links on its end-to-end path and then ramped up linearly to the advertised rate over two Round Trip Time [SAW08]. Therefore MaxNet source converges to the target rate more quickly than Reno source.

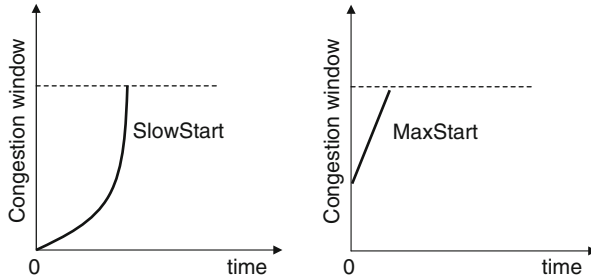


Fig. 3 SlowStart and MaxStart

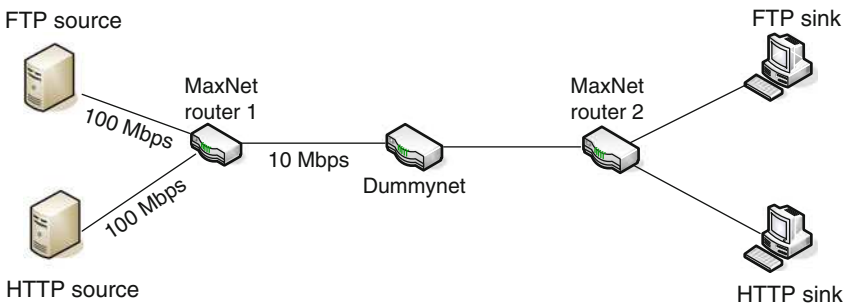


Fig. 4 MaxNet networks tested

## 4 Experiment and Evaluation

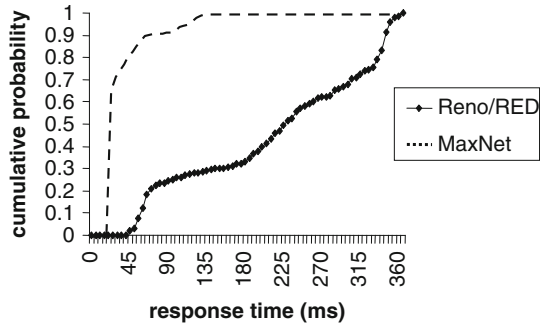
### 4.1 Testbed Layout

In this testbed, Pentium IV PCs (CPU 1.8 GHz, 256 MB RAM, 100 Mbps network cards) are used. The MaxNet Router1 is configured with the output capacity 10 Mbps to make sure it is the bottleneck link. The target utilization of both MaxNet routers ( $\mu_i$ ) is set to 0.95. Dummynet [DUM] is configured with 20 ms RTT delay.

The testbed of Reno/RED is same as MaxNet testbed where MaxNet routers are changed to RED routers. All of RED routers are configured with the RED parameters for web traffic [FKS99] as following:

- $w_q = 0.002$  : weighting factor for computing average queue size as suggested in [CH04] for web traffic.
- $q_{avg} = (1 - w_q).q_{avg} + w_q.q$  with  $q$  is instantaneous queue size.
- $min_{th} = 30$ : average queue length threshold for triggering probabilistic drops/marks.

**Fig. 5** Response time of TCP Reno vs MaxNet with 200 HTTP connections



- $max_{th} = 90$ : average queue length threshold for triggering forced drops/marks.
- $max_p = 0.1$ : maximum mark/drop probability.

We simulated elephant traffic and mice traffic with iperf [IPE] and httpperf [HTT] tools. For the both networks of Reno/RED and MaxNet, one long live FTP connection is generated at the “FTP source” in approximately of 60 s. Then 20 s after, HTTP connections are generated on the “HTTP source”. Each HTTP connection sends one request of 62 bytes and HTTP response size is 4 KB. The HTTP response time is computed at the application layer by the duration from the first byte being sent out to the time when the first byte of response received.

## 4.2 Response Time of HTTP Connections

We adopted the cumulative probability for statistical analysis of the response time of HTTP requests. In Fig. 5, the response time of HTTP requests in MaxNet is significantly less than in TCP Reno/RED. Particularly, in the experiment with 200 HTTP requests spawn, 100% of MaxNet HTTP requests receive the reply at most at 135ms while in TCP Reno/RED, only 30% of the total HTTP requests receive the first reply less than 135ms.

## 4.3 Throughput of Elephant Flow

In Figs. 6 and 7, packets drop occur at the RED bottleneck link, thus the throughput of Reno/RED is decreased. The greater the number of connections is, the more severity the drop becomes. In contrast, the throughput of elephant flow in MaxNet networks is not impacted regardless the number of arrival flows.

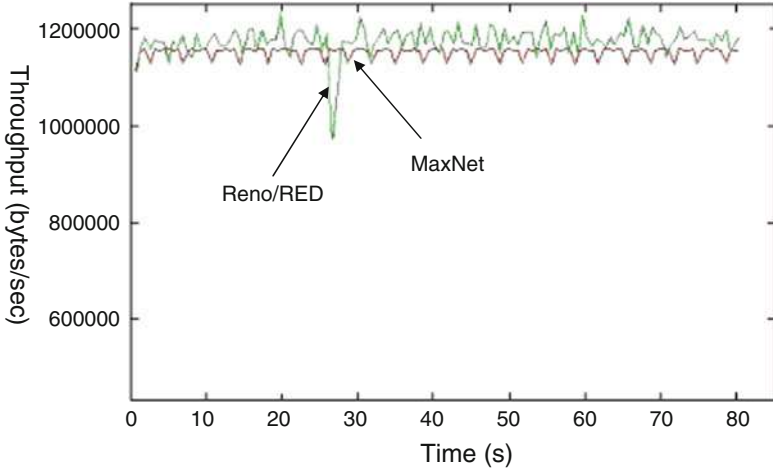


Fig. 6 Throughput of elephant traffic when 50 HTTP flows join

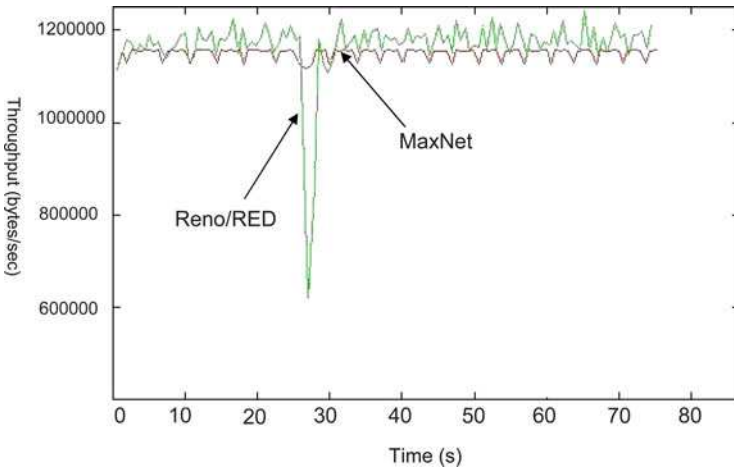


Fig. 7 Throughput of elephant traffic when 200 HTTP flows join

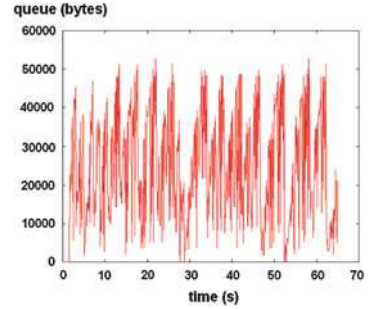
### 4.4 Transient Queue

In this section, we analyse the transient queue size in comparison between RED and MaxNet routers (Figs. 8 and 9).

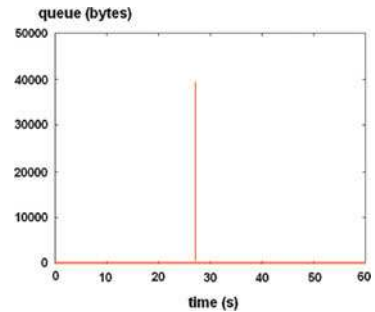
In this experiment, we configure two same 100Mbps links and keep the other configurations and parameters of MaxNet and RED router as the same as the above experiments to compare transient queue. Under MaxStart mechanism, the transient



**Fig. 8** Backlog at RED router



**Fig. 9** Backlog at MaxNet router



queue happens within short duration when HTTP connections join, meanwhile RED router always keep a backlog all the time.

## 5 Conclusions and Future Work

At the equilibrium point, MaxNet can clear the buffer while Reno/RED always keeps a backlog in routers. Therefore, when elephant traffic is mixed with mice traffic, MaxNet has a shorter response time for mice traffic than TCP Reno/RED. If the number of arrival mice flows is large, Reno without proper treatment can cause packet loss which in turn eventually degrades the throughput of elephant traffic. In addition, MaxStart mechanism of MaxNet (using multi-bit signaling) can control mice flows to the target rate more quickly than Reno sources. By the experiments, we showed that the performance of mice and elephant traffic when using with MaxNet is better than with TCP Reno/RED's in terms of response time and network utilization. For future work, more experiments should be conducted with realistic web workload and other performance properties such as fairness, TCP friendly should be evaluated.

**Acknowledgements** The authors would like to thank Bartek Wydrowski, Lachlan Andrew and MaxNet team [TEAM] for their advices and support.

## References

- [BLM03] Bartek Wyrowski, Lachlan L.H. Andrew, Moshe Zukerman, “MaxNet: A Congestion Control Architecture for Scalable Networks”, IEEE Communications Letters, Oct. 2003.
- [CH04] Claypool, M. Kinicki, R. Hartling, M., “Active Queue Management for Web traffic”, Proceedings of IPCCC04, 2004.
- [DUM] <http://info.iet.unipi.it/~luigi/ip.-dummynet/>.
- [FJ93] Sally Floyd and Van Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, ACM Transactions on Networking, 1993.
- [FKS99] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha and Kang G. Shin, “A Self-Configuring RED Gateway”, Proceedings of INFOCOM’99, March 1999.
- [HTT] <http://www.hpl.hp.com/research/linux/httpperf/>.
- [IPE] <http://noc.pregi.net/iperf.html>.
- [Jac88] Van Jacobson, “Congestion Avoidance and Control”, in Proceeding of SIGCOMM’88, August 1988.
- [Jac90] Van Jacobson, “Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno”, in Proceeding 18th Internet Engineering Task Force, August 1990.
- [LAJ03] Long Le, Jay Aikat, Kevin Jeffay, F. Donelson Smith, “The Effects of Active Queue Management on Web Performance”, Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM’03.
- [MRB05] Martin Suchara, Ryan Witt and Bartek Wyrowski, “TCP MaxNet - Implementation and Experiments on the WAN in Lab” in Proceedings of IEEE International conference on Networks (ICON), 2005.
- [SAW08] M. Suchara, L. Andrew, R. Witt, K. Jacobsson, B. Wyrowski and S. Low, “Implementation of Provably Stable MaxNet” Proceedings of BROADNETS, September 2008.
- [TEAM] <http://netlab.caltech.edu/maxnet/>.
- [WJL06] D.X. Wei, Cheng Jin, Steven H. Low and Sanjay Hegde, “FAST TCP: motivation, architecture, algorithms, performance”, IEEE/ACM Transactions on Networking, 14(6):1246-1259, Dec 2006.