Failure of TCP Congestion Control under Diversity Routing

Yupeng Zhang
Department of Information Engineering
The Chinese University of Hong Kong
E-mail: s0760929@cuhk.edu.hk

John Chapin, Vincent W.S. Chan, *Fellow IEEE*
Claude E. Shannon Communication and Network Group
Research Laboratory of Electronics
Massachusetts Institute of Technology

Abstract [1] — **TCP does not perform well in networks with stochastic channels, with links that randomly drop packets or have long outages. Diversity routing has been proposed to improve TCP's performance in these networks. In diversity routing, a sublayer between TCP and the network replicates each transmitted packet and sends the multiple copies along parallel paths. As long as at least one of the copies reaches the receiver, TCP considers the transmission successful and maintains high throughput. Previous investigations of diversity routing have analyzed TCP's performance when there is a single flow and no congestion. In this paper, we analyze the performance of multiple flows in a congested network. The contribution of this work is the discovery of three adverse effects that occur when TCP is combined with diversity routing. The effects are: link capacity overflow, rate unfairness, and lock-out of late-arriving flows. Simulation is used to analyze the root cause of these effects. Our conclusion is that naïve diversity routing fundamentally breaks the TCP congestion control mechanism and cannot be used for TCP performance improvement in networks with stochastic channels, such as many wireless and satellite networks. We propose an improvement to diversity routing that may overcome these problems, enabling use of unmodified TCP.**

Keywords-transmission control protocol; congestion control; stochastic networks; diversity routing; network simulation

I. INTRODUCTION

As the Internet continues to grow towards a universal infrastructure, the next generation of networks will be heterogeneous, combining traditional wired networks with new modalities such as mobile wireless and satellite networks. A difficult problem is how to make traditional network protocols work well in these new types of networks. At Layer 4, Transmission Control Protocol (TCP) is one of the most important protocols for the current Internet. It provides end-to-end reliability and transmission rate control. However, TCP suffers performance degradation in the new networks, where there are random packet losses and link interruptions due to physical layer effects. We call such links "stochastic links" in this paper. As is well known, TCP treats all, including stochastic, packet loss as congestion and responds by reducing transmission rate. When combined with the long end-to-end delays of satellite or multihop wireless networks, this effect reduces application throughput to a small fraction of available capacity.

To improve TCP's performance in networks with stochastic links, one approach that has been proposed is diversity routing. The strategy is to replicate packets and send them on multiple links in parallel. This may be done by the network layer or by a sublayer between the transport and network layers. The first copy that reaches the receiver is delivered; later copies are discarded. This approach masks packet loss due to physical layer effects from higher layers.[2]

Our work is focused on how TCP performs when combined with the diversity routing strategy proposed in prior work from our group [2]. We consider in particular what happens when the network is congested and there are many flows. Our results show significant performance degradations and suggest a direction for further protocol development.

There has been substantial previous work on diversity routing, some variants of which are called "path diversification" by various authors. Prior work includes efficiency optimization, QoS improvement, delay minimization and security improvements. Most work on diversity routing considers network layer effects in isolation, without considering how TCP behaves when combined with diversity routing. Projects that have analyzed or simulated TCP and diversity routing together have usually considered only uncongested situations with a single flow. We are aware of only two projects that studied TCP congestion control under diversity routing [3], [4].

[3] proposes a horizon algorithm for load balancing and multi-path routing in wireless mesh networks, based on TCP. In [3], congestion control and fairness are achieved through establishing a utility function that determines the optimal TCP window size for each sender (the window size is the TCP internal variable that controls transmission rate). When the window size exceeds the optimal value, the sender is signaled to reduce its window size. While this is a useful direction for protocol development, [3] did not consider the effect of link dropouts, or how to differentiate congestion and link dropouts so congestion can be treated properly.

[1] Research performed while first author was a Charles Kao Visiting Undergraduate Student Fellow at MIT during summer of 2010

[2] One variant of this method is to apply error correction codes to packets and send fragments down individual paths. Packets will be successfully delivered provided enough fragments are received, reducing resource use compared to making a copy. This change does not significantly affect our results.

[4] studied the same problem that we consider in this paper. Their term for diversity routing is Multi-Radio Transmission Diversity (MRTD) With Redundancy. They simulate the performance of TCP under diversity routing in networks with stochastic links, but there are only two flows in the network and the capacity of the links is high enough that congestion does not occur. The focus of [4] is improvement of throughput and reduction of retransmission rate. In contrast, we focus on TCP with diversity routing when congestion control is triggered.

In our work, we use simulation to observe three adverse phenomena of TCP congestion control under diversity routing. The combined rate of all flows overflows link capacity for long periods of time, which may cause the network to be unstable. Flows that reduce their window size first during congested periods do not receive their fair share of link capacity. Flows that start up after the network is already congested are locked out. We use analysis and simulation to identify the root cause of these effects. Our conclusion is that diversity routing fundamentally breaks TCP congestion control. Therefore, diversity routing cannot be used for TCP performance improvement in networks with stochastic channels, such as many wireless and satellite networks, without significant modifications to improve performance under congestion.

Section 2 gives intuitive explanations of the three ways TCP congestion control fails under diversity routing. Section 3 provides a brief mathematical analysis and shows that the problem worsens as the number of links increases. Section 4 presents the simulation setup, validation, and results. We discuss the simulation results and assumptions in Section 5. Finally, we conclude the paper with suggestions for modifications to diversity routing in Section 6.

II. INTUITIVE EXPLANATION OF ADVERSE PHENOMENA

First, we briefly review the TCP congestion control algorithm. TCP controls transmission rate by limiting the number of packets sent per Round Trip Time (RTT), which is the measured delay from sending a packet until receiving the corresponding acknowledgment (ACK). The window size of the sender is the number of packets that the sender sends to the receiver in one RTT. Window size starts from one packet per RTT and initially increases exponentially with each RTT in which all packets are delivered and ACKed. The exponential growth period is called slow start. After the window size reaches a limit called the slow start threshold, further window size increase is linear up to an upper limit called the maximum window size. The linear growth period is called congestion avoidance. When packet loss occurs, TCP considers it to indicate network congestion and reduces the window size to half of its previous value or all the way to one, depending on the situation. This algorithm prevents capacity overflows because senders begin reducing their rates as soon as congestion causes routers to start dropping packets. This algorithm achieves rough fairness under the assumption that the probability of packet loss for a flow is proportional to the share of link capacity consumed by that flow and all the RTTs are comparable.

TCP does not work well in networks with stochastic links because it interprets packet loss as congestion and reduces the sender's window size. If the packet loss is due to random physical layer effects, the correct response is to maintain the transmission rate. Diversity routing reduces the probability that TCP will incorrectly reduce transmission rate, by sending multiple copies of the packet along separate links. TCP only notices packet loss and reduces window size if all copies of the packet are dropped.

TCP with diversity routing has a bad side effect: congestion on one link leading to packet loss on that link does not necessarily cause senders to reduce their transmission rate over that link. Even if there is congestion on all the parallel links, a sender will only reduce its rate if all links happen to drop the same packet. An example is illustrated in Figure 1. TCP sends five transport layer packets. The diversity routing sublayer under TCP makes two copies of each packet in order to exploit two parallel links. Assume both of the links are congested. One link drops packet number 2 and the other drops packet numbers 4 and 5. The diversity routing sublayer in the receiver combines the copies from the two links and delivers all the packets successfully to TCP at the transport layer. As a result, TCP fails to recognize that congestion exists. The sender keeps increasing its window size even though the link capacities have already been reached.

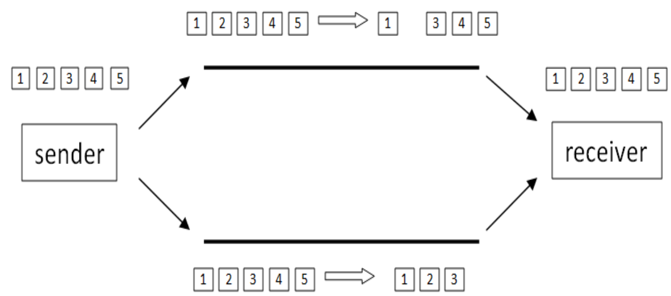

Fig. 1. Example of diversity routing side effect

This effect leads to three phenomena. First, the total packets sent in each RTT may exceed network capacity for a long period of time, because flows keep increasing their transmission rate even though there is congestion. With excess packets transmitted for so long, router queues upstream from the bottleneck link all reach maximum length, leading to high end-to-end delay. A router trying to avoid this problem with Random Early Discard or a similar algorithm will have no effect on sender rates, since copies delivered via other links mask the discarded packets.

The second problem is that once a flow loses the same packet on all parallel links, causing TCP to reduce its window size, it can take a long time for the flow's transmission rate to increase back to parity. The available network bandwidth is consumed by other flows that never slow down. More precisely, when congestion occurs, it is likely that only a few flows observe packet loss and reduce their rates while the others in the network continue increasing their window size. Thus, more often than not, most flows keep increasing window size by one each RTT. The flow or flows that slowed down first will have a lower rate/s until all the flows at the higher rate happen to suffer the same packet lost on all parallel links, which is rare. The result is a period of unfair throughput.

The third problem is closely related to the unfairness effect. A flow that starts up when the network is already congested starts with a small window size. In TCP over a single congested link, most other flows will observe packet loss and reduce their window size relatively soon, enabling the new flow to acquire a fair share of the available capacity. Under diversity routing, it takes a much longer time for other users to reduce their transmission rates, even after congestion develops. As a result the new flow can observe an extended period with unfairly low throughput.

III. ANALYSIS

Consider the case where all users use the same group of parallel paths. We use m_i to denote the number of packets the ith user sends in one RTT, measured at the point where TCP sends packets to the diversity routing sublayer. We use n to denote the number of paths. Because each packet is replicated over all paths, the ith user sends m_i packets on each path. Assume packets are dropped independently over each path and the probability of drop is p. The probability P_i that TCP of the ith user observes at least one packet loss is the probability of all paths dropping the same packet for any of the m_i packets the ith user sent. This can be approximated as

$$P_i \approx m_i p^n$$

where the Union Bound has been invoked to approximate the exact expression which is easily found but tedious.

This analysis shows that the probability of flows reducing their window size decreases as n increases and thus the expected capacity overflow and unfairness problems worsen.

IV. SIMULATION

A. Description of simulation program

The simulation analyzes the performance of TCP under diversity routing with multiple flows in the system. We make several assumptions to simplify the model. First, we assume each link has the same capacity R_{max}, and each link has independent packet loss and failure. This assumption usually holds in the case of mesh networks, where there are multiple similar disjoint paths from sender to receiver. The assumption does not hold in a heterogeneous network such as simultaneous use of a wireless WAN and a wireless LAN. This case is deferred to future work.

The second assumption is that each packet has the same size K. In reality most of the packets transmitted after establishing a TCP connection are full sized segments. Thus this assumption is quite accurate as long as flows are not unusually short.

We assume that physical layer fading is total: capacity drops to 0 when a fade occurs, rather than to some rate lower than the specified maximum. In the same manner, we assume the sender's window size is reduced to 1 whenever TCP observes packet loss. This assumption reflects links where the outages are long enough to drop multiple packets from the same user. In a wider class of links TCP sometimes reduces the window size by half and sometimes all the way to one, depending on conditions. Section 5 discusses the effect of these assumptions on our results.

We assume all packets sent in the last TCP window before a link drops out are lost. The number of packets lost only slightly affects the number of successfully transmitted packets, but won't affect the window size of flows because the packet loss has already happened.

We assume the window size of users grows linearly from 1. That is, we do not model the exponential growth of the slow start phase of TCP's window control algorithm. Section 5 analyzes this assumption and shows that it does not affect the conclusions drawn from the simulation.

Finally, we define the maximum window size of users to be $W_m = RTT * R_{max}/K$, which means the only limitation of window size is link capacity. This assumption highlights the effects of congestion. In current networks there are limits on maximum window size. However, for heavily congested networks the maximum window size is never reached. Thus, assuming unlimited window size does not affect the analysis done in this paper, which focuses on congested situations.

Each link can be modeled by a two state Markov process [2]. In the Up state, the link is good and supports transmission at normal capacity. In the down state, the link is faded and the capacity is assumed to be 0. Parameter γ represents the state transition rate from state up to down, and ν denotes the rate from down to up. Thus the expected time that the link is in state up and state down are respectively $\frac{1}{\gamma}$ and $\frac{1}{\nu}$. The continuous-time Markov process is approximated in the simulation by iterating a discrete time process over intervals much smaller than a RTT.

The simulation is executed for 500000 RTTs for each configuration of parameters. We validated that this duration is long enough by running a particular test configuration 100 times and observing that the difference in results is extremely small. Also when $RTT = 0.1s$, which was measured in wireless networks in [2] and [7], and is already a short RTT compared to the range studied in this project, 500000 RTTs is around 14 hours. This is a long enough duration to observe stable results with the effects of randomness and link variation smoothed out.

B. Validation of simulation

We checked our simulation program by comparing the throughput it computes to the analytical results presented in [2], which reports the throughput of TCP under diversity routing with a single user in the system. In our simulation, the range of up durations considered is 1s to 100s and the range of down durations considered is 0.01 second to 1 second. This range is consistent with measurements of wireless and satellite networks. Situations where the links are extremely bad and thus mostly down are excluded from this study. In this range of parameters, there is a close match to the analytical results.[3]

As a further consistency check, we compare our simulation output to the results of [5], which studies non-diversity TCP over networks with stochastic links. The model used in [5] incorporates buffer size limitations of the link and the receiver, which are not modeled in our simulation. We can still execute our simulation under parameters comparable to those used in their study. The parameter $E[x]$ in Figure 2 is the expected time of one renewal process defined in [5]. We represent it by $\frac{1}{\gamma}+\frac{1}{\nu}$ in our work. The throughput is defined as the total successful transmission rate of the user divided by the link capacity. The absolute value from our simulation is close to the results from Figure 6 in [5]. The difference is due to the effect of buffer size which is not modeled in our simulation. The results of our simulation are close enough to the prior results to provide a decent level of confidence in the simulation's correctness.

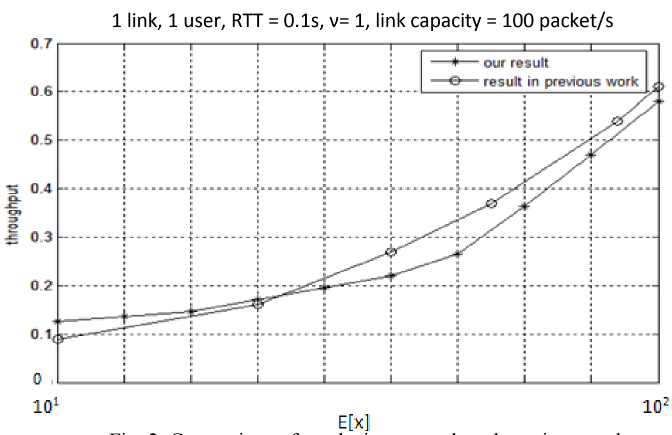

Fig. 2. Comparison of results in our work and previous work

C. Simulation results

This section presents the simulation results that show in detail the three effects described earlier. To study capacity overflow, we measure the total packets sent in each RTT by all users and compare to the link capacity. We also integrate the total packets sent over time to see whether the capacity is exceeded when averaged over long periods. If this occurs, the problems previously described such as high latency due to long router queues will occur.

To study rate unfairness, we record the total number of packets successfully transmitted by each flow since the start of the simulation. Because all the flows experience the same conditions, if the total packets transmitted over time vary significantly, the technique is deemed unfair. Finally, to study lock-out of late-arriving flows, we delay the startup of one flow for long enough that the other flows have stabilized and fully occupy the available bandwidth. After starting the last flow, we record its window size over time and compare it to the average value of the other flows.

1) Overflow.

We execute the simulation with 5, 10, 20, 50 and 100 flows sharing the network. This enables the evaluation of any effects that become stronger or weaker as the number of flows changes. The parameters chosen for these simulation runs match those measured for satellite networks [1, 5].

In Figures 3-5, the y axis is the total number of packets all flows send in each RTT. Figure 3 shows that on a single link, once the total number of packets sent by all flows exceeds capacity, enough flows immediately reduce their transmission rate that capacity is only exceeded for one RTT. However, when using diversity routing over 4 links (Fig 4), capacity is exceeded for many RTTs. As the number of flows increases, the overflow increases. With 100 flows sharing the network (Fig 5), the excess per RTT reaches 10% of link capacity. Although the number of packets sent per RTT fluctuates below the capacity limit, it never stabilizes below the limit.

To look for longer term effects, we integrate the rate over time. When there are 5 flows over 4 links, the total packets sent over 2000 RTTs is just less than capacity (8.7×10^5 compared to 10^6). However, when there are 100 flows, the total packets sent exceeds capacity (2.12×10^7 compared to 2×10^7). In this case the queues in the routers will always be full and overflow problems may occur. The two cases are different because the probability of a flow observing packet loss and reducing its window size is directly proportional to how many of the total packets sent in each RTT are from that flow. Most realistic networks have many more than 5 flows so sustained overflow is likely to happen.

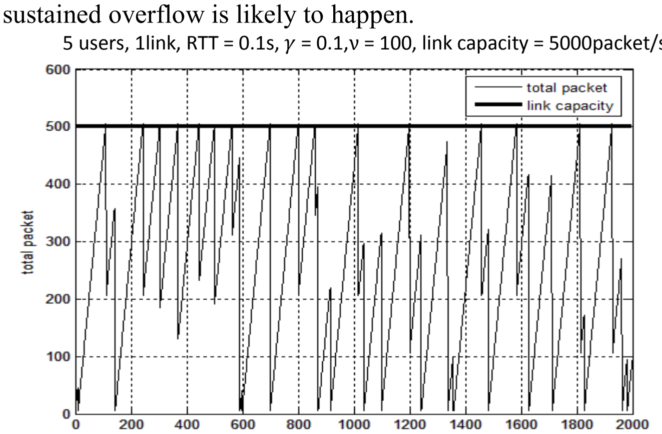

Fig. 3. Total packets per round trip time on a single link

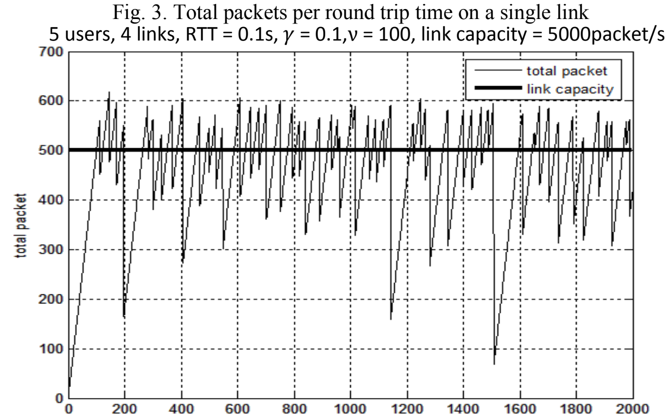

Fig. 4. Overflow of capacity under diversity routing (5 users)

[3] Chart omitted due to space constraints.

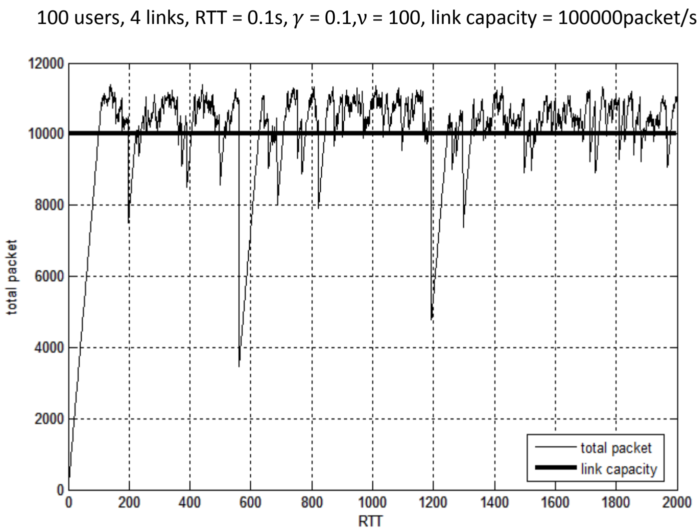

Fig. 5. Overflow of capacity under diversity routing (100 users)

2) *Unfairness*

The graphs in Figures 6 and 7 represent the total packets sent by each flow since the start of the simulation. When there are five flows, they observe similar total throughput over one link, but quite different throughputs over diversity routing, even after a long period of 2000 RTTs. Moreover, unfairness increases when capacity increases because it takes longer for the flow that first reduced its window size to increase back to parity. Although in the long run on the order of 10000 RTTs, the total number of packets sent by each user is basically the same, most application transfers would not last that long.

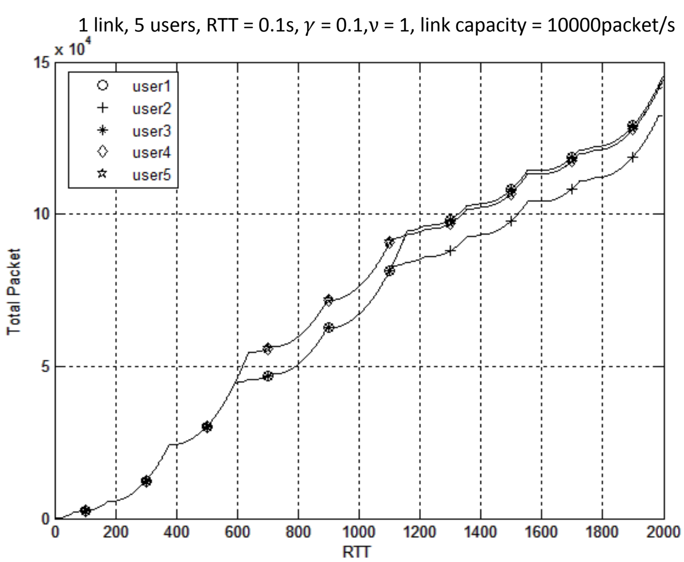

Fig. 6. Total packets each user sends (single link 5 users)

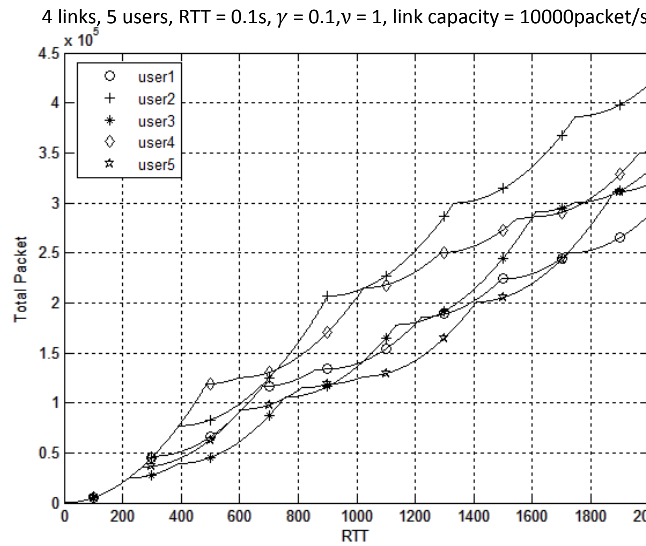

Fig. 7. Total packets each user sends (multiple links 5 users)

3) *Lock-out*

We simulate 99 flows in the network running for 8000 RTTs. This is long enough for them to stabilize consuming most available capacity. We then let the 100th flow start up. Figure 8 shows the window size of the 100th flow compared to the average window size of the other 99 flows. We can see that in this period of 500 RTTs, the throughput of the last user is only half of the average level. This is extremely unfair to the late-arriving user. Running this configuration 100 times with different random seeds, the unfair case occurs 30% of the time. Even though there is a 70% chance of no lock-out of late arriving flows, we regard the 30% chance of lock out as unacceptable.

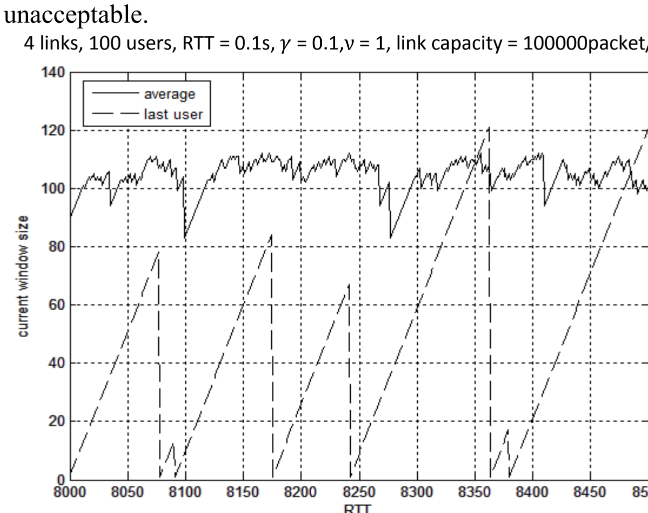

Fig. 8. Last user entering at 8000th RTT

V. DISCUSSION

The well-known problem of TCP over networks with stochastic links is its inability to differentiate congestion and link dropouts. Diversity routing sends replicated packets to mask link dropouts and thus improves the throughput of TCP. However, it also masks packet loss caused by congestion. In other words, traditional TCP treats link dropouts as congestion, while TCP under diversity routing is likely to treat congestion as a link dropout. Responding in a way appropriate for link dropouts, TCP does not reduce its transmission rate when congestion develops. This leads to the three bad effects we described and observed.

We now consider whether our results are affected by the assumptions used to simplify the simulation. We assumed that window size always reduces to one when packet loss is detected. If we had more accurately modeled TCP by reducing to half of the previous window size in some cases, the overflow effect we observed would have been even larger. When considering unfairness and lock-out, dropping always to one or only to half of the previous window affects flows currently at a high rate and flows currently at a low rate equally, so the ratios we observed would be similar to the behavior of the full TCP algorithm.

We assumed that window size increases linearly from one, not exponentially up to the slow start threshold as TCP actually does. In practice when congestion occurs, flows will be in congestion avoidance rather than the slow start mode. Therefore, linear increase is a good approximation of the dynamics of the real case. The most important difference is that actual TCP takes less time to reach full rate. However, the slow start threshold is always half of the window size at which packet loss was detected. Thus actual TCP spends at least half its time increasing window size linearly. This shortening of time scale does not affect the overflow phenomenon at all, and it also does not change the rough duration of the period of unfairness and lock-out of late arrivals we measure in the simulation. So our results are still valid under this assumption.

VI. CONCLUSION AND FUTURE WORK

This paper has reported an analysis of congestion under the diversity routing strategy proposed in previous work for improving the performance of TCP over stochastic links. We observed three adverse effects: capacity overflow, rate unfairness, and lock-out of late arriving flows. We conclude that TCP congestion control fails when TCP is combined with naïve diversity routing.

The simulation results suggest a potential path forwards. When congestion occurs, a flow observes packet loss on many links at the same time. Because senders keep increasing window size even though the links are congested, the same set of links will still have packet loss over the next several RTTs. On the other hand, if the packet loss is due to link dropouts, each flow sees packet loss on only a few links, and the links where loss occurs are normally different for each RTT. This is a useful signature to differentiate the two cases. Furthermore, the simulation results suggest that the number of links experiencing packet loss when congestion occurs is always larger than the number if the cause is random dropouts, because the two effects are cumulative.

It will be valuable to investigate exploiting this information. When the diversity routing sublayer observes the congestion signature, it can set the Explicit Congestion Notification bit in the TCP header of packets it delivers, triggering rate reduction. When it observes the signature of stochastic packet loss, it delivers the packets without ECN and TCP will continue to increase the transmission rate. This approach may allow diversity routing and TCP to be successfully combined without modification to TCP.

Our simulation assumed that all flows are replicated across the same set of links. Further study is needed to determine if the method we propose to differentiate congestion from random link dropouts will work in cases where flows exploit different but overlapping link sets, or where some or the majority of flows are not using diversity routing.

REFERENCES

[1] V.W.S. Chan and A. H. Chan, "Reliable Message Delivery Via Unreliable Networks," 1997 IEEE International Symposium on Information Theory, (MS-12104) Ulm, Germany, 29 June - 4 July 1997

[2] Yinuo Qian, "Effects of Diversity Routing on TCP Performance in Networks with Stochastic Channels," Master Thesis of Massachusetts Institute of Technology, unpublished.

[3] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: Balancing tcp over multiple paths in wireless mesh networks," Technical report, Microsoft Research, 2008.

[4] G.P. Koudouridis, A. Yaver and M.U. Khattak, "Performance Evaluation of Multi-Radio Transmission Diversity for TCP Flows," Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th, vol., no., pp.1-5, 26-29 April 2009

[5] A.A. Abouzeid, S. Roy and M. Azizoglu, "Stochastic Modeling of TCP over Lossy Links," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol.3, no., pp.1724-1733 vol.3, 26-30 Mar 2000

[6] J.P. Choi and V.W.S. Chan, "Adaptive communications over fading satellite channels," Communications, 2001. ICC 2001. IEEE International Conference on, vol.9, no., pp.2635-2639 vol.9, 2001

[7] Qinqing Zhang and S.A. Kassam, "Finite-state Markov model for Rayleigh fading channels," Communications, IEEE Transactions on, vol.47, no.11, pp.1688-1692, Nov 1999

[8] S. Gheorghiu, A.L. Toledo and P. Rodriguez, "Multipath TCP with Network Coding for Wireless Mesh Networks," Communications (ICC), 2010 IEEE International Conference on, vol., no., pp.1-5, 23-27 May 2010

[9] J.P. Rohrer, A. Jabbar and J.P.G. Sterbenz, "Path diversification: A multipath resilience mechanism," Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on, vol., no., pp.343-351, 25-28 Oct. 2009

[10] M. Alasti, K. Sayrafian-Pour, A. Ephremides and N. Farvardin, "Multiple description coding in networks with congestion problem," Information Theory, IEEE Transactions on, vol.47, no.3, pp.891-902, Mar 2001

[11] Hsu-Feng Hsiao, A. Chindapol, J.A. Ritcey, Yaw-Chung Chen and Jenq-Neng Hwang, "A new multimedia packet loss classification algorithm for congestion control over wired/wireless channels," Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, vol.2, no., pp. ii/1105- ii/1108 Vol. 2, 18-23 March 2005

[12] B. Sardar and D. Saha, "A survey of tcp enhancements for last-hop wireless networks," Communications Surveys & Tutorials, IEEE, vol.8, no.3, pp.20-34, 3rd Qtr. 2006

[13] B. Yahya and J. Ben-Othman, "REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks," Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, vol., no., pp.1-7, Nov. 30 2009-Dec. 4 2009