Study For Congestion Control in Mobile Adhoc Networks

K.Thirunadana Sikamani
HOD, CSE Dept
St.Peters University.
Chennai

N.Kirubananda Sarathy
Asst Prof, ECE
St.Peters University
Chennai

M.Rajmohan
Asst Prof, ECE
Hindustan University
Chennai

Abstract

In recent years a number of papers have presented theoretical solutions to this problem that are based on combining differential-backlog scheduling algorithms with Utility-based congestion control .Reducing packet loss in MANETs typically involves congestion control running on top of a mobility and failure adaptive routing protocol at the network layer. In the current designs, routing is not congestion-adaptive. Routing may let a congestion happen, which is detected by congestion control, but, to deal with this fact, it may be too late (i.e., long delay and many packets already lost) or require significant overhead if a new route is needed. This problem becomes more visible especially in large-scale transmission of high traffic such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is more of significance.

1 Introduction

Recent years have seen a stream of TCP-friendly congestion control mechanisms designed for the Internet. They are driven by the need of multimedia streaming over the network, which requires smooth rate adaptation, instead of TCP's abrupt "cut-half" rate change policy. At the same time, they attempt to maintain long-term throughput fairness with other competing TCP flows in the network, i.e., their long-term throughput should approximately equal to that of a TCP flow under the same network condition. Among the class of TCP-friendly congestion control mechanisms, the TCP equation-based approach has been one of the most well studied algorithm . It relies on a "TCP throughput equation" which captures the TCP throughput over a network path with certain loss rate and roundtrip time (RTT). Past studies have shown that the TCP equation is able to achieve reasonable fairness with competing TCP flows under a wide range of traffic conditions in wire line networks . Real experiments over the Internet also suggest that it is safe to be deployed . In fact, the protocol that implements the TCP-equation based approach, TFRC (TCP Friendly Rate Control), has recently become a standard RFC . Now we shift our attention from Internet to a mobile ad hoc network (MANET). In MANET, each node is free to move about, creating not only fluctuating wireless link bandwidth, but also link breakage, route breakage and dynamic routing. Currently TCP remains the de facto standard for congestion control in MANET (despite its many well-known deficiencies in this environment), simply because of its wide acceptance and deployment over the Internet. With the emerging need of multimedia streaming over MANET, equation-based congestion control is likely to find its way into MANET as well, for example, by reusing the same software that has been developed for the Internet. However, the behavior of equation-based congestion control (TFRC) is very much unknown in MANET where the degrees of network dynamics are far more diverse than those in wire line networks. For instance, wireless link's bandwidth can vary greatly in very small timescale, due to the randomness in channel contention and signal fading. Packet loss can occur due to congestion related queuing loss, wireless-related random loss, and mobility-related routing loss. Under this environment, it is unclear whether TFRC will be able to compete fairly with TCP, and if not, what are the factors that contribute to such behavior. In this paper, we study the behavior of TFRC in MANET. Our finding indicates that, while TFRC is able to maintain smooth rate change, its throughput is often "beaten" down by competing TCP flows to a certain degree, especially under heavy background traffic and dynamic topology conditions. To explain TFRC's conservative behavior, we analyze several factors including loss rate discrepancy, inaccuracy of loss rate prediction, and lack of auto-correlation in MANET's loss process. We also explore TFRC's response to the tuning of its loss event interval estimator, and show that its conservative behavior cannot be completely corrected. Our study shows the limitations of applying TFRC to the MANET domain, and reveals some fundamental difficulties in doing so. Our findings in this paper also open up the question of how to properly perform multimedia streaming over MANET. To this end, we propose an alternative scheme (called EXACT-AA) based on router's explicit rate signaling and application's adaptation policies. We demonstrate the feasibility of our scheme using an audio streaming application over a real MANET test-bed.

2. BEHAVIOR OF IN MANET

In this section we study the behavior of TFRC in terms of long-term and short-term fairness and smoothness, under various static and dynamic MANET topologies and with different levels of background traffic. *A.* Simulation Network and Parameters We consider two types of MANET topologies: static and dynamic. In static topology, we consider a chain that consists of 2 to 7 stationary nodes, which provides a controlled environment where TFRC can be evaluated over a path with increasing number of hops. In dynamic topology, two scenarios are considered: a small 600 S 600m network with 50 nodes (where a path has 1 to 4 hops), and a larger 1500 S 300m network with 60 nodes (where a path has 1 to 7 hops). In both scenarios, random way-point mobility is used with maximum speed of 10 m/s and pause time of 0 seconds, and the network is not partitioned at any time. We hope to use these scenarios (6 static and 2 dynamic) to represent the spectrum of MANET topologies. In each scenario, 10 TCP-SACK flows and 10 TFRC flows are created to compete with each other over the same path. 2 In the static chain scenarios, TCP and TFRC flows run from one end of the chain to the other. In the dynamic scenarios, a pair of nodes are randomly chosen to be the sender and receiver of the TCP and TFRC flows. Since they travel through the same path, they should encounter the same network conditions. Sharing a path also shields the potential discrepancy of route discovery for different paths. We use Dynamic Source Routing (DSR) as the underlying routing protocol.There are many existing studies in enhancing TCP performance in MANET, e.g. TCP-ELFN . Similar techniques may be applied to TFRC as well. In this paper, we only focus on the behaviors of unmodified TCP and TFRC flows. Background traffic consists of non-adaptive CBR flows to create consistent but varying levels of congestion within the network. In the chain scenarios, a CBR flow is created with various data rates, from one end of the chain to the other. In the dynamic scenarios, in order to spread out the background traffic across the network, 10 CBR flows are created each between a pair of randomly selected nodes. In order to avoid stalling the TCP and TFRC flows, we have carefully selected different levels of

CBR data rates for each of the simulated scenarios, such that the non-adaptive CBR traffic does not over- flood the whole network. We keep most of TFRC's default settings in the ns- 2 (2.1b9a) simulator, which mostly corresponds to the parameters suggested in . We use the same data packet size (1000 bytes) for TCP and TFRC, so that we can also compare their throughputs by the number of data packets. Each simulation run lasts for 1000 seconds.

3. CONGESTION ADAPTIVE ROUTING (CRP)

In CRP, every node appearing on a route warns its previous node when prone to be congested. The previous node then uses a "bypass" route bypassing the potential congestion to the first non-congested node on the route. Traffic will be split probabilistically over these two routes, primary and bypass, thus effectively lessening the chance of congestion occurrence. CRP is on-demand and consists of the following components: (1) Congestion monitoring, (2) Primary route discovery, (3)Bypass discovery, (4) Traffic splitting and congestion adaptivity, (5) Multi-path minimization, and (6) Failure recovery.

A. Congestion Monitoring

A variety of metrics can be used for a node to monitor congestion status. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion. Any of these methods can work with CRP in practice. We further classify the congestion status at a node into 3 levels:"green", "yellow", and "red". A node is said to be "green"if it is far from congested, "yellow" if likely congested, or "red" if most likely or already congested. As later discussed, a bypass is a path from a node to its *next green node*. The next green node is the first green node at least two hops away downstream on the primary route.

B. Primary Route Discovery

To find a route to the receiver, the sender broadcasts a REQ packet toward the receiver. The receiver responds to the first copy of REQ by sending toward the sender a REP packet. The REP will traverse back the path that the REQ previously followed.

This path becomes the *primary route* between the sender and the receiver. Nodes along this route are called *primary nodes*. To reduce traffic due to route discovery and better deal with congestion in the network, we employ two strategies: (1) the REQ is dropped if arriving at a node already having a route to the destination, and (2) the REQ is dropped if arriving at a node with a ”red” congestion status.

C. Bypass Discovery

A node periodically broadcasts to neighbors a UDT (update) packet. This packet contains this node’s congestion status and a set of tuples *fdestination R, next green node G, distance to green node mg*, each for a destination *R* that the node has a route to. The purpose is that when a node *N* receives a UDT packet from its next primary node *Nnext* regarding destination *R, N* will be aware of the congestion status of *Nnext* and learn that the next green node is *G* which is *m* hops away on the primary route. If *Nnext* is yellow or red, a congestion is likely ahead if data packets continue to be forwarded on link *N ! Nnext*. Since CRP tries to avoid congestion from occurring in the first place, *N* starts to discover a bypass route toward node *G* - the next green node of *N* known from the UDT packet. This bypass search is similar to primary route search, except that: (1) the bypass request packet’s TTL is set to 2 £ *m*, and (2) the bypass request is dropped if arriving at a node (neither *N* nor *G*) already present on the primary route. Thus, it is not costly to find a bypass and the bypass is disjoint with the primary route, except that they join at the end nodes *N* and *G*. It is possible that no bypass is found due to the way the bypass request approaches *G*. In which case, we continue using the primary route. However, [1] finds that the chance for a “short-cut” to exist from a node to another on a route is significant.

D. Traffic Splitting and Congestion Adaptability

At each node that has a bypass, the probability *p* to forward data on the primary link is initially set to 1 (i.e., no data is sent along the bypass). It is then modified periodically based on the congestion status of the next primary node and the bypass route (see Table I). The congestion status of the bypass is the accumulative status of every bypass nodes. The key is that we should increase the amount of traffic on the primary link if the primary link leads to a less congested node and reduce otherwise.

E. Multi-path Minimization

To reduce the protocol overhead, CRP tries to minimize using multiple paths. If the probability *p* to forward data on a primary link approaches 1.0, this means the next primary node is far from congested or the bypass route is highly congested. In this case, the bypass at the current node is removed. Similarly, if the next primary node is very congested (p approaches 0), the primary link is disconnected and the bypass route becomes primary. To make the protocol more lightweight, CRP does not allow a node to have more than one bypass. The protocol overhead due to using bypass is also reduced partly because of short bypass lengths. Each bypass connects to the first non-congested node after the congestion spot, which should be just a few hops downstream.

VI. CONGESTION CONTROL

In this section we outline the methods by which we implement the transport-layer congestion control component of wGPD. In particular, we show how decisions regarding whether or not to inject a packet can be made by examining the size of the PDQ at the source node. We define two types of congestion control, an “unreliable version” that fits within the standard UDP protocol and can be used for flows that can tolerate loss, and a “reliable version” that fits within the TCP protocol and ensures that all data is eventually delivered to the destination. We remark that the entire notion of transport layer congestion control only makes sense for elastic traffic (or semi-elastic traffic) since for inelastic traffic we are expected to inject all the data that arrives. Hence we assume that we have a utility function for each traffic flow for which congestion control is being applied.

Unreliable version: In the unreliable version of the congestion control protocol, we decide whether or not to inject a packet whenever it arrives from the application layer. The decision is made as follows. For each flow f we maintain an average rate xf that is an exponentially-filtered average of the amount of data admitted to the flow. The time constant for this filter is some small parameter _, i.e. xf is multiplied by a factor 1 − _ in each time step and is increased by _`p whenever a packet of size `p is injected into flow f. Suppose that flow f has source sf and destination df . Decisions regarding whether to inject a packet into flow f are made by sf , and are based on the utility function for flow f and the size of the PDQ   we use the same modification that was described for the theoretical GPD algorithm. In this case, a packet is injected as long as Whenever a packet of size `p is injected into flow f, Kf is decremented by an amount `p (but is never allowed to dropbelow zero).

Reliable version: In the reliable version of the protocol,we need sequence numbers and acknowledgments in order to keep track of which bytes are received at the destination. In keeping with the philosophy of making the minimal number of changes to existing protocols, we incorporate the wGPD congestion control protocol within an implementation of TCP which allows us to reuse the TCP sequence number / acknowledgment mechanism. We make two fundamental changes to the TCP congestion control mechanism however. First, we set the TCP congestion window so that it is always equal to the maximum received window size. In this way we essentially disable the effect of the congestion window. Second, whenever TCP makes a decision regarding whether or not to send additional data, instead of using the the TCP congestion window we use the wGPD criterion, namely whether or not We reuse a number of other components from TCP, in particular the timeout and retransmission mechanism. Whenever TCP suffers a timeout and needs to retransmit, we place the data that has timed out in a retransmission queue. The decision regarding when to inject such data again relies on the wGPD criterion. However, the fact that the timeouts and retransmissions operate in essentially the same way as in TCP means that we can reuse many of the enhancements to TCP that are known in the literature, such as Selective Acknowledgment (SACK).

4 Conclusion

In this paper several solutions proposed in the literature for MANETs are discussed. Although all of the research focuses on different problems, they are highly related to each other and have to deal with some common difficulties, which include mobility, limited bandwidth and power consumption. We study the behavior of TFRC equation-based congestion control and multimedia streaming in MANET. Using ns-2 simulations, we show that while TFRC is able to maintain smoother throughput than TCP, it obtains less throughput (0.2 to 0.8) than the competing TCP flows (i.e., being conservative).We analyze several factors contributing to TFRC's conservative behavior, including loss rate discrepancy, inaccuracy of loss rate prediction, and lack of auto-correlation in MANET's loss process, many of which are inherent to the MANET network. We also explore the effect of tuning TFRC's loss event interval estimator, and show that its conservative behavior cannot be completely correct. Our study reveals the limitations of applying TFRC to the MANET domain, and shows that it can be used only when strict throughput fairness is not a major concern. To address the open problem of multimedia streaming in MANET, we propose an alternative scheme (called EXACT-AA) based on router's explicit rate signaling and application's adaptation policies. We demonstrate

the feasibility of our scheme using an audio streaming application over a real MANET test-bed.

References

1. Kaixin Xu, Ken Tang, Rajive Bagordia,; "Adaptive Bandwidth Management and QOS provisioning in Large scale adhoc networks"; Proc.MILCOM; pp. 1018-1023; 2003.
2. Ning Zhang and Alagan Anpalagan; "A Comprehensive Simulation Study of SWAN QoS Model in MANETs with Proactive and Reactive Routing"; Canadian Conference on Electrical and Computer Engineering; 2009.
3. Lyes Khoukhi and Soumaya Cherkaoui; "Experimenting with Fuzzy Logic for QoS Management in Mobile Ad Hoc Networks", Vol. 8(8), August 2008.
4. Jitendranath Mungara, S. P. Setti, G. Vasanth; " Design and a New Method of Quality of Service in Mobile Ad Hoc Network (MANET)"; European Journal of Scientific Research, Volume 34, Issue 1; pp.141-149, 2009.
5. Q.Xue and A.Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks", Journal of Parallel and Distributed Computing, Vol.63, Issue 2, pp.154- 165, 2003.
6. Consolee Mbarushimana and Ali Shahrabi, "Congestion Avoidance Routing Protocol for QoS-Aware MANETs"; Proc. of International Wireless Communications and Mobile Computing Conference, pp. 129-134; 2008.
7. Chin-Fu Kuo, Ai-Chun Pang, S. Chan, "Dynamic Routing with Security

Considerations"; IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 1, pp. 48-58, 2009.

8. Mounir Frikha, Manel Maamer; "Implementation and simulation of OLSR protocol with QoS in Ad Hoc Networks"; Proc. of the 2nd International Symposium on Communications, Control and Signal Processing; 2006.