ECN BASED TCP ENHACEMENT WITHIN WIRELESS MESH NETWORK

[1] *Dhaval B. Patel,* [2] *Sagar J. Dholariya,* [3]*Prof.Naren V. Tada*
[1, 2] **M.E. Student, Computer Engineering Department,**
[3]**Asst.Professor, Computer Engineering Department,**
V.V.P. Engineering College, Rajkot.
Gujarat Technological University,
Ahemedabad, Gujarat, India.

dhavalpatel459@gmail.com,sagarjtp@gmail.com,naren.tada@gmail,com

ABSTRACT : *In the networking field, Now a days Wireless Mesh Network(WMNs) is the very easy and low cost way to access the Internet. Transmission Control Protocol (TCP) which is embedded in the Transport layer is responsible to provide internet access to those wireless devices. While handling Traffic in the wireless mesh network, transmission control protocol faces many challenges. Among these challenges, main four challenges are transmission errors, packet reordering due to multipath routing, multihop connections and congestion. If these are not handled in a proper manner, Transmission Control Protocol causes a poor performance in the Wireless Mesh network. Here new proposal namely Explicit Congestion Notification based Transmission Control Protocol is presented to improve the performance in Wireless Mesh Network. Simulation Results Confirm that new proposal significantly improves TCP performance in Wireless Mesh Network.*

KEYWORDS: Wireless Mesh Network, Transmission Control Protocol,

1. INTRODUCTION

Wireless Mesh Networks [1] is the very easy and low cost way to access the Internet. In Traditional IEEE 802.11 wireless network, access points and the mobile clients use single radio for simultaneous transmission and higher throughput [2], While WMNs uses multiple radios for the same. However, this may cause a conflict with the Transmission Control Protocol (TCP). Designed mainly for wired networks where transmission errors are rare, TCP assumes that the majority of packet losses are caused by congestion and all packets of a connection follow the same path from the source to the destination. Under such assumptions, TCP triggers congestion control mechanisms upon retransmission timeout and three duplicate ACKs. However, these assumptions are no longer true in WMNs. When deployed on WMNs, TCP faces the following challenges:

Transmission Errors: Packet losses due to transmission errors are more frequent and may be incorrectly regarded as indications of network congestion.

Multipath Packet Reordering: Packets may travel through different paths and reach the destination out of- order. If not handled properly, the reordered packets may trigger spurious retransmissions and cause confusions in TCP congestion control.

Multihop Connections: Mesh clients may be connected to the Internet through multihop wireless links. The last-hop access network model used in many wireless TCP enhancements (as in [3, 4]), which assumes the only wireless link is the last hop, is no longer valid.

Congestion: As the wireless medium is shared by mesh routers and clients, congestion may happen in both the WMN and the wired network. Any network model that assumes congestion happens only in the wired network or in the wireless network (such as in the last-hop access network model) will not work.

To handle above four challenges in this paper we present new enhancement. Using a special scheme called congestion coherence; this enhancement distinguishes congestion losses from transmission errors and multipath reordering, and then invokes or suppresses TCP congestion control accordingly. Thus it can significantly improve TCP performance in WMNs.

2. RELATED WORK

There are various wirelesses TCP enhancements have proposed because TCP was found poor performance in the Wireless Networks.

They fall roughly into four categories. Local link layer retransmission methods such as [5] use FEC and ARQ to hide the lossy characteristic of the wireless link so that upper layers are less affected by transmission errors. Split connection methods like [6] divide the entire transmission path into two connections (a wired one and a wireless one) and run TCP separately on both connections. Sender-side enhancements like [7,8] modify TCP code at the sender to estimate the available bandwidth, experienced delay or other congestion signal and change the congestion control accordingly. Wireless-side enhancements [3, 4] modify the behavior of base station or mobile host in cooperation with the TCP algorithm at the source. Of these four categories, the wireless-side enhancements are local to the wireless nodes and links, so are regarded as more practical. Our enhancement proposed in this paper belongs to this category. TCP enhancements that handle packet reordering can be found in [9], but they usually do not treat transmission errors. Our review of current literature finds that no existing enhancements address all the four challenges altogether.

3. ASSUPTIONS

We assume that the Explicit Congestion Notification (ECN) is used in the entire network. ECN has been approved as an Internet official protocol standard in RFC 3168 and is recommended to be widely deployed as a router mechanism. When a router's queue length exceeds a threshold, the incoming packet is marked as *Congestion Experienced*. When the marked packet arrives at the TCP destination, the corresponding ACK is marked as an ECN-Echo and sent back to the source.

Our second assumption is that local retransmissions are implemented in the link layer. All packets are kept in the buffer until positively acknowledged. Failed packets are re- transmitted with higher priority.

4. OBSEVATIONS

Congestion Coherence with ECN, such a distinguishing scheme is possible. When a packet is dropped by a congested router, the ECN congestion signal carried by that packet is lost, but packets before and after the lost packet are likely to maintain coherent congestion information. This is because congestion neither happens nor disappears suddenly. Before congestion becomes so severe that a packet has to be dropped, the queue length has to increase gradually and some packets have to be marked. Similarly, after a packet is dropped, congestion does not disappear immediately. The queue length has to fall gradually. From the time of no congestion to the time a packet is dropped, some packets must be marked.

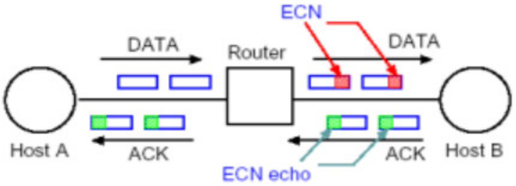

Fig 1.Congestion Coherence

Figure1 Depicts a likely queue length change at a congested router. The abrupt change in Figure 2 is unlikely. As a result, congestion losses are normally preceded and followed by marked packets as in Figure 3. We call this phenomenon **congestion coherence** of ECN marking. The neighborhood of a lost packet is defined by the *coherence context*. In this paper, we use $\{n-1, n+1, n+2\}$ as the coherence context of packet n.

A packet loss is considered a congestion loss if any packet in its coherence context is marked. In this case, the receiver responds with duplicate ACKs to trigger an end- to-end retransmission and window reduction at the source. When the coherence context contains no marks, the missing packet is likely not a congestion loss; the receiver should hold the duplicate ACKs until the retransmitted packet or the packet traveling through a longer path is received.

This idea can apply to the wireless sender case as well. When a wireless sender receives duplicate ACKs, it checks whether the coherence context contains an ECN-Echo. If ,yes the duplicate ACKs are most likely caused by a congestion loss, so the sender invokes the congestion control. Otherwise, the duplicate ACKs are most likely caused by a transmission error or multipath reordering and are ignored.

The distinguishing scheme may make mistakes. If a lost packet is mistakenly classified as a congestion loss, the triggered congestion control action is needed anyway because its congestion context is marked. To deal with other rare mistake cases, a timer can be used to release the incorrectly held duplicate ACKs or retransmitted packets.

5. PROPOSED ENHANCEMENT

TCP Sending algorithm:

Step 1: The TCP sink follows existing algorithm for sending new ACKs, first and second duplicate ACKs.

Step 2: When the Second duplicate ACK is to be sent, TCP sink checks whether the coherence context is marked. If yes, the ACK is sent right away.

Step 3: When the third duplicate ACK is to be sent, TCP sink checks whether the coherence context is marked. If yes, the ACK is sent right away.

Step 4: Otherwise, it is deferred for w ms, and a timer is started.

- If the expected packet arrives during the w ms, a new ACK is generated and the timer is cleared.
- If the timer expires, all deferred duplicate ACKs are released.

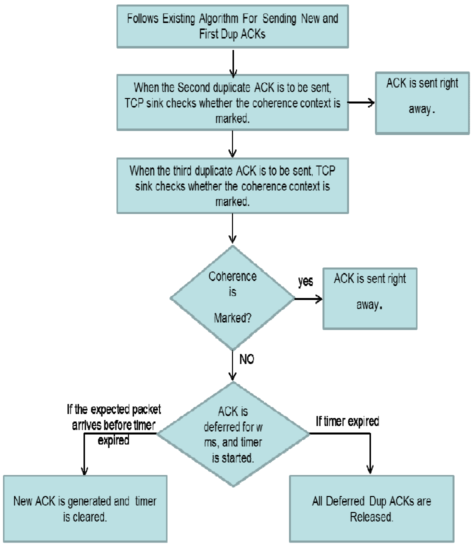

Fig 2.TCP SENDING ALGORITHM

TCP Receiving algorithm:

Step 1: The TCP sender follows existing algorithm for sending packets and updating the congestion window upon receiving new ACKs, and first and second Duplicate ACKs.

Step 2: When the Second duplicate ACK arrives, the sender checks whether any ACK in the coherence context is an ECN-Echo. If yes, the packet corresponding to the duplicate ACKs is sent right away and the congestion window is reduced to 70% if a reduction has not been done in the previous RTT.

Step 3: When the third duplicate ACK arrives, the sender checks whether any ACK in the coherence context is an ECN-Echo. If yes, the packet corresponding to the duplicate ACKs is sent right away and the congestion window is reduced to 90% if a reduction has not been done in the previous RTT.

Step 3**:** Otherwise, the sender ignores the duplicate ACK and a timer of w ms is started.

- If a new ACK arrives during the w ms, the timer is cleared and new packets are sent as if the duplicate ACKs did not happen.
- If the timer expires, the packet corresponding to the duplicate ACKs is sent and the congestion window is reduced to half if a reduction has not been done in the previous RTT.

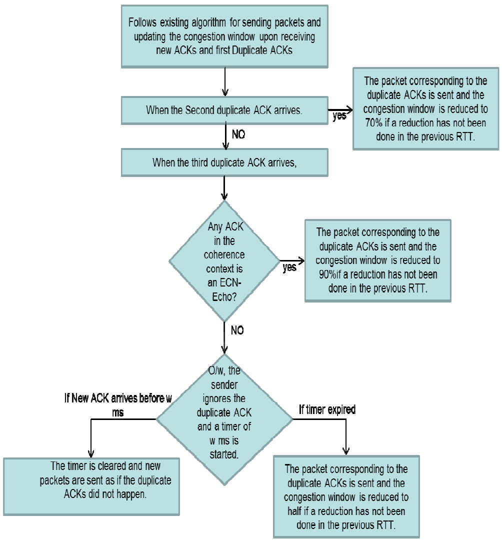

Fig 3. TCP RECEVING ALGORITHM

We emphasize that the modifications to the receiving and sending algorithms are made on the same end. They hide the lossy link characteristics so no changes are needed in the fixed end, intermediate routers and base stations.

6. PERFORMANCE ANALYSIS

In order to evaluate the performance of the proposed Congestion Coherence enhancement, we perform a set of simulations with the *ns*2 simulator [10] on the simplified network model.

Our first group of results, shown in Figure 4, is the TCPcongestion window and queue length of each method. They are collected from 40-second simulation traces. The packet error rate in the simulation is 0.1. The delay and bandwidth should support a window size of about 10 packets, but as shown in the figure, the window size of TCP Reno, Vegas and ECN is frequently reduced. Their corresponding queue length is almost always 0, indicating a low link efficiency. The window size of DDA is significantly increased, but the spikes in the bottom of its cwnd figure show that it suffers severe degradation from timeouts. Congestion Coherence is a thorough solution. Unnecessary window reductions and timeouts are avoided. The queue length figure shows that Congestion Coherence has high link efficiency.

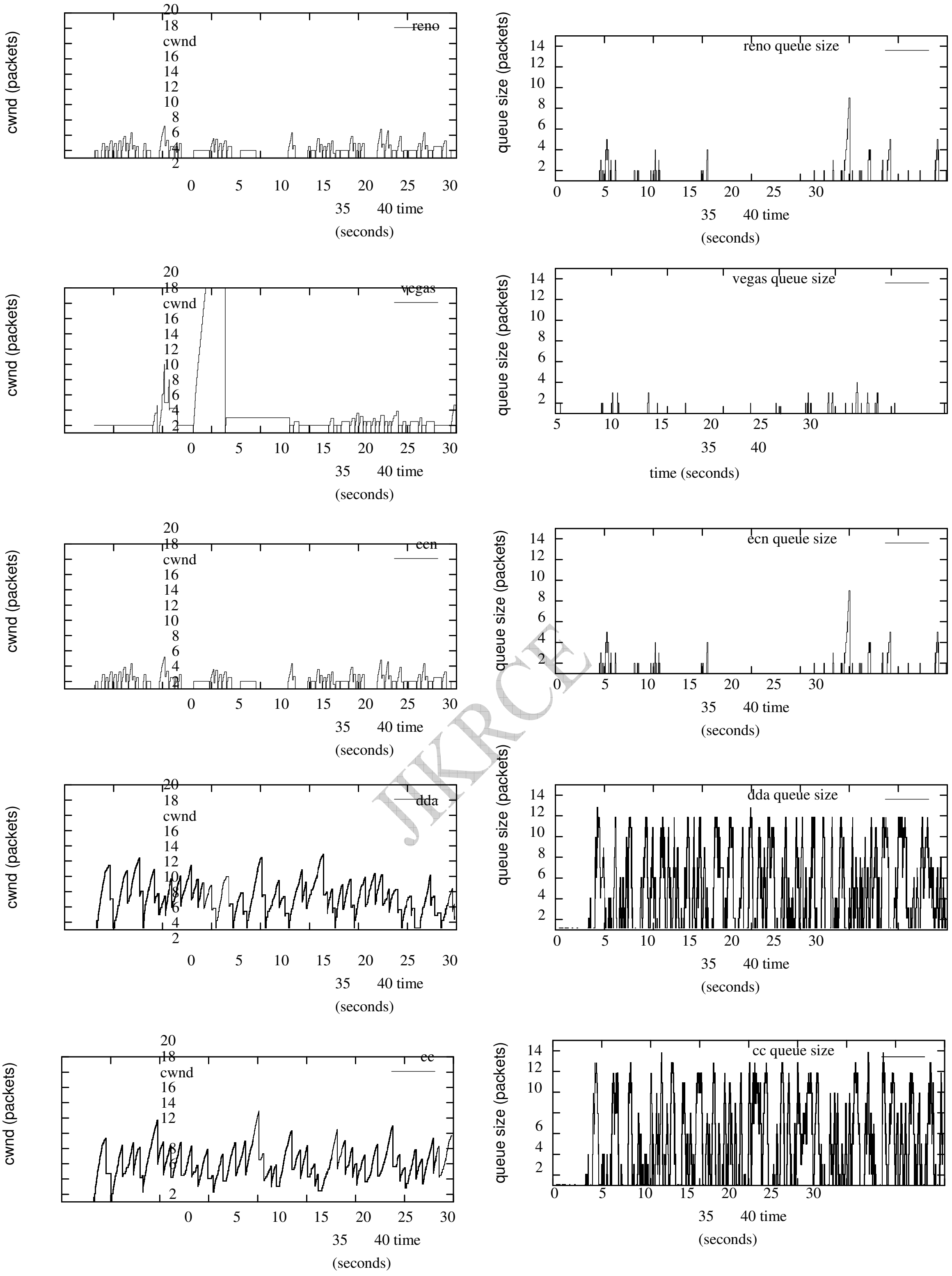

Fig 4. Congestion window and queue length For TCP Reno, Vegas, ECN, DDA and CC

7. CONCLUSION

The Congestion Coherence enhancement we propose in this paper separates the end-to-end task of congestion control from the local operations of wireless retransmission and multipath routing. With local changes at the wireless end and accumulative ECN marking, this enhancement provides a unified enhancement to the four challenges of TCP in WMNs. Simulation results show that Congestion Coherence significantly reduces spurious retransmissions, timeouts, unnecessary congestion window reductions, and therefore provides better improvement than existing wireless TCP enhancements.

8. REFERENCES

[1]http://en.wikipedia.org/wiki/Wireless_mesh_network.

[2] I. F. Akyildiz, X. Wang and W. Wang, Wireless mesh networks: a survey, *Computer Networks*, vol. 47, pp. 445–487, 2005.

[3] H. Balakrishnan, S. Seshan and R. H. Katz; Improving reliable transport and handoff performance in cellular wireless networks; *Wireless Networks*, 1(4), pp. 469 – 481, 1995.

[4] H. Balakrishnan and R. H. Katz; Explicit loss notification and wireless web performance, *Proceedings of IEEE Globecom 1998*.

[5] G. Xylomenos and G.Polyzos, Internet protocol performance over networks with wireless links, *Computer Networks*, 37(5):601–615, 2001.

[6] A. Bakre and B. R. Badrinath, I-TCP: indirect TCP for mobile hosts, *Proceedings - International Conference on Distributed Computing Systems*, pp. 136–146, 1995.

[7] L. Brakmo, S. O'Malley and L.Peterson, TCP Vegas:New techniques for congestion detection and avoidance, *Proceedings of SIGCOMM*, 1994.

[8] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi and R. Wang, TCP Westwood: Bandwidth estimation for enhanced transport over wireless links, *ACM Mobicom*, pp. 287–297, 2001.

[9]K. Leung, V. Li and D. Yang, An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 4, pp. 522–535, 2007.

[10] http://www.isi.edu/nsnam/ns.