Volume 2, Issue 2, April 2015 ISSN: 2348-8565 (Online)

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Congestion Control in MANET Using the Dynamic Queue Management

Deepesh Nigam
M.Tech. Research Scholar
Department of Computer Science & Engineering
Lakshmi Narain College Of Technology
Jabalpur (M.P.), [INDIA]
Email: deepeshngm@gmail.com

Sujeet Tiwari
Assistant Professor
Department of Computer Science & Engineering
Lakshmi Narain College Of Technology
Jabalpur (M.P.), [INDIA]
Email: sujeet.tiwari08@gmail.com

Raghvendra Kumar Agrawal
Assistant Professor
Department of Computer Science & Engineering
Lakshmi Narain College Of Technology
Jabalpur (M.P.), [INDIA]
Email: raghvendraagrawal7@gmail.com

ABSTRACT

In MANET have limited bandwidth and are more prone to error than wired networks which further impose limits on the amount of data that can be sent. In order to conserve the limited resources, it is highly desirable that transmission should be as efficient as possible with minimal loss. The objective of congestion control is to limit the delay and buffer overflow caused by network congestion and provide better performance of the network. The traditional congestion control mechanism, applied by the transport control protocol is unable to catch up the network dynamics of ad-hoc networks. Congestion control assumes all losses induced by congestion. In this paper, a novel approach of congestion control for supporting applications like multimedia streaming over MANET is being proposed.

Keywords:*—Link failure, Velocity Change, Position Congestion Control, Non-congestion Loss.*

I. INTRODUCTION

A mobile Adhoc network is composed of a group of mobile computing devices (nodes) that are equipped with wireless-LAN capability. In contrast to infrastructure wireless network, which uses base station to manage nodes in its area, MANET does not require any fixed infrastructure. Nodes in multi hop MANET help each other to forward packets from hop to hop such that two nodes that cannot hear each other can transmit data to each other. In this way, the connectivity of a MANET is greatly enhanced. In expensive deployment of MANET due to absence of fixed infrastructure as well as mobility feature for all nodes have considered MANET as a subject of research. In a MANET environment, communication links are unstable due to various reasons such as interference of radio signal, radio channel contention, mobility of the nodes and battery depletion. The wireless network have limited bandwidth and are more prone to error than wired networks which further impose limits on the amount of data that can be sent. Hence, in order to conserve the limited resources, it is highly desirable that transmission should be as efficient as possible (minimal loss and transmission). The main objective of congestion control is to limit the delay and buffer overflow caused by network congestion and provide better performance of the network[1]. In wire line networks, congestion control is implemented at the transport layer and is often designed separately from functions of other layers. Since wired links have fixed capacities and are independent, this methodology is well justified and has been extensively studied[18]. However, these results do not apply directly to Adhoc networks because the ad hoc networks result in large amount of packet loss, high delay, unfair scenarios and low throughputs. In Adhoc networks, each mobile node has limited transmission capacity and buffer and they mostly intercommunicate by multi-hop relay [1]. The random behavior of Adhoc networks cause the topology of wireless network to be changed rapidly and unpredictably. Moreover, node's mobility puts an extra burden on TCP"s congestion control mechanism. As a result, traditional congestion control mechanism, applied by the Transport Control Protocol (TCP) [8], is unable to catch up the network dynamics of ad hoc networks. Congestion control is the most controversial part of TCP which degrades performance when encounters non-congestion loss in MANET. Congestion control assumes all losses induced by congestion. For example, link breakage lasts greater than Retransmission Time Out (RTO) is miss-interpreted as congestion loss. Thus regardless of kind of loss, it decreases sending rate to alleviate congestion and grows retransmission timeout exponentially to wait more for receiving acknowledgment. It is plausible in wired network since non-congestion loss occurs rarely and some application can tolerate some degrees of error. However, this unnecessary throughput drop that waste available resources such as bandwidth arises in MANET. Link failure needs TCP to explore how much new route is congested in comparison to the broken one. Traffic characteristics can affect queuing delay and processing delay of intermediate nodes that consequently influences Round Trip Time (RTT). If discovered route suffers heavier traffic than old one, retransmission timer must wait more to receive acknowledgment and RTO should be increased. Otherwise, when new route is approximately non-congested, data packets and acknowledgment transferred quicker than old route. Thus sender must wait less than before to receive acknowledgment and RTO should decrease.

II. Related Work

Sender should recognize state of MANET and wireless link to act accordingly. For example, specifying available buffer of intermediate nodes that assess congestion can greatly influence recovery operations. Measuring remained energy of nodes can assists sender to change route before link breakage. Calculating distance between nodes based on signal strength can help sender to predicate future link failure and switch into another route before breakage. All these information can be either measured explicitly with support from intermediate nodes or estimated implicitly from information in received acknowledgment. In First mechanism, TCP sender entirely does the job and estimate MANET situation implicitly without any support from intermediate node. It does not create processing overhead at intermediate routers. The main drawback of it is the lack of detailed information about state of wireless link at the sender [16]. For example, Fixed RTO interprets two successive timeout as route failure. Then retransmit unacknowledged packet while it keeps value of RTO unchanged [9]. However two successive timeout can be sign of congestion in congested MANET and is highly based on existing traffic pattern. That's why it is not precise enough. In feedback (cross layers) approaches, sender get detailed information from network state by collaborating between

TCP layers of intermediate nodes. For example, since congestion control is not aware from losses due to wireless medium contention over 802 MAC protocol, it must collaborate with MAC layer to address these losses. Although Feedback methods are more precise than end-to-end approach [13], modifications in intermediate nodes make implementation complicated for WAN. Moreover, extra overhead produced due to transmission notification packet. In addition, it reduces flexibility [5]. For example, TCP Muzha forces Intermediate nodes to fill special field in acknowledgment header to clarify sender how many empty rooms are available in their buffers [19]. TCP-F [12] and TCP-BUS[3] are feedback approaches that pursue the same mechanism. When intermediate node detects link breakage, route notification message informs source to stop sending further packets and freeze state variable such as RTO. When route rebuilt, route reconstruction notification packet informs source to resume transmission with old RTO. Westwood VT is an end-to-end approach, which classifies packet loss by estimating existing data packet in buffer of intermediate nodes. It is too resemble to TCP-Veno [2]. Actually both inherit policy of TCP Vegas to differentiate causes of packet loss [6]. After received acknowledgment, they measure the difference between expected rate and actual rate and assign it to Δ which is indication of amount of buffer in queue of middle nodes. Interpreting causes of loss is done based on two predefined threshold α and β and available buffer of intermediate nodes as Δ. If it becomes smaller than α, buffers of intermediate nodes still can accommodate incoming packets. So WestwoodVT relates any loss due to the wireless error. If Δ is larger than β, it shows that buffers are approximately full and any packet loss is due to congestion [6]. If estimated Δ becomes between two thresholds, decision is postponed to next losses. Main drawback of WestwoodVT that degrades performances (throughput and energy consumption) is revealed when Bit Error Rates (BER) increases[17]. In addition, WestwoodVT cannot address link failure. TCP-Feno introduces another challenge on TCP VEGAS proponents (WestwoodVT and TCP-Veno). It claims TCP VEGAS performance degrades in network when nodes use small buffer size [7]. MANET with nodes carrying small buffer size can quickly enter into congestion mode. However, since TCP VEGAS does not contribute maximum buffer size in estimation, it just compares Δ with threshold α and find out it is less than it. So TCP VEGAS declares loss as non-congestion loss while congestion exists. However, TCP-Feno still cannot cope with two first mentioned problems.

LDA_RQ is an implicit end-to-end approach which tries to estimate queue usage rate of intermediate nodes. It does not need any support or feedback from middle nodes. Available Information in transport layer is congestion window size (cwnd), round trip time (RTT) [14]. It defines two loss classification formulas, one for beginning and another for rest of transmission. It compares first classification metric with threshold until maximum EROTT exceeds three times greater than minimum EROTT. After this gap appeared, it uses second classification until the end. In addition, special ROTT called congestion ROTT calculated to show border between normal and congested MANET. When TCP recognizes loss through third duplicate acknowledgements, it verifies weather queue usage exceeds 50% or current ROTT becomes greater than congestion ROTT. If either former or latter satisfied, detected loss is due to congestion. Otherwise, loss is induced by non-congestion factors [14]. However, it cannot detect link failure. In addition, gap between minimum and maximum of EROTT achieved experimentally that definitely varies based on experiment. Moreover, in situation which gap cannot reach to three queue usage remains less than 30% and congestion EROTT might not be initialized. TCP-welcome is an implicit end-to-end scheme, which differentiates causes of packet loss based on history of Round Trip Time. Ascending growth of RTT increment induced by congestion. However, If RTT didn't fluctuate and remained around averaged value, the way packet loss recognized becomes important. Three duplicate acknowledgements are a consequence of wireless channel error while retransmission timeout is due to link failure[17]. However, TCP-WELCOME uses RTT, which includes both delays of forward and reverse path while only delay of forward path must be considered. In addition, it offers recovery method based on RTT comparison. TCP-Welcome claimed that RTO adjustment should be done based on the Capabilities of discovered route such as length, load and link quality. After link breakage, total delay for new route varies from broken route. Hence, RTT comparison seems to be suitable Parameter for tuning = (1).

However, RTT is not enough for depicting capabilities of discovered route. In addition, it Includes both delay of forward and backward path. ABRA does not offer method to classify packet losses. However, it uses smoothed Round Trip Time instead of RTT to set RTO after link breakage. When link failure lasts more time than RTO, timeout happens. Standard TCP grows RTO exponentially due to multiple successive back-offs. When route come back, TCP cannot retransmit last unacknowledged packet since it must wait until this long RTO expires. Thus, it is serious deficiency since route recovered but TCP remains idle unnecessarily. ABRA claims that new RTO is dependent on the smooth round trip time (SRTT) [4]. In end-to-end threshold-based algorithm, this enhances congestion control to address link failure loss in MANET. It consists of two parts. Threshold-based loss classification algorithm uses queue usage to classify network state periodically into congestion or non-congestion mode. Any retransmission timeout in period which MANET is non-congested mode is an indication of link failure loss. In addition, implementation showed that small percentage of three duplicate acknowledgments which emerge immediately after route recovery might be result of route changes. After detecting losses due to link failure, it should adjust RTO for reconstructed route by comparing its capabilities with broken route using available information in transport layer. This enhances congestion control by transmitting packet as soon as route recovered rather than being idle unnecessarily. when we send to destination the ack packet travels from source to destination through the intermediate nodes but the calculation is performed only at the source node. In case we want to find out the losses at intermediate nodes then the overhead of calculation will increase as then we have to send ack Packet at each intermediate node and thus calculation will be performed at each node respectively.

III. PROPOSED WORK

Work done by [1] uses ODMRP, which is not a popular reactive protocol for implementation of multicasting in MANET. It applies measurement based detection and accusation-based reaction techniques which are applied when attack has been detected means attacks already exist in the system and might have harm the performance and theft data. It bounds the impact of attacks means minimizes it.

In this work I will be using a proactive mechanism for avoiding the security threats and attacks in the system using following Algorithm:

Step 1: A Network topology shall be created using Network Simulator Software Version with moving nodes

Step 2: Nodes shall be placed randomly to map the wireless sensor network

Step 3: Nodes will be using AODV routing protocol for routing between them

Step 4: Nodes will be initialized with Constant Bit Rate (CBR) traffic for mapping the communication between them.

Step 5:Nodes will communicate with neighbours which are lying under a minimum and maximum distance between them

Step 6: Distance shall be measured by storing their current position in each node.

Step 7: Throughput and End-to-end delays shall be measured for existing network without modification and with modification to compare.

Step 8: The experiments shall be executed with different number of nodes and different communication packet sizes.

This work proposes a new approach technique to initialize the routing protocol for wireless networks application. The proposed work steps has been discussed and for implementation of the proposed work, we have modified mac_802_11.h and mac_802_11.cc files to include the authentication and keys generated for AODV protocols. It also includes functions for generating keys, authentication status, malicious nodes, performing checks for the various nodes data transfer requests.

Generally speaking, network simulators try to model the real world networks. The principal idea is that if a system can be modeled, then features of the model can be changed and the corresponding results can be analyzed. As the process of model modification is relatively cheap than the complete real implementation, a wide variety of scenarios can be analyzed at low cost (relative to making changes to a real network). NS-2 is an Object-Oriented, discrete event network Simulator developed at UC Berkeley. It is written in C++ and OTcl (Object-Oriented Tcl) and primarily uses OTcl as Command and Configuration Language. NS is mainly used for simulating local and wide area networks. It simulates a wide variety of IP networks. It implements network protocols such as TCP and UDP, traffic source behavior such as FTP, Telnet, Web, CBR & VBR, router queue management mechanisms such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations. The NS project is now part of the VINT project that develops tools for Simulation results display, analysis & converters that convert n/w topologies generated by well-known generators to NS formats.

As shown in Figure 1-(a), in a simplified user's view, NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup (plumbing) module libraries (actually, plumbing modules are implemented as member functions of the base simulator object).

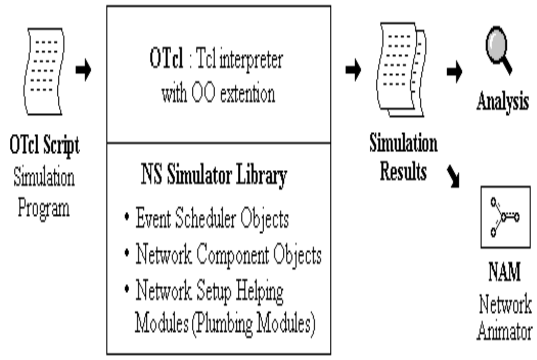

Figure 1: Working of NS2 Simulation Environment

IV. CONCLUSION

This paper motivated the need for loss classification in mobile ad-hoc networks and presented a novel approach which manages the mobility of node proactively, is easy to implement and imposes minimal computational burden on the resource constraint device. This approach will be required to less number of acknowledgement packets than Enhance congestion control techniques are used to address link failure and to control the congestion [11]. Result from this paper have shown that MANET performance can be improved by using novel approach as it reduces packet loss ratio (Thereby reducing the number of retransmission) and increase transmission efficiency. Moreover, as its computational burden is negligible, it is ideally suited for resource constrained environment such as MANETs.

REFERENCES:

[1] C.E. Perkins, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" Proc. ACM SIGCOMM, pp.234-344, 1994.

[2] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," IEEE J. Sel. Areas Communications, Vol. 21, No. 2, pp. 216–228, Feb 2003.

[3] D. Kim, C.K. Toh, Y. Choi, "TCP-BUS: Improving TCP performance in wireless ad hoc networks", IEEE International Conference on Communications, Vol. 3, pp. 1707 – 1713, New Orleans, LA, USA, 2000.

[4] Ghanem T.F, Elkiliani, W.S, Hadhoud, M.M "Improving TCP performance over Mobile Ad Hoc Networks using an adaptive back off response approach", IEEE Conferences on Networking and Media Convergence, ICNM, pp.16-21, Cairo, 2009.

[5] HM El-Sayed, O Bazan, U Qureshi M. Jaseemuddin, "Performance Evaluation of TCP In Mobile Ad-Hoc Networks", The Second International Conference on Innovations in Information Technology, 2005.

[6] Jahwan Koo, Sung-Gon Mun, and Hyunseung Choo, "Sender-Based TCP Scheme for Improving Performance in Wireless Environment", Proceedings of the 7th international conference on Computational Science, 2007.

[7] Jae-Hyun Hwang, See Hwan Yoo, Chuck Yoo, "TCP Feno: Enhancement for Higher Accuracy of Loss Differentiation over Small Buffer Heterogeneous Networks", Local Computer Networks, the 34th IEEE Conference, pp. 249 – 252, 2009.

[8] J. Postel, "RFC 793: Transmission Control Protocol", 1981.

[9] T. Dyer and R. Boppana. "A comparison of TCP performance over three routing protocols for mobile ad hoc networks" In Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc‟01), Long Beach, pp.56-66 CA, USA, 2001.

[10] Kai Chen, Yuan Xue, Nahrstedt K" On Setting TCP Congestion Window Limit in Mobile Ad-Hoc Networks", ICC '03, IEEE International Conference, pp. 1080– 1084, 2003.

[11] Kim K W, Lorenz P, lee MMO, A new tuning maximum congestion window for improving TCP performance in MANET", Systems Communications, Proceedings in IEEE Conferences, 2005.

[12] K. Chandran, S. Raghunathan, S.

Venkatesan and R. Prakash, “A feedback based scheme for improving TCP performance in ad-hoc wireless networks”, IEEE Journals Personal Communications Magazine, Vol. 8 pp.34–39, Amsterdam 2001.

[13] Mi-Young Park, Sang-Hwa Chung, “Analyzing Effect of Loss Differentiation Algorithms on Improving TCP Performance” the 12th IEEE International Conference on, (ICACT), Vol.1 pp.737– 742, Phoenix Park 2010.

[14] Mi-Young Park; Sang-Hwa Chung; Sree kumara,” Estimating Rate of Queue Usage to differentiate Cause of Packet Loss in Multi-hop Wireless Networks”, Computer Software and Applications, COMPSAC'09.33rd Annual IEEE International Conferences. Vol. 1, pp.500–508. Seattle, WA 2009.

[15] Mohammad Amin Kheirandish Fard, Sasan Karamizadeh, Mohammad Aflaki, “Enhancing Congestion control to address link failure loss over mobile ad-hoc network”, International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.5, Sep 2011.

[16] Samaraweera N.K.G, “Non-Congestion packet loss detection for TCP error recovery wireless links” Communications, IEE Proceedings-IET Journals, 1999.

[17] Seddik-Ghaleb, A. Ghamri-Doudane, Y. Senouci, S.-M, “TCP Welcome TCP variant for Wireless Environment, Link losses, and Congestion packet loss Models”, IEEE Conferences on Communication systems and Networks, pp.1–8, Bangalore 2009.

[18] T. Senthil Kumaran, Sankaranarayanan, “Early detection congestion and control routing in MANET,” In Proceedings of the Seventh IEEE and IFIP International Conference on wireless and Optical communications Networks (WOCN), pp. 1- 5, 2010.

[19] Yao-Nan Lien; Ho-Cheng Hsiao, “A New TCP Congestion Control Mechanism over Wireless Ad Hoc Networks by Router-Assisted Approach”, 27th IEEE International Conference on Distributed

* * * * *