

Performance Evaluation of TCP/UDP and Its Variants for A Network Using NS-2

Shalini V S¹, Karthik S², Jayalaxmi³, Kavyashree M G⁴

Asst. Professor, Department of Electronics & Communication, ATME College of Engineering, Mysuru,
UG students(BE) Department of Electronics & Communication, ATME College of Engineering, Mysuru.

Abstract— The method or the way from which the information is passed from one node to another node and vice versa is defined as communication network. So those network structures uses different protocols such as TCP/UDP to efficiently route the messages across the node. Hence the throughput and performance of different protocols such as TCP, UDP and variants of TCP such as Tahoe, Vegas, New Reno and Sack are evaluated using NS-2. By considering the nodes in a network their throughput are evaluated by using TCP, UDP and share topology scenarios. Here the Simulation is performed on network in order to study the performance of different protocol by keeping Bandwidth.

Keywords— Tahoe, Vegas, New reno, Sack, Network Simulator-2(NS-2), Efficiency.

I. INTRODUCTION

Network Simulator (Version 2), widely known as NS2, simulation tool which is meant for studying dynamic nature of communication i.e. for both wired and wireless network functions and for routing algorithms, TCP, UDP protocols. Basically it provides specific way of simulating such network protocols.

This paper investigate the performance comparisons of throughput for a network using different protocols such as TCP,UDP and TCP variants as aforementioned and find out which one is better in which cases. By varying the bandwidth for each protocol network, throughput is compared.

II. LITERATURE REVIEW

TCP was officially adopted as a standard in RFC793 in 1981[2] and was designed to deal with message flow control and error correction, ensuring reliable delivery of message from source to destination. Later IP was adopted as a standard in RFC791in 1981[3], which deals with logical addressing and specifies source and destination addresses. Michael Welzl describes the background and concepts of Internet congestion control, in an accessible and easily comprehensible format He aimed to give a thorough understanding of the evolution of Internet congestion control. V.Jacobson proposed the congestion window CWND to reflect the network capacity. Later an algorithm called as Fast-Re- Transmit was developed[5], For faster and congestion avoidance .Now in this project/paper we propose the high efficiency and congestion avoidance by using of different protocols such as

TCP, UDP, Combination of TCP/UDP and by using variants of TCP.

III. OVER VIEW

A. TCP

The important protocol is Transmission Control Protocol (TCP) used for transmission in a network. Internet widely uses TCP protocol for data transmission in communication network. In order to perform transmission connection is established between the client and the server.

Connection is initiated by the Client and sends Sequence number along with the segment. So the Server acknowledges it back with its own Sequence number and ACK of client's segment which is one more than client's Sequence number.

Client after receiving ACK of its segment sends an acknowledgement of Server's response..

B. UDP

The User Datagram Protocol (UDP) is simplest Transport Layer communication protocol available of the TCP/IP protocol suite. Communication mechanism involved is minimum. UDP is an unreliable transport protocol which uses IP services to provide best delivery mechanism.

In UDP, there is no generation of acknowledgement of received packets to sender and senders do not wait for any acknowledgement in turn of packet. This makes UDP protocol unreliable on processing. UDP is implemented where packet acknowledgement have same significance on bandwidth as that actual data. In case of video streaming, thousands of packets are generated towards its users. Acknowledging all those packets is troublesome and which results in huge amount of bandwidth wastage. The best delivery mechanism of underlying IP protocol ensures best efforts to deliver its packets, but even if some packets in video streaming get lost, the impact is can be ignored easily. Loss of few packets is sometimes neglected in case of video and voice traffic sometimes goes unnoticed.

C. TCP Variants

i. Tahoe

Tahoe refers to the TCP congestion control algorithm. 'conservation of packets' is the principle on which TCP is based on, i.e. packets are not inserted to network unless they are taken out only if the packets are running at designed bandwidth in a connection. The basic rule followed by this principle is "acknowledgements" which ensures packets received by receiver as it traces the outgoing packets by acknowledging to clock. It also helps in maintaining a congestion window CWD that determine the network capacity. Among the five variants TCP Tahoe is the simplest one. Packet recovery is not that fast. The triple duplication ACKs is treated as same as timeout only in the phase of congestion avoidance. Fast packet retransmission is performed only after receiving timeout or triple duplicate ACKs, which in turn reduce congestion window to 1, and proceed to slow-start phase.

ii. Reno

Slow starts and the coarse grain re-transmit timer is the principle preferred by Reno which is the basic principle of Tahoe. Added advantage by the principle is that early detection of lost packets and avoids the emptying of pipeline at every packet loss. Acknowledging immediately after the reception of the segments i.e. packets is necessary element in Reno. The duplicate acknowledgment is received only when packets of next segment in sequence is expected or delayed by the network or packets received is out of order or lost which is the logic behind the principle. Perception behind the duplicate acknowledgements is that sufficient time is has passed in transmission and if the packets have taken a longer path, which is now available for reception. To overcome all this problems Reno assists algorithm known as 'Fast ReTransmit' which reduces probability of packet loss. It is assumed that after reception of 3 duplicate ACK's packets in the segment is lost, hence packet re-transmission is performed in the segment without any timeout. There by managing re-transmission to make pipe almost full. Modification done by RENO after a packet loss is reduction of congestion window to 1 by emptying pipe.

iii. New Reno

New RENO is a slight modification over TCP-RENO. It is much more efficient than RENO since it enables multiple packet losses. New-Reno also enters into fast-retransmit algorithm same as that Reno immediately after receiving multiple duplicate packets, only difference between Reno and New Reno is that it come out of fast-recovery unless the data enters fast recovery algorithm at the reception time while acknowledging. There by it avoids the problem of multiple CWD faced by Reno. New Reno as same fast-transmit phase as in Reno. The multiple re-transmissions is permitted by fast recovery phase difference in new-Reno. After entering into fast recovery maximum not received segments are noted down. The fast-recovery phase is same as that of Reno, but there are two cases whenever a fresh ACK is received i.e. if all packets that are not received is acknowledged as soon as on entering fast recovery it exits out of fast recovery and assign CWD to slow start threshold and proceeds with congestion avoidance like Tahoe. If partial ACK is done it specifies that

packet of the next segment in pipeline has lost then it re-transmits the packets of segment and assigns number to duplicate acknowledgments received to zero. Finally it comes out of Fast recovery when the entire packet acknowledged in the window.

iv. Sack

TCP with 'Selective Acknowledgments' is an extension of TCP Reno and it solves the problems faced by TCP RENO and TCP New-Reno, i.e. detection of multiple packet loss, and re-transmission of more than one lost packet per RTT. The slow-start and fast retransmits parts of RENO is retained in sack. It also retains the coarse grained timeout of Tahoe to track on, packet loss detection doesn't follow modified algorithm. SACK TCP doesn't ask for the cumulative acknowledgement but selectively acknowledgment is required. Thus segments being acknowledge is defined by block in each ACK. Thus the sender will have clear picture of acknowledged segments and segments that not received (outstanding). Outstanding data is estimated by initializing variable pipe whenever the sender enters fast recovery and it also set congestion window to half the current size. By receiving an acknowledgment every time pipe is reduced by 1 and segment is incremented by 1 on retransmission. Congestion window checks for the pipe size as soon as pipe goes smaller than window the unreceived segments are transmitted. If there are no such unrecievd segments then it continues to transmit a new packet.

v. Vegas

TCP Vegas is a TCP congestion avoidance algorithm that is concerned on packet delay, rather than packet loss, which helps signal to determine the rate at which packet are sent. TCP Vegas detects congestion at an earlier stage based on increasing values of Round-Trip Time (RTT) the packets in the connection unlike other flavors such as Reno, New Reno, etc., which detect congestion only after it has actually happened via packet loss. The algorithm depends heavily on accurate calculation of the Base RTT value. If it is too small than throughput of the connection will be less than the bandwidth available while if the value is too large then it will overrun the connection. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network. It does not depend solely on packet loss as a sign of congestion. It detects packet losses before the occurrence congestion. But still it retains the other mechanism of Reno and Tahoe, and a packet loss can also be detected by the coarse grain timeout when other mechanisms fail.

IV. METHODOLOGY

The proposed network is simulated using NS2 software. In order to simulate the algorithm contains the following steps

- New simulator is created.
- Files are opened to store simulation results.
- Nodes are created such as n0, n1, n2, n3, n4 and n5.

- Links are established between all the nodes present .
- Transport agents are created, transport agents might be TCP, UDP or variants of TCP.
- Connection is established between the transport agents used in the network.
- Later traffic agents are created.
- Once the traffic agents are created finish procedure is written.
- Simulation time or the run time is set.
- Finally, Simulation is performed.

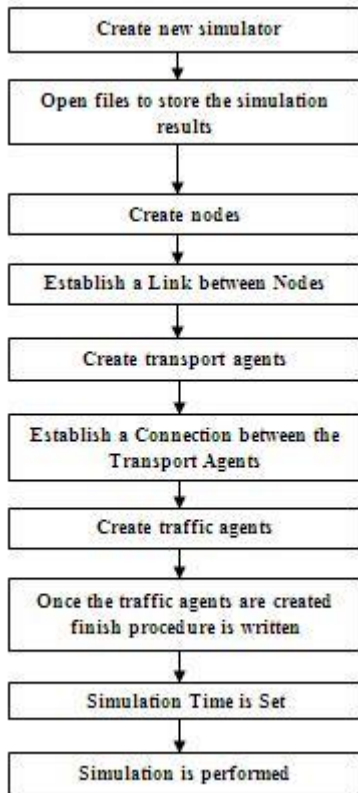


Fig 2: Flow diagram

D. PROPOSED NETWORK

As shown in figure the network is built to perform the simulation using ns2 software. The network consists of 16 nodes out of which n0, n1, n2, n3, n4 are made as source node and n16 is made as destination node. Here n0 is linked to n5, n5 to n6, n6 to n7, n7 to n8, n9 to n10 and n10 is connected destination node n16. But n1 is linked to n11, n11 to n12, n12 to n5 and n5 is linked to destination node n16 via nodes n6, n7, n8, n9 and n10 also n2 is linked to n13, n13 to n14, n14 to n9 and n9 is linked to destination node n16 via n10. Node n3 is linked n15 which is further connected to n16 via n14, n9, n10 but source node n4 is directly linked to n16.

60.86% Packet generator such as FTP is attached to all these source nodes which in turn generate packets required for transmission. The duplex link is

established between all the nodes in order to provide two way communications between the nodes.

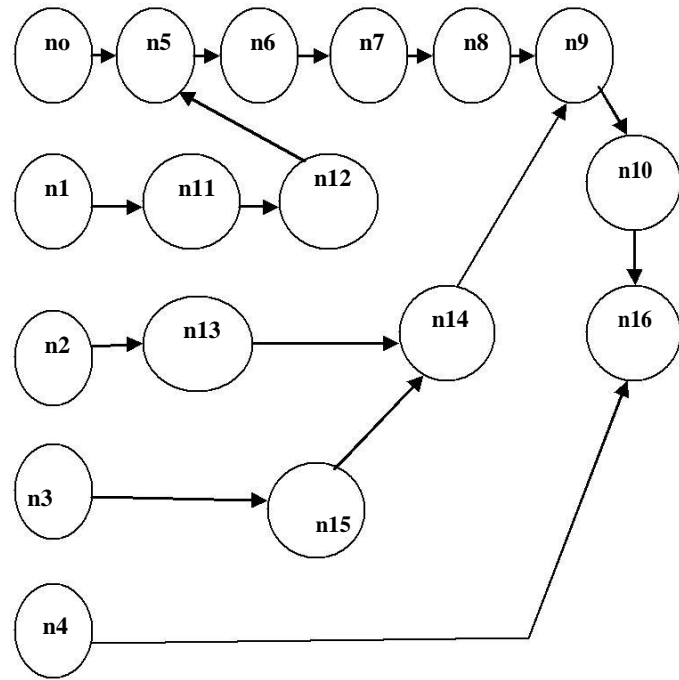
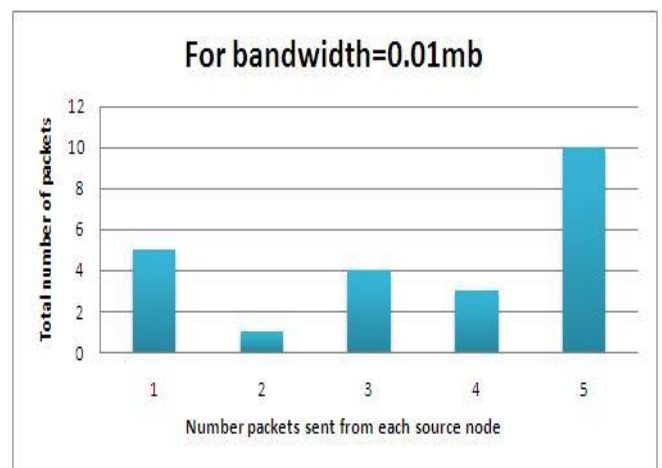


Fig 2: Wired Proposed Network using NS-2

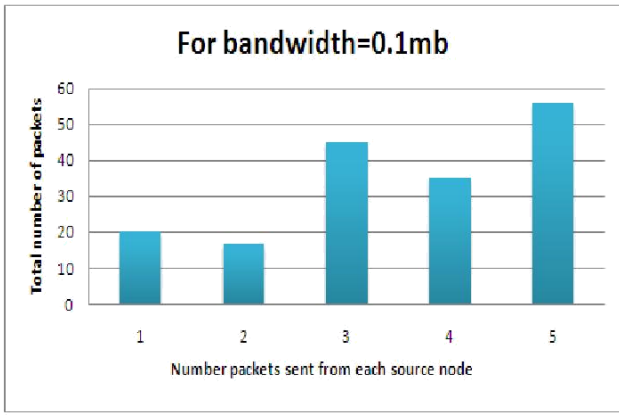
V. SIMULATION AND RESULT ANALYSIS

Scenario 1:Using TCP

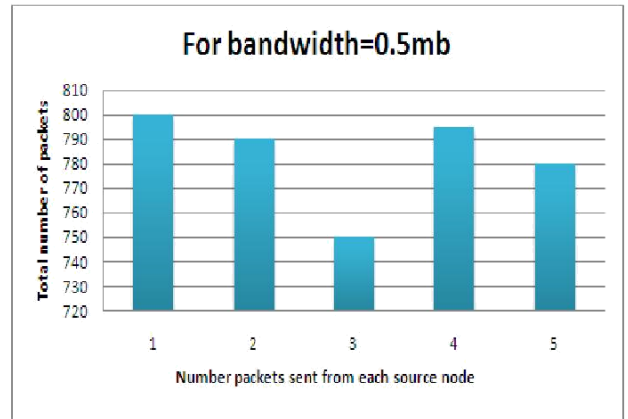
In this scenario, for the above proposed network TCP protocol is connected to all source nodes and the bandwidth of links are kept constant. Throughputs and efficiency are calculated. For bandwidth 0.01mb



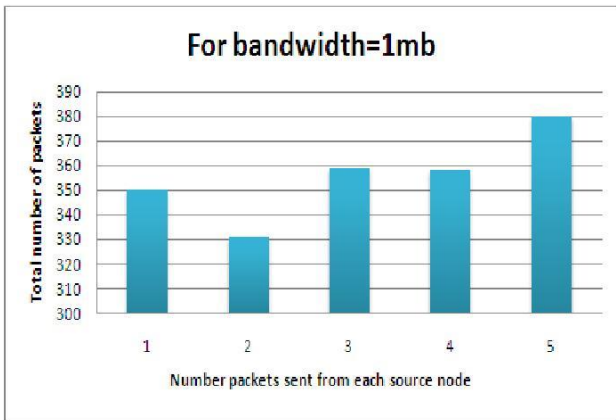
Efficiency =



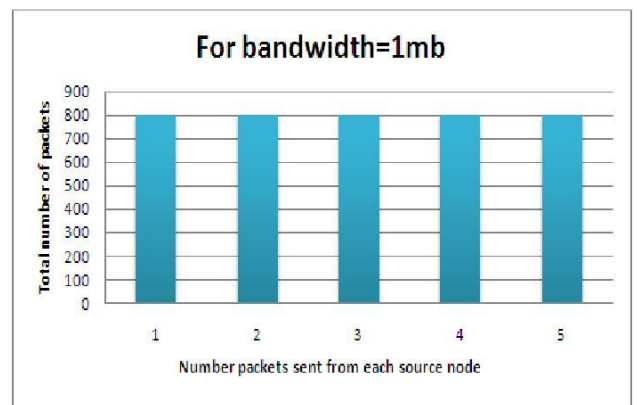
Efficiency = 88%



Efficiency = 75%



Efficiency = 98.8%



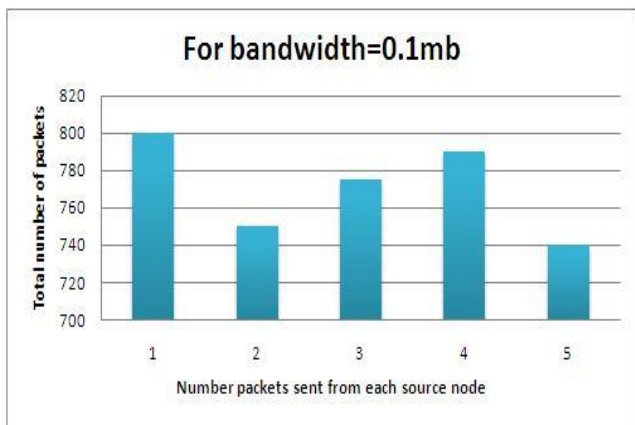
Efficiency = 100%

Scenario 2: Using UDP

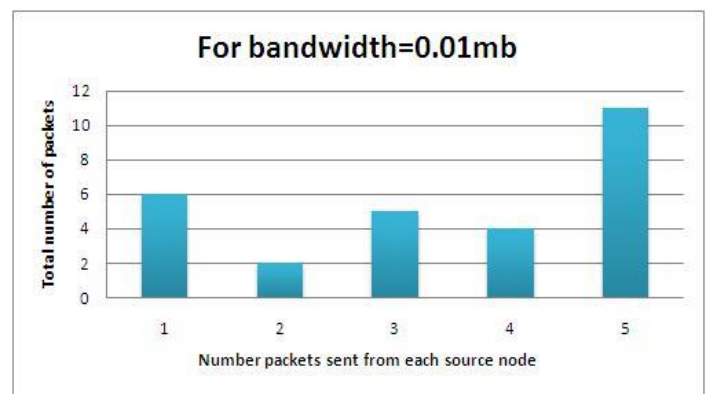
In this scenario, for the above proposed network UDP protocol is connected to all source nodes and the bandwidth of links are kept constant. Throughput and efficiency are calculated.

Scenario 3: Using various variants of TCP

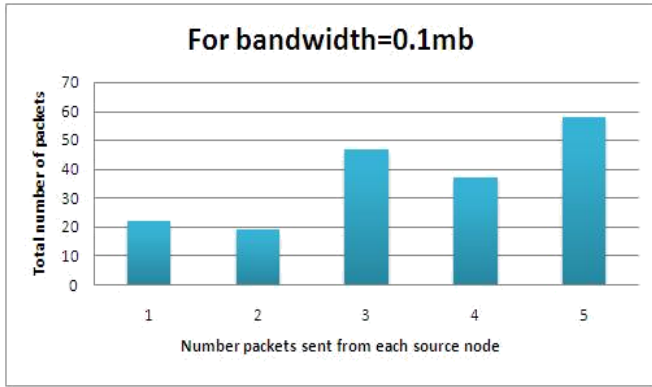
In this case various variants of TCP such as Tahoe, Reno, New Reno, Vegas and Sack are connected to source nodes of the network. By using these variants the throughput and the efficiency is calculated. All 3 scenarios' together provides the performance such as throughput of TCP, variants of TCP and UDP for a proposed network. All the scenarios are repeated by considering different bandwidth for the proposed network and the performance of TCP, variants of TCP and UDP is analyzed.



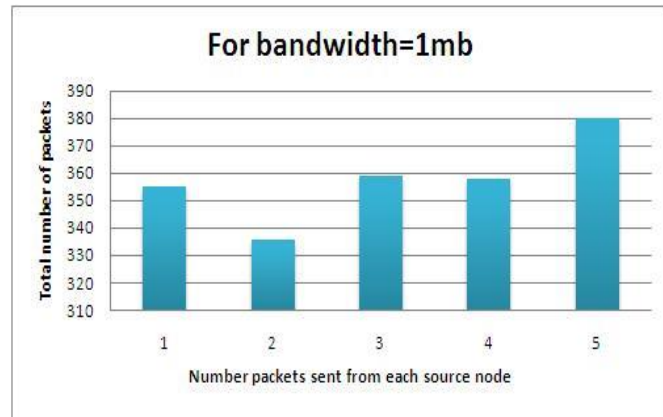
Efficiency = 25.9%



Efficiency = 67.8%



Efficiency = 89.07%



Efficiency = 99.42%

CONCLUSION

For a network consisting a number of 16 nodes, comparison of different wired protocol such as TCP, UDP, TCP Variants. using NS2 have been evaluated. The analysis depends on bandwidth, throughput, End to end. As the bandwidth increases efficiency increases.

ACKNOWLEDGMENT

The authors are grateful to institute for the great opportunity. we are thankful to DR.MAHESH P K, HOD, Dept of ECE . for his support and guidance and SHALINI V S, Asst. Professor, Dept of ECE, for or guiding and helping us to complete the project successfully. Special thanks to all supporting faculties and lab assistants of E&C department of ATME College of engineering for their continuous guidance and support.

REFERENCES

- [1] Nosiba Ibrahim Alfadil Altahir & Hamid Abbas Ali,” Performance Evaluation of TCP Congestion Control Mechanisms Using NS-2” Conference of Basic Sciences and Engineering Studies (SGCAC), 2016 IEEE.
- [2] M. Welzl, Network congestion control. Chichester, West Sussex, England: J. Wiley, 2005.
- [3] J. Postel, DOD standard transmission control protocol. Marina del Rey, Calif.: Information Sciences Institute, University of Southern California, 1980.
- [4] J. Postel, DoD Standard Internet Protocol. Ft. Belvoir: Defense Technical Information Center, 1980.
- [5] V. Jacobson, “Congestion avoidance and control”, ACM SIGCOMM Computer Communication Review, vol. 18, no. 4, pp. 314-329, 1988.
- [6] M. Welzl, Network congestion control. Chichester, West Sussex, England: J. Wiley, 2005.
- [7] S. Floyd, The New Reno Modification to TCP’s Fast Recovery Algorithm. International Computer Science Institute, 2004, pp. 1-18.
- [8] K. Fall and S. Floyd, “Simulation-based comparisons of Tahoe, Reno and SACK TCP”, ACM SIGCOMM Computer Communication Review, vol. 26, no. 3, pp. 5-21, 1996.