ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2012.0083

ANALYSIS OF CROSS LAYER SCHEMES TO ENHANCE TCP PERFORMANCE IN MANET

I. Sumaiya Thaseen[1], Aswani Kumar[2] and Sulavna Singh Mahapatra[3]
[1]*School of Computing Science and Engineering, VIT University, India*
E-mail: sumaiyathaseen@gmail.com
[2, 3]*School of Information Technology and Engineering, VIT University, India*
E-mail: [2]aswanis@gmail.com

Abstract:
Wireless communication is a major component of mobile computing. Transmission Control Protocol suffers from performance degradation in wireless environments. Due to high mobility and varying bit error rate in these environments, any packet loss that occurs is misinterpreted by the TCP as congestion and invokes congestion control mechanisms thereby degrading performance. Hence the performance of wireless networks is improved by introducing a cross layer design to exchange information between different layers. Cross layer optimizations produced many promising results which initiated research activity in this domain. This paper mainly focuses on cross layer proposals between network and transport layer and various TCP schemes employed to enhance performance.

Keywords:
Congestion, Cross Layer Design, MANET, TCP Reno, TCP Vegas

1. INTRODUCTION

Traditional layered architecture of the OSI reference model are organized and divided in to layers. Each layer offers services to the higher layer with a limited and well defined purpose. There exists direct coupling between physical layers and upper protocol layers. Cross layer design is an adaptive protocol design to meet the fast growing demands of wireless networking. The problems fixed locally inside the layers and optimization leads to unsatisfactory result. Cross layer design [15] increases performance by exploiting the dependencies and interactions between layers. CLD should decide which layers correspond to channel variations, which layers should be jointly optimized or designed so that scalability is achieved. Providing knowledge about physical and MAC channel conditions to routing, transport and application layers has been a promising paradigm for performance optimization in wireless systems. Wireless communications carry real time traffic such as voice traffic, video, audio, multimedia, video conferences, gaming and data traffic such as web browsing, messaging, file traffic etc. All these applications are diverse in nature and has different requirement of Quality of Service (QoS) guarantees and provide different types of traffic [18]. Hence a cross layer protocol interaction is required to increase network efficiency and provide better QoS support. Any network using wireless technology must employ the principle of cross layer design as there is a change in the state of the physical medium over time. Network throughput is highly optimized due to the exchange of information between different layers.

TCP is a reliable protocol and it is widely used and accepted for traffic that requires reliability. File transfer (FTP), remote login (TELNET), HTTP etc, uses TCP as transport layer protocol. TCP ensures guaranteed delivery of services between process to process. Congestion control algorithm is invoked whenever there is a tendency to develop congestion in the network due to heavy traffic load. Many services can be achieved in wired environment but in wireless Networks especially in MANET due to highly mobile nodes, error prone channel, dynamic topology, bandwidth constraints etc, it is difficult to achieve guaranteed delivery of services provided by TCP. The rest of the paper is organized as follows: The second section overviews the background and related work of cross layer design. Section 3 discusses the cross layer approaches and proposals. Section 4 discusses the TCP solutions to mobile ad hoc network characteristics and various TCP solutions. Section 5 includes the application of cross layer design. Section 6 concludes the paper.

2. BACKGROUND AND RELATED WORK

Transfer Control Protocol/Internet Protocol version 4 (TCP/IPv4) is today the most successful implementation of the OSI reference model and Fig.1 shows how these protocols relate to the layered network stack. Since TCP/IP is loosely based on the layered design of the OSI reference model, it also inherits its potential flaws and weaknesses. First of all, the stack design is highly rigid and strict, and each layer worries only about the layer directly above it or the one directly below it. This results in nonexistent collaboration between the different layers, presumably because no-one at that time saw any need for such a feature [1].

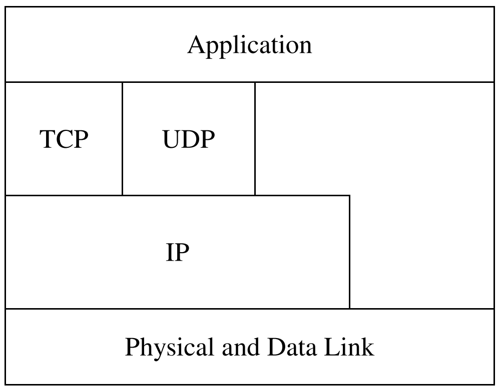

Fig.1. TCP/IP and UDP in a network stack

Fig.2 shows the structure of cross layer and communication among various layers of OSI. Routing and transport layer issues can be handled by information exchange among layers so as to optimize the performance of various layers resulting in better throughput, good broadcast latency. Cross layer design is required to achieve varying QoS guarantees, suitable to varying traffic, efficient capacity utilization, reduced processing overhead per packet. Frank Aune [1] has classified Cross layer design into evolutionary and revolutionary. Revolutionary are those that discard the existing protocols and layered architecture in favor of an entirely new design [16,17]. Such designs can be backwards compatible with existing systems, but since by definition there is less freedom in the design, the potential for gains is not as large. Among Evolutionary designs there are varying degrees of complexity and departure from the layered design. Earlier designs add coupling between layers that otherwise does not exist in the layered architecture to allow a layer to know information about the state of another layer that will help it make better decisions to improve performance. Examples such a 4G Mobile Broadband Wireless Access system and system-wide Cross- Layer Design enabled network stack are presented in the evolutionary approach; whereas Wireless Sensor Networks was discussed in the revolutionary approach case.

Most CLDs today are evolutionary. The reason for this being an evolutionary CLD is a layered structure; Revolutionary approach is not actually bounded by the existing layered architecture and also doesn't follow any compatibility. An evolutionary CLD approach prioritizes compatibility first and performance later whereas a revolutionary design does the opposite. The main reasons being compatibility and economy. Hence revolutionary approach is not favoured only in CLD but pretty much in any research or engineering task.

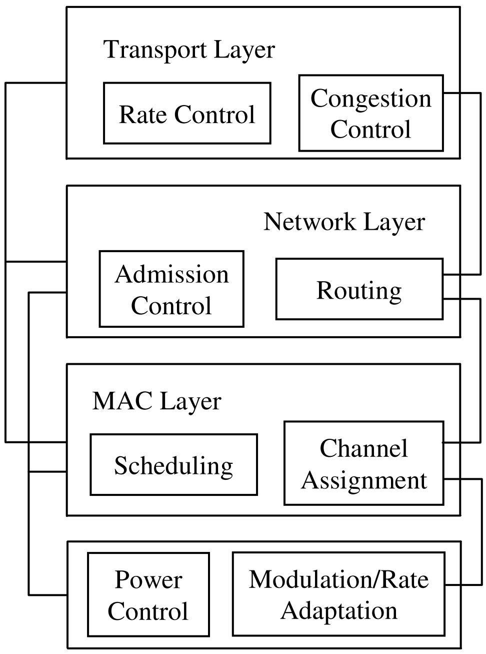

Fig.2. Framework of cross layer and its interaction

2.1 RELATED WORK

Cheng and Lin [22] proposed a cross layer design for improving the TCP end-to-end performance in multi hop wireless networks. The proposed protocol named as TCP-CL is an extension of the original IEEE 802.11 standard and TCP protocol. TCP-CL achieved a significant improvement when compared to the existing TCP Reno schemes by measuring the effective throughput. In this approach, when retransmitted packets are lost due to transmission errors, the original packets can be retransmitted by the receipt of ACK with a NAK option. TCP-CL avoids timeout problem due to the ability to react immediately to link layer corruption losses. As a result, unnecessary reduction in the number of window size gets reduced. TCP-CL overcomes frequent transmission losses, corruption losses, and ability to distinguish between congestion and transmission errors to take proper remedial action. This scheme is very advantageous for deployment in heterogenous wireless networks.

Kilazovich and Granelli [24] proposed a C^3TCP (Cross layer Congestion Control Transmission Control Protocol) in ad hoc wireless networks to obtain greater performance by observing capacity information such as bandwidth and delay at the link layer. This method adds a module with the protocol stack to adjust the outgoing data by considering the capacity. This scheme implements a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with binary exponential back off. This scheme overcomes hidden terminal problem thereby reducing data losses caused by collisions. It also estimates the delay in forward and backward direction so that the TCP sender adjusts the outstanding data to the product of bandwidth delay in the path. A cross layer collaboration scheme is used with the usage of additional module so as to avoid changes at the transport layer. End-to-end throughput is the major metric for evaluating the performance of TCP flows. There is stability in throughput during all phases of experiment. This scheme achieves results that are met with design considerations and effective utilization of the available resources.

Wang et al [21] proposed a cross layer optimization in TCP/IP networks to maximize aggregate utility over source rates provided congestion prices are considered as link costs. A distributed primal dual algorithm over the internet was designed to maximize aggregate utility. This model has been useful to understand the properties such as throughput loss, delay and fairness of large scale networks under the proposed scheme. The model is simplistic. It ignores randomness in real networks and finite duration flows. The major advantage of this scheme it solves the utility maximization problem.

Chang et al [23] designed a TCP congestion control and routing scheme using cross layer for ad hoc networks. This approach allows the ad hoc lower layer to identify most of the network events such as channel errors, buffer overflow, link layer contention including disconnections. The combination of routing algorithms and cross layer optimized TCP improves the performance of DSR and TCP.

2.2 CROSS LAYER DESIGN CHALLENGES

Cross Layer design requires additional processing and storage capabilities. Boangoat and Jarupan [3] have presented few challenges in implementing cross layer design proposals.

2.2.1 Requirement Analysis:

It is one of the major challenges in implementing cross layer design. Requirement analysis can be either application or performance oriented [3]. For example, rapid reliable communications have to be established for safety related applications. Requirements are set by system objectives in performance oriented design.

2.2.2 Implementation Strategy:

The implementation strategy involves modification of existing layers in comparison to traditional layer design [3]. Creation of new interfaces requires minimum modification as they simply rely on shared databases, whereas merging of adjacent layer or vertical calibration requires modification to existing layered design, adjusting parameters also requires joint tuning of all the layers to optimize performance as it requires higher degree of modification. They demand closer interaction between the layers.

2.2.3 Standardization of Cross Layer Design:

Standardization of protocol design helps in achieving compatibility and interoperability. Lack of standardization may lead to reduced performance [2].

3. CROSS LAYER APPROACHES AND PROPOSALS

3.1 CROSS LAYER APPROACHES

Vineet and Mehul [2] discussed about various CLD architectures. Fig.2 describes the various cross layer approaches to achieve cross layer optimization.

a. Design of New Interfaces: CLD can be implemented by designing interfaces between the layers. It can be divided into three subcategories depending on the direction of information flow along the new interfaces:
    - Upward: From lower to higher layer(s)
    - Downward: From higher to a lower layer(s)
    - Back and forth: Iterative flow between layers
b. Merging of Adjacent Layers: This will produce a new super layer and will work jointly. It is not required to create any new interface in the stack.

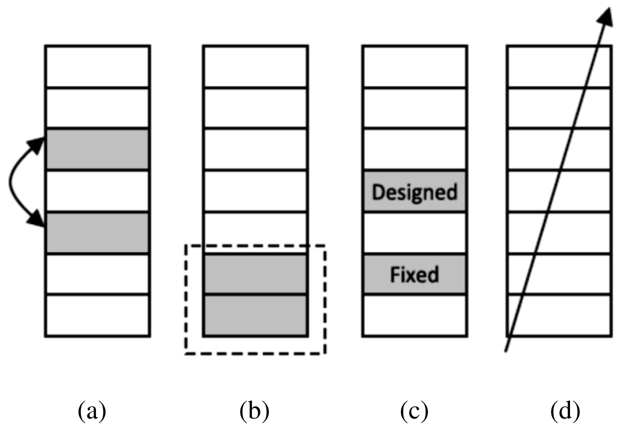

Fig.2. Cross Layer Design Approaches (a) Creation of New Interfaces (b) Merging of adjoining layers (c) Design coupling with no new interfaces (d) Vertical Calibration across layers

c. Designing Coupling without New Interfaces: This engages coupling with more than one layer at design time and no need to create any other interface for sharing data at run time. One layer is dependent on another layer and that creates a dependency between layers. The reference layer is then called as fixed layer and the new layer is called as designed layer. The designed layer will be based on fixed layer; therefore, it is not required to create clear interface between the layers.
d. Vertical Calibration across Layers: This implementation refers to changing parameters that extend across various layers. By means of cooperative tuning better performance can be obtained than changing individual parameter settings.

3.2 CROSS LAYER PROPOSALS

3.2.1 Direct Communication Among Layers:

Information sharing at run time between layers can be done by allowing the layers to communicate with each other. The layers can communicate with one other by many ways. Protocol headers allow flow of data between layers. Any other interlayer information can be sent as internal packets. E.g. Cross Layer Signalling Shortcuts (CLASS).

3.2.2 Shared Databases Among Layers:

All layers can access the shared database. The shared database is another layer providing the storage/retrieval service information to all the layers. Shared database interfaces with different layers by means of an optimization program.

3.2.3 New Abstractions:

Example for new abstractions is organizing the protocols in heap and not in stacks following a layered architecture. A higher amount of flexibility is provided during design as well as runtime.

4. TCP SOLUTIONS TO MANET ISSUES

In this section we discuss solutions available in the literature for MANET issues like Topology change, effect of path break, misinterpretation of congestion window, packet retransmission and temporal handoff.

4.1 MANET ISSUES

4.1.1 Topology Change:

Since the nature of MANET is dynamic, topology changes frequently and thereby misinterpretation of congestion control. This can be solved using TCP Feedback., TCP with Explicit Link Failure Notification, TCP Bus and Ad Hoc TCP.

4.1.1.1 TCP Feedback:

The main aim of this TCP scheme is to reduce the throughput degradation that results from common path breaks [6,4]. It maintains two states snooze and connected. In the snooze state, whenever a failure is detected by an intermediate node a RFN (Route Failure Notification) packet is sent to all the prior nodes till the source node. On reception of RFN packet it stops sending further any more packets and moves to SNOOZE state. The intermediate node is known as failure point.

In the Connected state, when the route re-establishment occurs the intermediate node sends a message to the sender and the sender moves from snooze to connected state. The intermediate node notifies the sender through the RRN (Route Re-establishment) message. Then the sender passes on all the packets to the buffer. The advantage of TCP feedback is that it is simple to implement and also permits the sender to invoke congestion control algorithm when the source is not present in snooze state. The disadvantages of the feedback scheme are it depends on intermediate nodes; too many notification packets can increase the traffic.

4.1.1.2 TCP with Explicit Link Failure Notification(Elfn):

This TCP scheme neither sends too many control packets nor maintains two states [6]. Explicit link failure notification has to be sent to the source whenever a failure is identified by an intermediate node. Source stops sending and buffers all the packets, waits for the route re-establishment to take place. Periodic probe packets are send to check for the route re-establishment if the link rejoins and then the sender transmits all the packets in the buffer to destination [4]. The benefit of this TCP scheme is that the performance is greatly improved by passing the route break information to sender thereby avoiding the invocation of congestion control algorithm. It is also less dependent on routing protocol as well as on intermediate nodes.

The disadvantage of TCP with ELFN is that the path failure is of more duration during the temporary partitioning of the network. Hence TCP with ELFN generates large number of periodic probe packets which results in wastage of bandwidth.

4.1.1.3 TCP-BUS:

TCP – BUS expanded as TCP- buffering capability and sequence information depends more on the routing protocol in comparison to TCP-F & TCP-ELFN [6]. Associativity based routing protocol (ABR) is used in this TCP scheme. Special messages such as localized query (LQ) and REPLY are used in ABR for determining a partial path. An upstream node called pivot node (PN) has to send an explicit route disconnection notification (ERDN) message whenever a path break is identified to the TCP-Bus sender. TCP-BUS source upon receiving the ERDN message has to pause the transmission and also the timers and windows. The packets in the passage at the intermediate nodes from the TCP-BUS source to the PN has to be buffered. It is the responsibility of the pivot node to identify a new partial route to the TCP-BUS destination and the information about such a route has to be notified to the TCP-BUS source by means of an explicit route successful notification (ERSN) packet. When TCP-BUS source receives an ERSN packet, it understands the packets lost in transition, resumes data transmission & retransmits lost packets. The advantage of this scheme is improved performance when compared to other schemes and fast retransmission is avoided by using techniques such as buffering, sequence numbering, and acknowledgement of selective packets. The disadvantage of this scheme is that the dependency is more due to the routing protocol and the buffering enabled at intermediary nodes.

4.1.1.4 Ad Hoc TCP (ATCP):

ATCP has a network layer feedback mechanism thus providing an alert to the TCP source based on the condition of network path [20]. The TCP sender changes to any one of the state such as persist state, congestion control state or retransmit state upon receiving information from the intermediate nodes. The ATCP puts TCP sender in the persist state when the network gets partitioned thereby avoiding unnecessary retransmissions by the TCP sender. The four states of ATCP are i) NORMAL state ii) CONGESTED state iii) LOSS state IV) DISCONN state. When a initial TCP connection is established, the ATCP is in NORMAL state. The destination TCP generates duplicate ACKs when it receives packets out-of-order. ATCP puts TCP in state so that TCP sender avoids invoking congestion control. When a new acknowledgment arrives from the TCP destination, TCP sender is removed from persist state and ATCP moves to NORMAL state. ATCP moves from LOSS state to the CONGESTED state when it receives an ECN message or an ICMP source quench message. It also removes the TCP sender from the persist state. If ECN message is received when ATCP is in NORMAL state, it moves to CONGESTED state and remains undetectable, allowing TCP to raise regular congestion control mechanisms. ATCP receives DUR message from the network layer if there is any route failure or network partition. ATCP sets the congestion window of the TCP sender to one segment. TCP sender generates probe packets sporadically to determine the connected condition of the network or path. ATCP changes to NORMAL state when the network is in connected state and it receives any duplicate acknowledgements. The advantage of ATCP is it is similar to traditional TCP and hence widely used in Internet. The disadvantage of this scheme is there is a lot of dependence on the network layer protocol to identify any route change or partitions. Another drawback of this scheme is the addition of thin ATCP layer to the TCP/IP protocol stack requires changes in the interface functions currently being used.

4.1.2 Effect of Path Break:

Whenever the path length increases there is a degradation in the TCP throughput [6]. An effective solution to this problem is Split TCP.

4.1.2.1 Split TCP:

Split-TCP overcomes the path break effect by dividing the objectives of transport layer into congestion control and end-to-end reliability [14]. This scheme divides a single TCP connection into many concatenated TCP connections. These connections are also known as segments or zones. Certain intermediary nodes are called as proxy nodes which are considered as finishing positions of short connections. It is the responsibility of the proxy node to accept the TCP packets, examine its contents, accumulate in the local buffer, and provide a acknowledgment to the sender or the earlier proxy. This type of acknowledgement is known as local acknowledgement (LACK) does not assure end-to-end delivery of packets. The proxy node has to take care of further delivery of packets.

The advantage of this scheme is improved throughput as the effective transmission path length is reduced; improved throughput fairness. The disadvantage of Split TCP is that it requires modification to existing TCP protocol. The intermediary nodes need to process the TCP packets. The performance of split-TCP is affected by path breaks or node failures that occur frequently.

***4.1.3 Misinterpretation of Congestion Window*:**

Packet losses in wireless networks need not be only due to congestion. It may also occur as a result of path break and dynamically changing topology. This misinterpretation can be avoided by Adaptive Congestion Window Limit Setting.

4.1.3.1 Adaptive Congestion Window Setting:

This involves setting the congestion window depending on the current hop count. This hop count will be obtained from routing protocol e.g. DSR. Congestion window limit is set to the product of bandwidth delay. The bottleneck is due to the bandwidth consumed by the forward path and round trip time wait in packet transmission [7]. The advantage is adaptively limiting the congestion window and thereby controlling the rates of transmission. The disadvantage is calculating the congestion window increases overhead in the network.

4.1.4 Packet Retransmission:

TCP is a reliable protocol and therefore lost packets are retransmitted. This retransmission consumes more bandwidth as described in [4].This problem can be overcome by TCP detection which uses an out of order scheme and response.

4.1.4.1 TCP - Detection of out of Order and response (DOOR):

Ziang et al [4] proposed TCP-DOOR scheme that detects and responds to packets that are not in order and thus avoiding unnecessary invoking of congestion control scheme. OOO(Out-of-Order) ordering information is added in the ACK and detects any data packet that is not in order. It notifies the sender who has invoked the congestion control algorithm. If TCP sender identifies not in order condition, any one of the following possible actions can be taken: momentarily stopping congestion control or immediate revival during congestion avoidance. Detection of not in order condition means that there is a possibility of occurrence of route change event. The disadvantage of TCP DOOR scheme is that it is not accurate in maintaining RTO consistency in TCP. This is very difficult due to its highly mobile nature, frequent path breaks are expected and RTO varies widely.

4.1.5 Temporal Handoff and Losing Packets:

Temporal handoff is a common problem in Ad hoc network. This problem can be overcome by Freeze TCP and Path recovery notification techniques.

4.1.5.1 Freeze TCP:

Freeze TCP selects one Round Trip Time (RTT) as notice duration. Once the warning period has started it sends no Window Advertisement to source so that there will be no packet loss and sender will stop sending. The problem with this approach is RTT variance. The RTT will vary because of the dynamic topology of the MANET and also it is difficult to determine the time duration from the current time till the link disconnect duration [5].

4.1.5.2 Path Recovery Notification:

The TCP receiver notifies the sender about the temporal link disconnection through special ACK (SACK) to avoid packet loss. The sender buffers all the packets when the link re-connects again the receiver notifies the sender about it through ACK. The sender then transmits all the buffered packets. This mechanism requires special modification in SACK [5].

4.2 PERFORMANCE COMPARISON OF TCP RENO AND TCP NEW RENO OVER MANETS

TCP Reno is the scheme that has been extensively used in Internet [6]. TCP Reno operates in four phases: Slow Start, Congestion Avoidance, Fast Retransmit and Fast Recovery. Slow start phase is initiated when the transmission begins or when a loss is identified. Congestion avoidance phase is initiated when the slow start phase ends or after identifying any loss by the copy acknowledgements and stops if the window size is the upper limit of packets the target recevier can receive. The Fast Retransmission and Fast Recovery begin mutually whenever the third duplicate acknowledgement is received. TCP New Reno overcame the disadvantage of TCP Reno because it was not able to recover more packet losses. Time out affects the throughput because connection has to wait for time out to occur and cannot send data during that period of time. In TCP Reno, after receiving partial ACKs, it comes out of fast recovery and no option left but time out to occur. TCP New Reno introduced a new concept known as fast retransmission phase beginning with the detection of packet losses and ending when the receiver acknowledges that all data has been received at the end of the phase. As the acknowledgement arrives in the transmitter side in a random manner, TCP New Reno sometimes may direct unnecessary retransmissions. The major advantage is that it avoids unnecessary timeouts of the transmitter thereby recovering from multiple packet losses which is very important for wireless ad hoc networks [10].

4.2.1 Duplicate Acknowledgement and Fast Retransmit:

A TCP sender detects a packet loss when it has timed out and waiting for an ACK. Whenever the TCP receiver receives a segment that is not in order, and then it has to send back a duplicate acknowledgement to the source. DUPACK indicates next BYTE number expected. Losses can be detected by Timeout waiting for an ACK and DUPACK due to out of order segment.

4.2.2 Fast Retransmission:

DUPACK’s are used to formulate retransmission decision if the number of duplicate acknowledgement=3, the sender assumes losses and retransmits. There can be two reasons for receiving DUPACK 1) either the packet has received but not in order. It is delayed due to Congestion in the network. Therefore, Jacobson waits for 3 DUPACK and then performs retransmission decision with 3 DUPACK which assures that the segment has been surely lost and needs retransmission. The segment is retransmitted without waiting for the retransmitted timer to go off and retransmits before RTO.

4.2.3 TCP Vegas:

TCP Vegas is a congestion control mechanism based on delay. TCP Reno implements a binary congestion signal to vary the size of window whenever any packet loss occurs whereas Vegas uses signal which is very fine grained, queuing delay, to overcome congestion. Vegas also out performs TCP New Reno because it is better with respect to network utilization, fairness, throughput and packet loss.

4.3 COMPARISON OF TCP VEGAS AND TCP RENO OVER MANET

4.3.1 TCP Reno:

TCP Reno manages the amount of data that is to be sent using a congestion window (CWND) in a round-trip Time [RTT] and a maximum window [MWND] that is initialised by the destination limited to the maximum value of CWND. There is an exponential increase in the window size when TCP Reno is in the slow-start phase, packets sent at increasing speed causes network congestion. This can be overcome by beginning the congestion avoidance phase begins whenever CWND goes beyond a predefined slowstart threshold [ssthresh] value [10].

4.3.2 TCP Vegas:

TCP Vegas implements a bandwidth estimation scheme that avoids congestion. The measured RTT is used to exactly determine the number of data packets that the source is about to send. Fig.3 describes the state transition diagram of Vegas consisting of three phases and the respective conditions that causes the TCP to change from one state to other. The major improvements of Vegas in comparison to TCP Reno are modification of slow-start, congestion avoidance and in addition a new transmission mechanism.

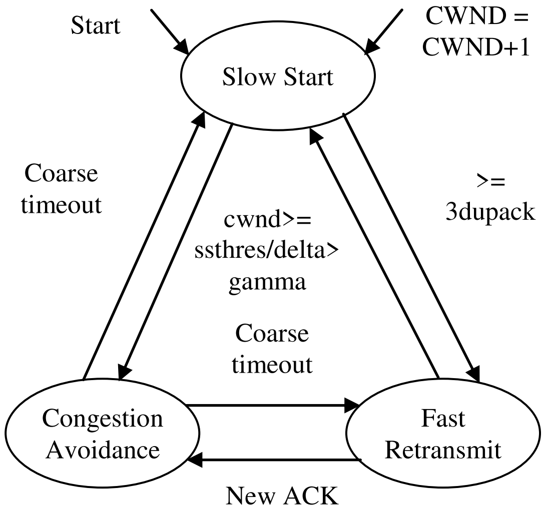

Fig.3. State transition diagram of Vegas

4.4 COMPARISON OF TCP-CL (CROSS LAYER) AND TCP RENO

TCP-CL [22] is an extension of the original IEEE 802.11 standard and the TCP protocol. This protocol improves the performance of TCP in multi hop wireless network. TCP-CL is compared with TCP Reno in terms of effective throughput. TCP-CL adjust the congestion window size based on the ACK received from the receiver according to the following function i) receipt of new ACK ii) receipt of NAK iii) receipt of duplicate ACK iv) upon timer expiry. The goodputs of TCP-CL and TCP Reno are compared over paths of increasing length. The goodput considers only the bytes delivered to the receiver. Throughput decreases as the number of hops increases. TCP-CL provides better response to packet losses than TCP Reno due to the support of extended link layer protocol. TCP Reno suffers frequent coarse-grain timeouts. TCP-CL does not require any node to cache any unacknowledged packet for every TCP connection. This scheme is very advantageous for deployment in heterogeneous wireless networks.

4.5 COMPARISON OF C^3 TCP AND TCP VEGAS

C^3TCP [24] introduces a Congestion Control Module (CCM) attached to the link layer of end nodes of the TCP connection.CCM dynamically adjusts TCP congestion window specifying its desired size by the received advertiser window (RWND) field of the TCP header. C^3TCP achieves good end-to-end throughput and keeps the throughput level close to the available bandwidth thereby utilizing the link capacity. Stable behaviour is observed when compared with the TCP Vegas flows.

5. APPLICATION OF CROSS LAYER DESIGN

Cross layer design has been implemented on a wireless LAN for Telemedicine Video Transmission. This design consists of one medical specialist in a expert station connected to some patient stations through an access point of WLAN IEEE 802.11g. This application involves data, video and voice to examine patients. CLD optimizes the existing OSI architecture thereby providing efficient communication between layers for the selection of an optimal solution. This kind of optimization is required for the system to adapt to wireless location and also support QoS for Telemedicine Video Application. Guaranteed bandwidth for connection requests is assigned for the telemedicine application by performing cross layer design of existing WLAN protocol stacks. Parameter abstraction is a process of gathering important information. The obtained information has to be optimized to achieve QoS requirements for telemedicine application.

6. CONCLUSIONS

Cross layer design is an emerging paradigm shift evolved to solve most of the issues in MANETs. As MANETs are characterized by frequent node mobility, unreliable network connections and distributed in nature there is a need for cross layer design optimization to enhance TCP performance. This paper presents the challenges in implementing cross layer design, various cross layer proposals, various issues in routing and transport layers and TCP solutions to resolve those issues. CLDs are employed in wireless networks to pass information from one layer to another in a cross-layer fashion at runtime to detect packet losses and determine the appropriate reason for packet loss in MANETs. In an effort to increase the performance of TCP, it is important to realize the design of layers and introduce some changes to existing architecture while maintaining the standardization and modularity of layered architecture. It can also be concluded from various studies that TCP Vegas outperforms TCP Reno as well as New Reno in many ways. But TCP Reno is widely accepted as TCP Vegas and TCP Reno are incompatible. Techniques like CODE-TCP can be implemented to improve the performance. Recently developed schemes such as TCP-CL outperforms TCP Reno in comparison with the measured good put. C^3TCP achieves good end to end throughput when compared with TCP Vegas. The various issues in MANET's are discussed and various TCP schemes addressing these issues are also explored.

REFERENCES

[1] Frank Aune, "*Cross-Layer Design Tutorial*", Technical report, Norwegian University of Science and Technology, Creative Commons License publishers, 2004.

[2] Vineet Srivastava and Mehul Motani, "Cross-Layer Design: A survey and the road ahead", *IEEE*

Communications Magazine, Vol. 43, No. 12, pp. 112-119, 2005.

[3] Boangoat Jarupan and Eylem Ekici, "A survey of cross-layer design for VANETs", *Ad Hoc Networks*, Vol. 9, No. 5, pp. 966-983, 2011.

[4] Xiang Chen, Hongqiang Zhai, Jianfeng Wang and Yuguang Fang, "TCP performance over mobile ad hoc networks", *IEEE Canadian Journal of Electrical and Computer Engineering*, Vol. 29, No. ½, pp. 129-134, 2004.

[5] Myungjin Lee, Monsoo Kang, Myungchul Kim and Jeonghoon Mo, "A cross-layer approach for TCP optimization over wireless and mobile networks", *Computer Communications*, Vol. 31, No. 11, pp. 2669-2675, 2008.

[6] C Siva Ram Murthy and B S Manoj, "*Ad Hoc Wireless Networks: Architectures and Protocols*", Pearson Education, 2004.

[7] Ka Cheong Leung, Victor O K Li and Daiqin 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.

[8] Yi Cheng Chan, Chaia Liang Lin, Chia Tai Chan and Cheng Yuan Ho, "CODE TCP: A competitive delay-based TCP", *Computer Communications*, Vol. 33, No. 9, pp. 1013-1029, 2010.

[9] Hai Jiang, Shixin Cheng and Xiangning Chen, "TCP Reno and Vegas Performance in Wireless Ad Hoc Networks", *IEEE International Conference on Communications*, Vol. 1, pp. 132-136, 2001.

[10] Lianghui Ding, Xinbing Wang, Youyun Xu and Wenjun Zhang, "Improve throughput of TCP-Vegas in multihop ad hoc networks", *Computer Communications*, Vol. 31, No. 10, pp. 2581-2588, 2008.

[11] M Rossi, R Vicenzi and M Zorzi, "Accurate Analysis of TCP on Channels with Memory and Finite Round-Trip Delay", *IEEE Transactions on Wireless Communications*, Vol. 3, No. 2, pp. 627-640, 2004.

[12] M. Zorzi, A Chockalingam and Ramesh R Rao, "Throughput Analysis of TCP on Channels with Memory", *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 7, pp. 1289-1300, 2000.

[13] Xin Yu, "Improving TCP performance over mobile ad hoc networks by exploiting cross layer information awareness", *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pp. 231-244, 2004.

[14] S Kopparty, S V Krishnamurthy, M Faloutsos and S K Tripathi, "Split TCP for mobile ad hoc networks", *Proceedings of the IEEE Global Communications Conference*, Vol. 1, pp. 138-142, 2002.

[15] S Shakkottai, T S Rappaport and P C Karlsson, "Cross-layer design for wireless networks", *IEEE Communications Magazine*, Vol. 41 No. 10, pp. 74- 80, 2003.

[16] I F Akjildiz, M C Vuran and O B Akan, "A cross-layer protocol for wireless sensor networks", *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, pp. 1102-1107, 2006.

[17] M Slavik, I Mahgoub, A Badi and F N Sibai, "A functional component based framework for cross-layer design", *Proceedings of the 4th International Conference on Innovations in Information Technology*, pp. 372-376, 2007.

[18] V T Raisinghani and S Iyer, "Cross-Layer Design Optimizations in Wireless Protocol Stacks", *Computer Communications*, Vol. 27, No. 8, pp. 720–724, 2004.

[19] R Madan, S Cui, S Lal and A Goldsmith, "Cross-Layer Design for Lifetime Maximization in Interference- Limited Wireless Sensor Networks", *IEEE Transactions on Wireless Communications*, Vol. 5, No. 11, pp. 3142-3152, 2006.

[20] Jian Liu and Suresh Singh, "ATCP: TCP for Mobile Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 19, No. 7, pp. 1300-1315, 2001.

[21] Jiantao Wang, Lun Li, S H Low and J C Doyle, "Cross-Layer Optimization in TCP/IP Networks", *IEEE/ACM Transactions on Networking*, Vol. 13, No. 3, pp. 582-595, 2005.

[22] Rung Shiang Cheng and Hui Tang Lin, "A Cross-layer Design for TCP End-to-End Performance Improvement in multi-hop wireless networks", *Computer Communications*, Vol. 31, No. 14, pp. 3145-3152, 2008.

[23] Hsung Pin Chang, Hui Wen Kan and Ming Hsin Ho, "Adaptive TCP congestion control and routing schemes using cross-layer information for mobile ad hoc networks", *Computer Communications*, Vol. 35, No. 4, pp. 454-474, 2012.

[24] D Kilazovich and F Granelli, "Cross layer congestion control in ad hoc wireless networks", *Ad Hoc Networks*, Vol. 4, No. 6, pp.687-708, 2006.