In this research, we investigate about the cross-layer interaction between TCP and routing protocols in the IEEE 802.11 ad hoc network. On-demand ad hoc routing protocols respond to network events such as channel noise, mobility, and congestion in the same manner, which, in association with TCP, deteriorates the quality of an existing end-to-end connection. The poor end-to-end connectivity deteriorates TCP’s performance in turn which significantly reduces the total network throughput decrease and our total network lifetime is also reduced. The concept of additive increase multiplicative decrease the congestion in low bandwidth delay products. Then, to address these problems, we propose complementary mechanisms that is, the TCP fractional window increment (FEW) scheme and the Route-failure notification using bulk-loss Trigger (ROBUST) policy and hybrid medium access control. The TCP FEW scheme is a preventive solution used to reduce the congestion-driven wireless link loss. The ROBUST policy is a corrective solution that enables on-demand routing protocols to suppress overreactions induced by the aggressive TCP behavior. The hybrid medium access control used along with TDMA Multiple hop in a single medium access control, End to End quality of service, Latency sensitive traffic flows, and high priority channel access. It is shown by computer simulation that these mechanisms result in a significant improvement of TCP throughput without modifying the basic TCP window or the wireless MAC mechanisms.

Index Terms: TCP, on demand ad hoc routing protocol, IEEE802.11, hybrid medium access control HMAC, Cross layer interaction.

1. INTRODUCTION
The IEEE802.11 is predominantly used in multi hop topology such as wireless mesh networks etc. When IEEE802.11 is being integrated with ad hoc networks it suffers from channel noise, node mobility, congestion, routing failure, low frequency reuse and limited bandwidth. It has been observed that TCP suffers poor bandwidth utilization and extremely unfairness in the IEEE802.11 environment. Since TCP is most dominantly used in many routing protocols Interoperability with wired and wireless network has been examined based on TCP friendly equations and long lived TCP flows. Based on TCP operational range analysis we observed TCP with small congestion window outperforms a TCP with a larger one in IEEE 802.11 [1]. So we conducted a quantitative analysis based on Grid Topology, Mobility Topology for static and dynamic routing based on TCP traffic for IEEE802.11 [1] It was shown when HMAC was used in sensor networks it yields an higher performance metrics similarly this concept was implemented in MANETS for contention based and contention free algorithms [2]. Comparisons was done for IEEE 802.11 and HMAC for throughput, per hop basis, packet arrival ratio, packet delivery ratio and routing control overhead [2].

1.1. Interaction Of TCP And The Core Layers
When there is network overload in TCP it informs the MAC layer as congestion this is being misinterpret as routing failure to network layer which tries to outperform it by rerouting, error messaging. Simultaneously when routing failure occurs there occurs TCP connection failure and TCP time out and finally TCP restart occurs [1].

Fig 1.1: Network Layer Affect TCP Performance

*(Assistant Professor)*
2. RELATED WORKS
To overcome this TCP ELFN was introduced to distinguish congestion from routing change in a wireless environment [3] [4]. The concept of TCP ECN was used from preventing congestion window from growing too much along with the AIMD (additive increase multiplicative decrease) which is too aggressive for a network with low bandwidth. To avoid congestion or routing failure prior to transmission handshaking signals was introduced RTS/CTS but this was applicable when A and C were in the transmission range of one and another (classical hidden terminal problem). But latter in extended hidden terminal problem this seemed to be a failure because they were not within the transmission range of a central node.

3. SYSTEM MODEL PROPOSED
The framework for MANETS is established by combining IEEE802.11 with ad hoc networks. By using HMAC along with TDMA approach Wake up slots and Data slots are issued for Channel assignments for contention free and contention based networks. The ROBUST algorithm is used to find link failure whether it is within tolerable limits based on which DSR is activated. The FEW algorithm is implemented by which the congestion windows size is set depending on which it grows fractionally or linearly. Finally it is optimized by cross layer approach in which routing failure occurs it freezes congestion window size and congestion occurs it is informed to network layer to stop finding new routes. So we finally obtain a good performance metrics by providing QOS and increase in throughput.

3.1. Data Link Layer Affect TCP Performance And Recovery Algorithms Used
TDMA is allocated based on slotted aloha basis here conflict arises for scheduling. Negative acknowledgement and retransmission occurs at the MAC layer it calculates bit error rate, packet loss ratio based on per hop basis and energy consumption is being calculated. Finally sleep wake scheduling algorithm is being carried on [7].
4.1. TCP FEW (Fractional Window Increment)

The congestion-driven link loss can be clearly distinguished from channel noise or mobility errors, the TCP behavior can be improved accordingly. However this depends on time out factor ($K$), windows growth rate ($\alpha$). The chosen proper $K$ and $\alpha$ values to control the TCP operation range while preserving the basic TCP window mechanism $K = 0$ and $\alpha = 1$ provide the upper bound for the shifted TCP operation range. Thus, we have this curve has a lower loss rate than the curve with $K = 0$ with respect to a fixed window value.

This is equivalent to adding one packet to the window size at every $1/\alpha$ round-trip time. Suppose that the current congestion window size is $W$, the TCP sender sends $W$ packets at every round-trip time and receives $W$ ACKs during one round-trip time from the TCP receiver. At every ACK reception, the TCP sender updates $W$ by,

$$w_{\text{new}} = w_{\text{current}} + \left(\frac{\alpha}{W}\right) \text{ "equation 1".}$$

This is possible only when the probing traffic is mild. However in the case of IEEE802.11 traffic is not so mild

4.2. TCP Robust (Route Failure Bulk Loss Trigger Policy)

Which is a simple link loss reaction policy for on-demand routing is following Node mobility, congestion and channel noise.

$\beta$ represents on demand routing protocols to control connection stability. The number of successive link failures is denoted by $L$

![Flowchart for ROBUST](image)

Fig. 4.2: Flowchart for ROBUST

When a packet loss occurs due to routing failure. If the no of successive link failure is within tolerable successive link failure limits the current packet is dropped, next packet is also transmitted using the same route but the successive link failure is increased to one. If the no of successive link failure is more than the tolerable successive link failure DSR protocol is activated which self configures and self stabilizes in which next route is formed and packet is being transmitted and the successive link failure value is set to zero.

4.3. TCP HMAC (Hybrid Medium Access Control)

To overcome the difficulties of data link layer we use HMAC. Previously Schedule based algorithm was used it had a central node to manage and broadcast the schedule to other nodes but it exposed failures to entire network itself. So we proposed contention based approach since it can be used in a distributed fashion. It had drawbacks like idle listening period to overcome this we made sleeping mode.
**International Journal of Computer Engineering**

**Fig. 4.3: Flowchart for HMAC**

Contention based allows many users to access the same radio channel without pre coordination

1. Each node turns on its radio during its own wake up slot and sleeps during all other wake up slots
2. Each sender randomly picks up a data slot and announces the data slot number along with the receiver node identifier via a wake up message in the receiver wake up slot
3. Upon reception of a WAKE UP message a node checks the embedded node identifier in the wake up message. If its intended receiver, then the node turns on its radio for the incoming data packets for the specified data slot otherwise it just sleeps
4. If any collision occurs in a node’s wake up slot, then the node turns on its radio for duration long enough to receive an RTS packet at the beginning of each data slot for a possible incoming data packet. If the node learns that is the intended receiver from the received RTS message, and then it keeps the radio on to receive the data packet; otherwise, it returns to sleep in the remaining period of the data slot. This way, a node can minimize the extra energy cost under such a situation.
5. In each data slot, uni-cast data transmission must follow the well-known RTS/CTS/DATA/ACK scheme in IEEE802.11 to avoid the “hidden terminal problem,” since two senders may choose the same data slot to send data to their receivers at the same time, and the transmissions happen to be in a common interference range

HMAC also provides support for one-hop broadcast operation. When a node has data to broadcast, it sends out a “WAKEUP” message containing a broadcast address and a data slot in each wake up slot. After receiving such “WAKEUP” messages, all neighbors will wake up in the same data slot to receive the broadcast message. In addition, different from uni-cast operation, when the sender finds an idle channel during the CS period in the chosen data slot, it will immediately send out the broadcast message without following the before mentioned RTS/CTS/DATA/ACK scheme.

**5. RESULTS AND DISCUSSIONS**

**GRID TOPOLOGY (static)**

**Fig. 5.1: Time vs Congestion Window**

The graph shows the grid topology of broadcasting based on TCP congestion for providing network stability. For a minimal congestion the congestion window value is chosen as 20 packets and plotted in Y axis and time duration taken in X axis.

**Fig. 5.2: End to End Distance vs Throughput**

Even as the end to end distance or the number of hops increases, the throughput remains constant without any degradation using FEW and ROBUST as shown in the graph.


**MOBILITY TOPOLOGY (dynamic)**

![Time Vs Congestion Window](image1)

**Fig. 5.3: Time vs Congestion Window**

The mobility topology graph shows the change of congestion window size which is taken along Y axis (with a maximum value of 24 packets) and the time in seconds taken along X axis (with time duration of 2 seconds).

![End to End Distance vs Throughput](image2)

**Fig. 5.4: End to End Distance vs Throughput**

The Mobility topology graph shows the end-to-end vs. throughput for the transmission range to calculation based on the time.

**COMPARISONS OF IEEE802.11 WITH HMAC**

![Routing Control Overhead](image3)

**Fig. 5.5: Routing Control Overhead**

The graph shows the total no of packets that are redundant (*i.e.*, total no of route request and route reply). The first packet arrival is checked after 10 seconds and this precedence is carried after every 10 seconds.

![End to End Delay Based on Per Hop Basis](image4)

**Fig. 5.6: End to End Delay Based on Per Hop Basis**

The graph shows the one hop neighbor preceded by its following neighbors with delay taken in milliseconds and per hop taken in seconds.

![End to End Delay Based on Packet Arrival Interval](image5)

**Fig. 5.7: End to End Delay Based on Packet Arrival Interval**
The graph shows the packet arrival interval in seconds with delay taken in milliseconds.

Fig. 5.8: Throughput

The graph shows the total no of packets arrival with respect to time in seconds.

Fig. 5.9: Packet Delivery Ratio

Pdr = \[ \frac{\text{total no of packets received}}{\text{total no of packets sent}} \] * 100.

6. CONCLUSION

Thus the TCP performance in IEEE802.11 adhoc network is a complicated cross layer problem. The cross layer problem between TCP and routing protocols in IEEE802.11 ad hoc networks was investigated. Based on well known TCP friendly equation, we conducted a quantitative study on TCP operation range using static and long lived TCP flows. Based on the analysis we propose three simple yet effective means and implemented those TCP Few, TCP ROBUST and TCP HMAC, to improve the system performance. It was shown via computer simulation that TCP performance can be significantly improved without modifying the basic TCP window or the wireless MAC mechanism.

The improvement achieved by TCP FEW demonstrates that simply limiting such heavy traffic can help improve the overall system performance.

REFERENCES


