CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

This chapter presents an overview of Heterogeneous Wired-Wireless Networks (HWWN) and the importance of congestion control in HWWN. It explains the different types of congestion control mechanism and different phases of congestion control mechanism. It continues with the issues of congestion avoidance algorithm in the Heterogeneous Wired-Wireless Networks. The motivation, problem statement and objectives of this research are discussed in Section 1.6, 1.7 and 1.8 respectively. Finally the overview of proposed system and an outline of the thesis structure are given.

1.2 HETEROGENEOUS WIRED-WIRELESS NETWORKS

The revolutionary advances in the communication technologies span across the variety of networks such as wired network, wireless network, high speed network, satellite network etc. Connecting these networks together form the heterogeneous environment in which the protocols and accessing technologies of these networks are different from each other. Now-a-days, the wireless networks are common and wireless links are everywhere in today’s networks. Although wireless networks are getting popular, the wired networks are still existence. This is because the bandwidth and reliability is very low in wireless network. It introduces a technological evolution which inspires the researchers to envision the Heterogeneous Wired-Wireless networks (HWWN). A typical HWWN architecture is shown in Figure 1.1. The HWWN architecture has distinct communication paradigms and challenges imposed by their unique characteristics. For example the asymmetric
networks such as Asymmetric Digital Subscription Line (ADSL), Digital Video Broadcast (DVB), and Packet Radio Network (PRN) have difference in transmit and receive capacity of the link. The satellite network takes high turn-around-time and the wireless network experiences high Bit-Error-Rate than the traditional wired networks. Consequently HWWN has emerged researchers to redesign the protocol stack and the protocol.

Considering the design of protocol stack, the TCP/IP was first developed in the late 1960s and OSI Reference model in the mid-1970s, to show how to facilitate communication between two different systems [1][2]. Both protocol stack models are organized as a set of functional layers. The layers of protocol stack consist of protocols which are associated with specific task like congestion control, routing, flow control, error control, access control, modulation etc. These protocol stack models are robust enough to meet the needs of wired networks and flexible enough to the constantly changing technology of wired networks. But the growing demand of data communication has led to rapid development of network technology like Cable TV, Digital Video Broadcasting (DVB), Asymmetric Digital Subscriber Line (ADSL), Ad hoc Network, Wireless Network, Sensor Network, Satellite Network, High Speed Network etc. Therefore many protocol stack models were proposed for various networks, for example ATM Reference model [3], Wireless

**Figure 1.1 Heterogeneous Wired-Wireless Network**

![Diagram of heterogeneous wired-wireless network](image-url)
Sensor Network Protocol Architecture [4], Cross-Layer Architecture (CLA) [5], GSM Protocol Architecture [6]. Cross-Layer Architecture (CLA) maintains the compatibility with all other protocol stack models since it is an extension of a protocol stack. The idea behind this approach is that the information used in a layer has partial or high impact on the task associated with another layer. Hence the information can be shared with adjacent or non-adjacent layers which results in significant improvement in the performance of protocols [7-9].

Considering the design of protocol, it is necessary to take into account the communication media, its characteristics and connecting hardware because they differ from network to network. In case of the protocol supporting the reliable data service in the HWWN, congestion control mechanism of Transmission Control Protocol (TCP) is facing serious performance problem [10-13]. This is because of high speed links (Optical Fibers networks), long and variable delay links (Satellite networks), lossy links (Wireless networks) and asymmetric links (Hybrid satellite networks) embedded with the Internet. Many TCP variants were proposed for improving the performance in the heterogeneous network to cope with different network environments [13-19]. The difference between these TCP variants relies on the mechanism used to control the congestion in the network.

1.3 TYPES OF CONGESTION CONTROL MECHANISM

The Congestion control mechanism used in the different TCP variants can be classified into four broad categories:

1. Loss-Based Congestion Control [19-22]
2. Delay-Based Congestion Control [23-26]
3. Hybrid Loss-Delay-Based Congestion Control [27-32]
4. Cross-Layer Based Congestion Control [15][33-36]

Loss-based congestion control uses reactive strategy in its congestion avoidance algorithm which reacts after the occurrence of packet loss. It takes the loss of packets as a signal for congestion in the network. It has no way of detecting the
incipient of congestion i.e. before occurrence of packet losses. Delay-based congestion control uses proactive strategy which reacts before the occurrence of packet loss. It takes the variation in the packet delay as a signal to detect the incipient of congestion in the network. Hybrid Loss-Delay-based congestion control combines the reactive and proactive strategy together. Cross-Layer-based congestion control takes the information from the non-adjacent layers of protocol stack to control the congestion. This is because the implicit information used for congestion control such as round-trip time, packet-in-flight, bandwidth, queuing time and queue size are not sufficient to detect the congestion. Hence the explicit information such as path changes, link failures and received signal strength from non-adjacent layers of protocol stack would be useful to detect and control the congestion in the network. Therefore information from physical layer, MAC layer and network layer can be shared with congestion control mechanism to detect the onset of congestion in the network.

In general, the congestion control mechanism has four phases which is explained in the following section. These four phases are used to control the data flow in the network.

1.4 PHASES OF CONGESTION CONTROL MECHANISM

The congestion control mechanism of TCP has the following four phases [37][38]:

1. Slow-Start phase (SS)
2. Congestion Avoidance phase (CA)
3. Fast Retransmit phase (FRXT) and

1.4.1 Slow-Start Phase (SS)

The Slow-Start (SS) is based on the idea of the congestion window size (cwnd). The cwnd determines the number of packets (bytes) that can be sent into the network. In other words, it denotes the current estimate of available capacity of
network. When a connection is established between the sender and receiver, the sender doesn’t know the available capacity of network. Therefore the congestion window is initialized to one packet. The default size of the packet is 512 bytes. The sender starts by transmitting one packet and waiting for its Acknowledgement (ACK). When a ACK is received, the congestion window size is incremented from one to two packets and two packets are transmitted as shown in Figure 1.2. When each of these packets is acknowledged, the congestion window size is increased so that four packets are transmitted. In this way, the congestion window size is increased exponentially. But the sender cannot increase the cwnd indefinitely. When the congestion window size of sender reaches a threshold value called SSThresh, SS stops the exponential growth of congestion window and switches to congestion avoidance phase. The initial value of SSThresh is set arbitrarily high (e.g. to the size of the largest possible advertised window obtained from the receiver during the connection establishment) because the sender doesn’t know the network capacity. The exponential growth of congestion window size at the SS phase is shown in the Figure 1.2.

![Figure 1.2 Illustration of Slow-Start Phase of Congestion Control Mechanism](image)

1.4.2 Congestion Avoidance Phase (CA)

Once the congestion window size reaches the SSThresh value and there is no packet loss in the network, SS phase switches to congestion avoidance phase which starts the linear growth of congestion windows size in order to utilize the available capacity. In the linear growth, the congestion window size is incremented
by one packet for every ACK. At some point, the bandwidth of link between the sender and receiver nodes and the buffer at router are fully utilized. It means that the congestion window size has gotten too large. Increasing the size of cwnd further causes packet loss at the buffer of router placed in between the sender and receiver nodes. In the sender point of view, the entire network is considered as black box because it doesn’t know what is happening inside of the network. Hence, the sender knows the condition of network based on some assumptions made at sender or the information obtained from the receiver. One of the basic assumptions made in TCP at sender side is that any packet loss in the network is an indication for congestion. If the sender detects the packet loss due to congestion in the network, then the SSThresh is reduced to half of the cwnd and cwnd is reset to one [20]. The packet loss in the network is detected by either reactive strategy or proactive strategy.

In case of reactive strategy, the loss of packets in the network is taken as a signal of congestion. The loss of packet at the receiver is identified by tracking the sequence number of the packets. The sequence number is a number generated by the sender and assigned to each packet that is being sent. Suppose any packet loss is identified by the receiver node, the ACK of previous packet is sent to the sender 3 times which is called triple duplicate acknowledgements (TDA) as shown in Figure 1.3. An ACK is called duplicate acknowledgement, if the packet is already acknowledged at the sender and the receiver is again sending the same ACK [20][38]. The purpose of the duplicate ACK is to let the sender know that a packet is lost or received out-of-order. Since the sender does not know whether a duplicate ACK is caused by a lost packet or out-of-order packet, it waits until receiving three duplicate acknowledgements. Upon receiving the TDA by the sender, it decides that there is a packet loss in the network. Therefore the congestion avoidance algorithm of sender reduces the congestion window size which results in reduction of sending data rate.

In case of proactive strategy, the onset of congestion is detected by the variation in the packet delay. The delay variation is measured by the Round-Trip Time (RTT). The round-trip time of a packet is defined as the time required for a packet to travel from sender to receiver and back again. The buffering of packets at
the intermediate node, somewhere in between the sender and receiver, increases the RTT of a packet. The proactive strategy assumes this packet delay as a signal of congestion and reduces the congestion window size before the packet loss at the intermediate nodes [25]. It avoids the unnecessary retransmission of packets. Suppose any packet loss or out-of-order packet is detected by the receiver, it sends duplicate acknowledgements.

1.4.3 Fast-Retransmit Phase (FRXT)

The fast retransmit is a technique used to retransmit the packets based on the expiry of Retransmission-Time-Out (RTO) and three duplicate acknowledgement (TDA) [20][38]. The RTO defines the time period that how long the sender can wait for the acknowledgement of a packet from receiver.

When a packet is lost in the network or delivered at receiver out-of-order, the receiver informs it to the sender by sending the duplicate ACKs as shown in Figure 1.3. The out-of-order packet means that the receiving order of the packets differs from its sending order due to multipath routing. Therefore the arrival of duplicate ACKs is an indication that the packet has been lost or out-of-order delivery of packet. When an out-of-order packet arrives, the receiver sends an immediate duplicate ACK. The purpose of this duplicate ACK is to inform the sender that a packet was received out-of-order. When first duplicate acknowledgment is received by the sender, it checks whether the RTT of a packet is less than the RTO.

If the RTT is greater than RTO, then it retransmits the packet without having to wait for the RTO to expire. This process speeds up the recovery of packet loss. At this point, the SSThresh is set to cwnd/2 and the cwnd is set to SSThresh. Suppose the RTT is greater than RTO, then it retransmits the packet without having to wait for third
duplicate acknowledgment. Expiry of RTO also indicates the sign of packet loss. Therefore the SSThresh is set to cwnd/2 and the cwnd is set to 1.

1.4.4 Fast-Recovery Phase (FRCV)

The Fast-Recovery phase allows the high throughput under moderate congestion. Receiving duplicate acknowledgment does not mean that there is some serious network congestion because having duplicate ACK implies that data is still flowing in the network. Therefore instead of going all the way back to SS phase, sender resumes transmission with a larger window and starts increasing it gradually as in the case of congestion avoidance. This results in increased throughput. This is referred to as Fast-Recovery phase.

After sending the missing packet, the Fast-Recovery Phase governs the transmission of new packet until a non-duplicate ACK arrives. On the first and second duplicate ACKs, the sender sends a packet of previously unsent data. When
the third duplicate ACK is received, the SSThresh is set to the value of \( \text{FlightSize} / 2 \) [20]. The FlightSize is the amount of outstanding packets in the network. When the next ACK arrives that acknowledges previously unacknowledged data, the sender sets cwnd to new SSThresh which deflates the congestion window size. The number of packets transmitted upon receiving the ACK is set to half the number of outstanding packets until all the lost packets are repaired. Finally, after all lost packet has been successfully retransmitted, the congestion window is set to no more than SSThresh and congestion avoidance is used to further linear increase in congestion window.

![Figure 1.4 Illustrations of Four Phases of Congestion Control Mechanism](image)

**Figure 1.4 Illustrations of Four Phases of Congestion Control Mechanism**

Figure 1.4 shows the evolution of congestion window size with respect to different phases of congestion control mechanism [13]. At the slow start phase, the congestion window size is increased exponentially until reaching the SSThresh. After that, it switches over to congestion avoidance phase where the congestion window size is increased linearly until the detection of packet loss in the network. The packet
loss is an indication of congestion in the network. The congestion in the network is detected either using reactive strategy or proactive strategy. The triple-duplicate-acknowledgement (TDA) or retransmission time-out is used to detect the congestion in the network reactively and implicit information like packet delay, bandwidth, queue size etc. is used to detect the inception of congestion proactively. If the congestion is detected, it enters into the FRXT and then FRCV phases of congestion control mechanism. Suppose the congestion is detected by retransmission time-out, the SSThresh and cwnd is updated as follow: SSThresh = cwnd/2 and cwnd = 1. If the congestion is detected by TDA, the SSThresh and cwnd is updated as follow: SSThresh = cwnd/2 and cwnd = SSThresh. Once the SSThresh and cwnd are updated after the packet loss, SS phase is again started with new SSThresh value and cwnd [38].

1.5 ISSUES ON CONGESTION AVOIDANCE ALGORITHM

Among the four phase of congestion control mechanism, the congestion avoidance algorithm is mainly the basis for congestion control in the Internet. Since the Internet is connected with various types of network as shown in Figure 1.1, it is very difficult to improve the performance in the HWWN. The performance of congestion avoidance algorithm degrades significantly in HWWN because of the following reasons.

1. HWWN is still expanding with different types of networks such as Wireless Sensor Networks, Asymmetric networks, Ad hoc networks, High-speed networks, Satellite networks etc. The characteristics of one network differ from another with respect to bandwidth, latency, mobility, channel losses etc. Therefore it is very difficult to develop a unified congestion avoidance algorithm which is suitable for all kind of networks.

2. The congestion control mechanism deploys exponential growth of cwnd at SS phase and linear growth of cwnd at CA phase as shown in Figure 1.4. It is to probe the network for determining the available bandwidth and to avoid the congestion with large burst of data. Since the number of
users sharing the same link vary from time to time in the HWNN, it is
difficult to estimate the available bandwidth exactly. It leads to the
congestion in the network.

3. Packet loss occurs when correctly transmitted packets from a sender never
arrive at the intended receiver. Packets are usually lost due to congestion
in the wired network. But in case of wireless network, the packet loss may
be due to poor link quality, high Bit-Error-Rate (BER) etc. The packet
losses affect data transfer throughput and overall end-to-end connection
quality. Consequently it is desirable to have an accurate packet loss
differentiation algorithm to distinguish the congestion loss from non-
congestion loss. But it is very difficult to determine the causes for packet
loss exactly in HWNN.

4. The implicit assumption that the congestion happens in the forward path
(data-flow path), is no longer valid in the asymmetric network because
the asymmetric networks have high bandwidth in the data flow path and
low bandwidth in acknowledgement flow path. Therefore the congestion
is mostly happening in the acknowledgment path resulting in the
acknowledgement packet loss. It causes unnecessary retransmission of
packets even though the data packets are received by the receiver.

5. The delay-based congestion avoidance algorithm takes the variation in
RTT to detect the incipient of congestion in the network in the early
stage. The RTT is composed of deterministic delay and random delay.
The transmission delay, propagation delay and delay caused by router
processing are called deterministic delay. The delay caused by medium
access contention and temporary link failure and queuing delay are
known as random delay. The RTT of a packet is affected by the buffering
time (queuing delay) of packet in the intermediate nodes (routers),
changes in the route and delay due to temporary link failure etc. These
delays are random and difficult to estimate.
1.6 MOTIVATION OF RESEARCH

The causes that motivated this research are briefly discussed in this section. In the last two decades, there has been a fast growth in the Internet which has been evolving from homogeneous network to heterogeneous network. The heterogeneous network has been continuing to draw research because new link technologies have evolved. For example, the past few years, there has been an increase in wireless network with variable bottleneck rates, data center networks with high data rates, cellular wireless networks with highly variable packet delays, links with non-congestive packet losses. The behavior of TCP in this network could be sub-optimal because it is expected to operate in diverse set of networks with different characteristics [39, 40].

Table 1.1 TCP Variants available in major Operating System families

<table>
<thead>
<tr>
<th>Operating System</th>
<th>TCP Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Family</td>
<td>TCP-Reno, CTCP</td>
</tr>
<tr>
<td>Linux Family</td>
<td>TCP-Reno, TCP-Vegas, TCP-Veno, HTCP, BIC-TCP, CUBIC-TCP, HSTCP, Hybla-TCP, Westwood+, Illinous-TCP, LP-TCP, STCP, YEAH-TCP</td>
</tr>
</tbody>
</table>

There are many TCP variants proposed in the literature [11] to cope with the Internet which is connected with diverse set of networks with different characteristics. Table 1.1 shows list of TCP variants presently used in the two major operating systems [41]. Among them, TCP-Vegas provides better performance as compared with TCP-Reno, TCP-Newreno and TCP-SACK in terms of throughput and network utilization in the multihop network, wireless network and heterogeneous network [42-45]. But the congestion avoidance algorithm of TCP-Vegas has attracted significant attention in improving the performance in the different type of networks as discussed in [26][29][46-49].
The IETF RFC 6077 describes the several challenges of congestion avoidance algorithm of TCP [12]. Among them, some of the issues are well known for many years and may require more study. The RTT estimation and packet loss differentiation (Discriminating the congestive packet loss and non-congestive packet loss) are two important issues among them which is playing vital role in congestion avoidance algorithm of TCP-Vegas.

1.6.1 Challenges in RTT Estimation

The RTT of a packet is defined as the delay between the sending of a packet and the reception of corresponding response (ACK). For every reception of the acknowledgement, sender measures the RTT of the packet concerned. The measured RTT is then used to estimate the RTT of packet which is to be sent next. Because of the varying data traffic in the network and different characteristics of communication media like wired, wireless etc, RTT is always uncertain in HWWN. Therefore it is difficult to estimate the RTT of a packet from the sender side at every instant. Since the performance of delay-based congestion control algorithm mainly depends on variability in the RTT, it is necessary to estimate the RTT of packet accurately [39]. The baseRTT and Smooth Round Trip Time (SRTT) are two different RTTs normally used in the congestion avoidance algorithm of TCP variants. These two RTTs are robust but far from being perfect [12][29][50-52].

The baseRTT is the smallest round-trip time measured throughout the life time of TCP connection. It is an estimate of the propagation delay without the queuing delay. When there is a change in route or any environmental changes in the wireless link, the RTT of packets is increased. Since the baseRTT is the minimum RTT of entire history of measured RTT, the increase in RTT due to re-route is not accounted for estimating the baseRTT. Therefore the baseRTT may become an unbefitting history value and hence not suitable for congestion avoidance [12][29]. C.P.Fu et al [29] proposed a method to reset the baseRTT when the route to the destination is changed. Whenever the sender receives triple duplicate acknowledgement (TDA), the baseRTT is reset to the RTT of recently acknowledged packet. Therefore the increase in RTT due to re-route is accounted.
But the baseRTT is affected by some additional variable delays caused by congestion, cross-traffic, recovery procedures, change in route or other sub-layer mechanisms. Due to this variability, one single measured RTT value is insufficient to characterize a path. Therefore delay-based TCP make use of an Exponentially Weighted Moving Average (EWMA) model [20] which is a low-pass filter, to derive the SRTT. The drawback EWMA is that the averaged value reacts more slowly to sudden changes in the measured RTT because it is biased towards long history of previous samples. The EWMA is robust, but it is far from being perfect [50, 51].

1.6.2 Challenges in Differentiating Packet Losses

Next, the differentiating the congestive losses from non-congestive loss is very difficult. Non-Congestive losses are mostly caused by transmission errors and packet corruption while crossing a poor quality wireless link. In practice, the wireless channel condition can dynamically vary due to the mobility of stations or obstacle crossing the wireless path, multipath interference, interference from other devices etc [53, 54]. The TCP assumes the packet losses due to these reasons as a signal of congestion and deploys the congestion avoidance algorithm. It slows down the growth of congestion window which results in poor performance of TCP in the HWWN. There are several techniques used in the delay-based congestion control mechanism to differentiate the congestive loss from non-congestive losses [18][25][29][55]. Since the delay-based congestion control mechanism takes the RTT variability as a signal for detecting the network congestion and presently used RTT estimation is not being perfect [9], it is necessary to have a good loss differentiation algorithm. It motivates to propose a loss differentiation algorithm using a perfect RTT estimation technique.

1.7 PROBLEM STATEMENT

This research focuses on two major problems of heterogeneous wired-wireless networks.
1.7.1 RTT Estimation Problem in Congestion Avoidance Algorithm

The delay-based congestion avoidance algorithms take baseRTT, Average RTT and SRTT to predict the incipient of congestion in the network. The baseRTT is the smallest round-trip time measured throughout the lifetime of TCP connection. SRTT is Smooth Round Trip Time which is the weighted average between the previous estimate of SRTT and the newly measured RTT i.e.

\[
\text{SRTT}(t) \leftarrow \alpha \cdot \text{SRTT}(t-1) + (1 - \alpha) \cdot \text{RTT}(t)
\] (1.1)

where RTT(t) is the measured RTT of recently acknowledged packet at time t and \( \alpha \) is filter gain constant or weight factor which is estimated by making the sum of square of prediction error into minimum by selecting the appropriate value of \( \alpha \). The difference between measured value of RTT and predicted value is called prediction error. The estimated value for \( \alpha \) is 0.875 [20]. SRTT(t-1) is the SRTT estimation at time (t-1) i.e. previous estimate of SRTT. The initial value of SRTT i.e. SRTT(0) is the value of first RTT measured.

If the route to the destination is changed and the new route has a shorter propagation delay, the estimate of baseRTT does not cause any serious problem in the congestion avoidance algorithm. This is because most likely some packets will experience shorter round trip delay and baseRTT will be updated accordingly. On the other hand, if the new route to the destination has a longer propagation delay, the congestion avoidance algorithm is not be able to distinguish whether the increase in the round trip time is due to congestion in the network or a change in the route. Since the TCP assumes that the increase in the round trip time is a sign of congestion in the network, it interprets it as congestion and decreases the congestion window size (cwnd). It results in performance degradation in the throughput of delay-based TCPs [46, 47].

If there is no congestion in the network, then RTT of a packet is equal to baseRTT. As the number of backlogged packet increases, the RTT also increases i.e. the RTT is the sum of baseRTT and delay caused by backlogged packets. The backlogged packets mean the packets which are yet to be acknowledged at sender. The increase in RTT is the signal to the incipient of congestion in the network

\[
\text{RTT} = \text{baseRTT} + \frac{N}{C_{\alpha}}
\] (1.2)
where $N$ is the number of backlogged packets in bytes and $C_a$ is the available bandwidth of bottleneck link in Mbps. $N/C_a$ is the delay caused by the backlogged packets in the network. Suppose the connection fails and router changes the route, then the route to new connection may have longer propagation delay. But Vegas still maintains the baseRTT for estimating the backlogged packets in the network. Therefore using the baseRTT directly for estimating the backlogged packets may be unbefitting.

The RTT of a packet is affected by random delay i.e. delay caused by the backlogged packets in the network, changes in the route, cross-traffic generated by other TCP connection, high BER, link failure and asymmetric characteristics of link. However this random delay is very difficult to estimate. Since the random delay affects the RTT of a packet, the prediction of congestion in early stage may also be disturbed. Even though the SRTT is robust, it is far from being perfect [50-52]. The SRTT and average RTT are used to estimate the number of packets in flight i.e. the number of packets yet to acknowledge. By analyzing the number of packet in flight, the onset of congestion is predicted. Therefore it is necessary to estimate the RTT perfectly.

Since the SRTT filters the high frequency components of RTT and smoothes the RTTs, the sharp and sudden changes in RTT caused by random delay are not able to predict by the SRTT. The average RTT also results in same behavior like SRTT. Therefore both SRTT and Average RTT are smoothing the RTT and allow the low frequency component of RTTs.

1.7.2 Problem in Packet Loss Differentiation

In the wired network, the packet losses are mostly due to the network congestion whereas in the wireless network, packet losses are caused by many reasons other than congestion. The reasons for non-congestive packet losses are due to wireless transmission error, poor link quality, high BER, temporary link failure. In practice, the channel condition or link quality of a wireless network varies dynamically due to the mobility of devices, multipath interference, interference from
other devices, etc. Hence the assumption that packet loss is an indicator of network congestion may not be applicable for HWWN [53,54]. Differentiating the non-congestive loss from congestive loss is very difficult. There are several loss differentiation algorithms in the literature [18][25][29][55,56] which are all using implicit information such as RTT, average RTT, baseRTT and SRTT to differentiate the cause for packet losses. TCP-Vegas takes average RTT and baseRTT to determine three states of network namely congestive state, non-congestive state and the state in between these two states. Based on these states, the cwnd is decreased/increased/not changed respectively. The performance analysis of loss differentiation algorithms given in [18][29][57, 58], shows that TCP-Vegas is not providing better performance in heterogeneous environment because it assumes non-congestive loss as congestive loss and reduces its cwnd consequently. This results in severe throughput deterioration when packets are lost for reasons other than congestion.

Further the implicit information such as average RTT and baseRTT used in TCP-Vegas are not sufficient to distinguish the reason for packet losses in HWWN. Therefore it is necessary to include the explicit information such as channel condition of wireless device etc. in the LDA [8][59].

1.8 RESEARCH OBJECTIVES

The objectives of this research are

➢ To design a RTT estimator using Auto Regressive Integrated Moving Average model that can be used to estimate the sudden increase/decrease in the RTT.
➢ To design packet loss differentiation algorithm using the proposed RTT estimation that can differentiate the congestive loss from non-congestive loss.
➢ To develop a congestion avoidance mechanism using the proposed packet loss differentiation algorithm.
➢ To enhance the congestion avoidance mechanism using the received signal strength of receiver node to differentiate the congestive loss from non-congestive loss.

1.9 OVERVIEW OF PROPOSED SYSTEM

Figure 1.5 shows an overview of the proposed enhancement of congestion avoidance algorithm and its components.

The congestion avoidance algorithm of sender side and receiver side of TCP-Vegas is modified in this research. The sender side algorithm takes the measured RTT for the RTT estimation. It also takes sequence number of a packet just sent ($S_{seq}$) and sequence number of a packet just acknowledged ($A_{seq}$) for backlogged

![Diagram](image-url)
packet estimation. The essential components of proposed system are 1) RTT estimation using ARIMA(2,1,1) model, 2) Backlogged packet estimation, 3) Channel status notification, 4) Loss differentiation algorithm, 5) Congestion window control. These components are explained below.

1. RTT Estimation using ARIMA(2,1,1)

This component collects the RTTs of recently acknowledged packets and estimates the probable time taken by a packet to be acknowledged. The estimation of RTT is done using Autoregressive Integrated Moving Average Model (ARIMA(p,d,q)). This model uses three parameters (p, d and q) as discussed in Section 2.3.4. ‘p’ is the order of Autoregressive part which indicates the number of prior RTTs needed for RTT estimation. ‘q’ is the order of Moving Average part which indicates the number of prediction error terms needed for RTT estimation. The prediction error is the difference between the estimated RTT and measured RTT. ‘d’ is the order of successive difference of RTT which is discussed in detail in Section 2.3.4. In this research, we propose the orders (p=2, d=1 and q=1) for RTT estimation for heterogeneous wired/wireless network because of the lag of autocorrelation and partial correlation of RTT. The detailed description of the proposed order is discussed in Section 3.3.1.2 and 3.3.1.3. The estimated RTT is then used to determine the backlogged packets in the network.

2. Backlogged Packet Estimation

An increase of the load by a sender would increase the round-trip time because the packets are queued in the intermediate nodes (routers). It shows that there is some correlation between the load variation and RTT variation. The increase/decrease in load is done by varying the cwnd. The variation in RTT is caused by the number of packets in flight which is known as backlogged packets. Therefore this component estimates the backlogged packets in the network using the ARIMA(2,1,1) model-based RTT estimation.

3. Channel Status Notification

The errors on the wireless channel are bursty and wireless channel is distinct and time varying for each wireless node. As the mobile nodes move, the received signal strength (RSS) varies significantly. In addition, there are some
serious effects on wireless channel due to fading, interference from other wireless node and shadowing from objects. All of which degrade the performance of reliable data service in the wireless network. Hence the wireless channel condition is determined by the receiver and informed to the sender. This component informs the wireless channel state to the sender to improve the performance of loss differentiation thereby the congestion avoidance. To do so, the receiver measures the RSS and compare with the threshold value (RXThresh). If the RSS is greater, then it assumes that the wireless channel is good, otherwise bad. This information is informed to sender by piggybacking a single bit in the ACK packet. If it set to 1, then channel is good, otherwise bad.

4. Loss Differentiation Algorithm

Since the HWWN composed of wired network and wireless network, it is necessary to distinguish the congestive losses from non-congestive losses. Hence this component takes the estimation of backlogged packets and channel status notification to differentiate the cause of packet losses. The estimation of backlogged packets uses the implicit information like estimated RTT using ARIMA(2,1,1) model and baseRTT. The received signal strength is the explicit information that is used to find the condition of the channel. Since this component make use of implicit and explicit information, the loss differentiation is better than other algorithms.

5. Congestion Window Control

Based on the type of packet loss (congestive loss or non-congestive loss), the load to the network is varied. The variation in load to the network is done by adjusting the congestion window size. TCP-Vegas adjust the congestion window size based on the state of network which discussed in the section 1.7.2. The state in between the congestive state and non-congestive state is very critical which is tuned with the help of explicit information because the implicit information is not sufficient to handle these states. Therefore this component decides the increase/decrease of the congestion window size based on the loss differentiation algorithm which takes both
implicit and explicit information to differentiate the non-congestive loss from congestive loss.

1.10 THESIS OUTLINE

The rest of the thesis is organized as follows.

A literature survey of the round trip time estimation, loss differentiation algorithm (LDA), congestion avoidance algorithm and cross layer approaches have been discussed in the Chapter 2. The Chapter 3 presents the proposed ARIMA(2,1,1) model-based RTT estimation. In Chapter 4, we propose a Loss Differentiation Algorithm (LDA). This algorithm makes use of the proposed ARIMA model based RTT estimation for discriminating the congestive packet loss from non-congestive packet loss. The performance of proposed LDA is compared with other LDAs. The Chapter 5 presents the congestion avoidance algorithm which uses the proposed LDA and its performance analysis in wired-wireless network and bandwidth-asymmetry network. The impact of high BER, long latency network and varying packet size are also analyzed in this chapter. Chapter 6 discusses proposed enhancement of congestion avoidance algorithm. It takes the received signal strength obtained from the wireless receiver to differentiate the non-congestive loss from congestive loss. The performance of the proposed enhancement of congestion avoidance algorithm is analyzed with varying network conditions. The Chapter 7 presents concluding remarks and directions to future research.

1.11 CONCLUSION

This chapter provides the overview and significance of heterogeneous wired-wireless networks and explains various types of congestion control mechanism in the literature. It also describes the principle of operation of congestion control mechanism and its issues in heterogeneous wired-wireless network. Following to this, the motivation of research, problem statement of research and the objectives of this research are discussed. Finally the system overview of our research approach and outline of thesis are provided.