Congestion Controlled Adaptive Routing in Wireless Networks

[1]Ajitha Rani, [2]S. Jerine
[1]M.Phil Scholar, [2]Associate Professor
[1,2]Noorul Islam Centre for Higher Education.

Abstract:

The wide deployment of Wireless network has increased the network traffic. Long congestion and frequent link failures in wireless network lead to more number of packets being dropped and incur high end-to-end delay, thereby degrading the overall performance of the network. For solving the issue of congestion in Wireless network, congestion controlled the adaptive routing mechanism is proposed. The proposed work has three phases namely congestion detection, alternative path computation and Fast retransmit and fast recovery the packets on a new path. Congestion control is mainly incorporated at the transport layer. Congestion in the path can be detected based on the free space available in the buffer. To identify multipath routing and load aware of each node to minimize the data drop but all given approach are using routing base congestion control. Different approaches to congestion avoidance have been introduced in the past few years. The traditional flooding and gossiping algorithm can cause loss of data. Random Early Detection (RED) is an alternative way to reduce packet loss as congestion measure. Instead of dropping only at a full buffer, RED maintains an exponentially weighted-queue length and drops packets with a probability. If the weighted queue length is less than a minimum threshold no arrival packets are dropped. When it exceeds a maximum threshold all packets are dropped. When it is in between minimum and maximum threshold a packet is dropped with a probability that is piecewise linear and increasing function of the CR (Congestion resistant) algorithm. This algorithm will find another way to send the packets and to maintain that, this way is shortest safe way and controls congestion.

Keywords-wireless network, Congestion control, Alternative path computation.

1. INTRODUCTION

The Congestion control mechanism has been responsible for maintaining stability as the internet scales up by many orders of magnitude in size, speed, traffic volume, coverage and complexity over the last three decades. The primary goal is to develop a coherent theory of internet congestion control from the ground up to help understand and design the equilibrium and stability properties of large – scale under end - to - end control. The internet is an interconnected set of computing, storage or communication resources shared by computing users for the purposes, A users is typically not a human, but a traffic flow from a source to a destination through a subset of these resources. A computing or communication resource to a characterized by how fast it can process or transmit information, in units of bits per second or packets per second. A storage resource queues up packets while they wait to be

processed or transmitted. Each resource abstractly as a "link" that consist of a single server with a buffer (waiting space); often assume that the buffer capacity is infinite.

2. RELAED WORKS

Traffic Allocating Mechanism:

The focal point of is congestion avoidance, detection and alleviation mechanisms. The mechanisms don't depend on any particular multipath routing algorithm. So the related work of multi-path routing is described for improving the reliability of forwarding and reducing the cost of route maintenance, investigators have proposed many multi-path forward schemes. Proposed two mechanisms of no intersection paths and twist paths, which aim is to recover from fault via maintaining more paths and to improve the reliability of routing.

Allocating resources:

The network protocol hierarchy to understand how data can be transferred among processes across heterogeneous networks. We now turn to a problem that spans the entire protocol stack - how to effectively and fairly allocate resources among a collection of competing users. The resources being shared include the bandwidth of the links and the buffers on the routers or switches where packets are queued awaiting transmission. Packets contend at a router for the use of a link, with each contending packet placed in a queue waiting its turn to be transmitted over the link. When too many packets are contending for the same link, the queue fills and two undesirable things happen: packets experience increased end-to-end delay, and in the worst case, the queue overflows and packets have to be dropped. When long queues persist and drops become common, the network is said to be congested. Most networks provide a congestion-control mechanism to deal with just such a situation.

Congestion control and resource allocation are two sides of the same coin. On the one hand, if the network takes an active role in allocating resources—for example, scheduling which virtual circuit gets to use a given physical link during a certain period of time-then congestion may be avoided, thereby making congestion control unnecessary. Allocating network resources with any precision is difficult, however, because the resources in question are distributed throughout the network; multiple links connecting a series of routers need to be scheduled. On the other hand, you can always let packet sources send as much data as they want and then recover from congestion should it occur. This is the easier approach, but it can be disruptive because many packets may be discarded by the network before congestion can be controlled. Furthermore, it is precisely at those times when the network is congested-that is, resources have become scarce relative to demand-that the need for resource allocation among competing users is most keenly felt. There are also solutions in the middle, whereby inexact allocation decisions are made, but congestion can still occur and hence some mechanism is still needed to recover from it. Whether you call such a mixed solution congestion control or resource allocation does not really matter. In some sense, it is both.

Congestion control and resource allocation involve both hosts and network elements such as routers. In network elements, various queuing disciplines can be used to control the order in which packets get transmitted and which packets get dropped. The queuing discipline can also segregate traffic to keep one user's packets from unduly affecting another user's packets. At the end hosts, the congestion-control mechanism paces how fast sources are allowed to send packets. This is done in an effort to keep congestion from occurring in the

first place and, should it occur, to help eliminate the congestion.

An overview of congestion control and resource allocation. Then discuss different queuing disciplines that can be implemented on the routers inside the network, followed by a description of the congestion-control algorithm provided by TCP on the hosts. The fourth section explores various techniques involving both routers and hosts that aim to avoid congestion before it becomes a problem. Finally, to examine the broad area of quality of service.

3. METHODOLOGY

Route/Path Discovery

Route discovery is initiated only when source wants to send data to the destination. In single route discovery multiple node disjoint path from source to destination are discovered and stored in a routing table. Only if all these paths fail, routing table. Only if all this path fail, route discovery phase is re-initiated. Random early detection (RED), this scheme in that each router is programmed to monitor its own queue length and, when it detects that congestion is imminent, to notify the source to adjust its congestion window. The first is that rather than explicitly sending a congestion notification message to the source, RED is most commonly implemented such that it implicitly notifies the source of congestion by dropping one of its packets. The source is effectively notified by the subsequent timeout or duplicate ACK. RED is designed to be used in congestion with TCP, which currently detects congestion by means of timeouts (or some other means of detecting packet loss such as duplicate ACKs). As the early part of the RED acronym suggests, the gateway drops the packet earlier than it would have to, so as to notify the source that it should decrease its congestion window sooner than it would normally have. In other words, the router drops a few packets before it has exhausted its buffer space completely, so as to cause the source to slow down, with the hope that this will mean it does not have to drop lots of packets later on.

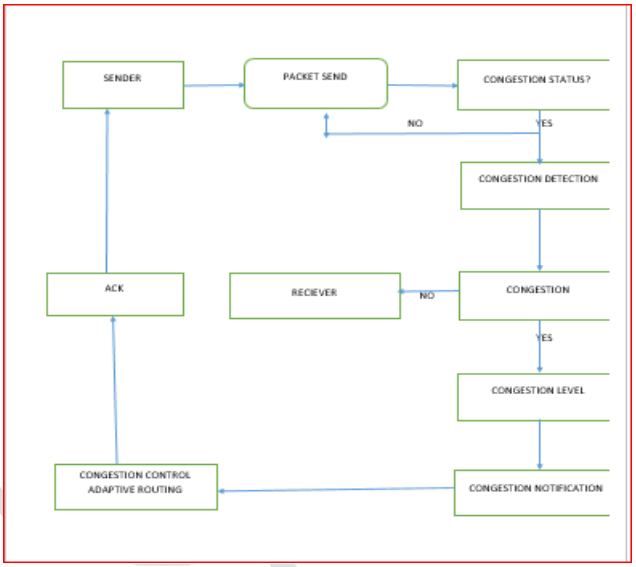
TCP Congestion Control Mechanism:

The end – to – end congestion control is implemented by TCP. The essential strategy of TCP is to send packets into the network without a reservation and then to react to observable events that occur. The internet was suffering from congestion collapse – hosts would send their packets into the internet as fast as the advertised window would allow, congestion would occur at some router (causing packets to be dropped), and the hosts would time out retransmit their packets, resulting in even more congestion. The congestion control is for each source to determine how much capacity is available in the network, so that it knows how many packets it can safely have in transit. Once a given source has this many packets in transit, it uses the arrival of an ACK as a signal that one of its packets has the network without adding to the level of congestion. By using ACKs to pace the transmission of packets, TCP is to be self –clocking. Because other connections come and go, the available bandwidth changes over time, meaning that any given source must be able to adjust the number of packets it has in transit.

Fast Retransmit and Fast Recovery:

The mechanisms described so far were part of the original proposal to add congestion control to TCP. It was soon discovered, that the coarse-grained implementation of TCP timeouts led to long periods of time during which the connection went dead while waiting for a timer to expire. Because a new mechanism called fast retransmit triggers the retransmission of a dropped packet sooner than the regular timeout mechanism. The fast retransmit mechanism does not replace regular timeouts, it just enhances that facility. The idea of fast retransmit is straightforward. Every time a data packet arrives at the receiving side, the receiver responds with an acknowledgment, even if this sequence number has already been acknowledged the data the packet contains because earlier data has not yet arrived – TCP resends the same acknowledgment it sent the last time. This second transmission of the same acknowledgment is called a duplicate ACK. When the sending side sees a duplicate ACK, it knows that the other side must have received a packet out of order, which suggest that an earlier packet might have been lost. Since it is also possible that the earlier packet has only been delayed rather than lost, the sender waits unit it sees some number of duplicate ACKs and then retransmits the missing packet. The TCP waits unit it has seen three duplicate ACKs before retransmitting the packet.

4. System Architecture:



Congestion Control

Now turning our attention to the host half of the mechanism, the source records how many of its packets resulted in some router setting the congestion bit. In particular, the source maintains a congestion window, just as in TCP, and watches to see what fraction of the last window's worth of packets resulted in the bit being set. If less than 50% of the packets had the bit set, then the source increases its congestion window by one packet. If 50% or more of the last window's worth of packets had the congestion bit set, then the source decreases its congestion window to 0.875 times the previous value. The value 50% was chosen as the threshold based on analysis that showed it to correspond to the peak of the power curve. The "increase by 1, decrease by 0.875" rule was selected because additive increase/multiplicative decrease makes the mechanism stable.

Congestion Avoidance:

In the Congestion Avoidance algorithm a retransmission timer expiring or the reception of duplicate ACKs can implicitly signal the sender that a network congestion situation is going on. The sender

immediately sets its transmission window to one half of the current window size, but to one segments. If Congestion is indicated by a timeout, the congestion window is reset to one segment, which automatically puts the sender into slow start mode. If congestion was indicated by duplicate ACKs, the fast retransmit and fast recovery algorithms are invoked.

When data is received during Congestion Avoidance, then the congestion window is increased. Slow Start is only used up to the halfway point where Congestion originally occurred. This halfway point was recorded earlier as the new transmission window. After this halfway point, the Congestion window is increased by one segment for all segment in the transmission window that are acknowledged. This mechanism will force the sender

Conclusion

An adaptive routing scheme for congestion control in Wireless Networks. The scheme finds the congestion free routing path from source node to target node, based on the congestion time. As the time of service plays a very important role in Wireless Network, the use is made of this to calculate the congestion time. The performance parameters analysed are route finding, congestion threshold and congestion ratio.