ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

The Survey of Congestion Control Protocols for Smart Transport System

Koushik reddy ramasani, Dr.syed umar, seelam eswar reddy

B.Tech Student, Dept. of ECM, K L University, vaddeswaram, Guntur, AP, India.

Assoc.Professor, Dept. of ECM K L University, Vaddeswaram, Guntur, AP, India.

B.Tech Student, Dept. of ECM K L University, vaddeswaram, Guntur, AP, India.

ABSTRACT: Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues). Think about freeway congestion. Many vehicles enter the freeway without regard for impending or existing congestion. As more vehicles enter the freeway, congestion gets worse. Eventually, the on-ramps may back up, preventing vehicles from getting on at all. Congestion control in computer network is an important issue to be addressed. Various congestion protocols are used for avoiding congestion in network. The survey of congestion control protocols in network is important and necessary for smart transport system. This paper discusses the advantages/disadvantages and the applications of various con-gestion control protocols for wired/wireless networks. It ex-plores the motivation behind the designed, and traces the evolution of these congestion control protocols.

KEYWORDS: ECN, VCP, XCP, DPCP, TFRC, LF

1. INTRODUCTION

Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues). Think about freeway congestion. Many vehicles enter the freeway without regard for impending or existing congestion. As more vehicles enter the freeway, congestion gets worse. Eventually, the on-ramps may back up, preventing vehicles from getting on at all. In packet-switched networks, packets move in and out of the buffers and queues of switching devices as they traverse the network. In fact, a packet-switched network is often referred to as a "network of queues." A characteristic of packet-switched networks is that packets may arrive in bursts from one or more sources. Buffers help routers absorb bursts until they can catch up. If traffic is excessive, buffers fill up and new incoming packets are dropped. Increasing the size of the buffers is not a solution, because excessive buffer size can lead to excessive delay.

Congestion typically occurs where multiple links feed into a single link, such as where internal LANs are connected to WAN links. Congestion also occurs at routers in core networks where nodes are subjected to more traffic than they are designed to handle. TCP/IP networks such as the Internet are especially susceptible to congestion because of their basic connection- less nature.

There are no virtual circuits with guaranteed bandwidth. Packets are injected by any host at any time, and those packets are variable in size, which make predicting traffic patterns and providing guaranteed service impossible. While connectionless networks have advantages, quality of service is not one of them. Shared LANs such as Ethernet have their own congestion control mechanisms in the form of access controls that prevent multiple nodes from transmitting at the same time. See "Access Methods" and "MAC (Media Access Control)."The following basic techniques may be used to manage congestion.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

End-system flow control this is not a congestion control scheme, but a way to prevent the sender from overrunning the buffers of the receiver. See "Flow-Control Mechanisms."

Network congestion control In this scheme, end systems throttle back in order to avoid congesting the network. The mechanism is similar to end-to-end flow controls, but the intention is to reduce congestion in the network, not the receiver.

Network-based congestion avoidance In this scheme, a router detects that congestion *may* occur and attempts to slow down senders before queues become full.

Resource allocation This technique involves scheduling the use of physical circuits or other resources, perhaps for a specific time period. A virtual circuit, built across a series switches with a guaranteed bandwidth is a form of resource allocation. This technique is difficult, but can eliminate network congestion by blocking traffic that is in excess of the network capacity. A list of related topics may be found on the related entries page. Caching is probably the ultimate congestion control scheme. By moving content closer to users, a majority of traffic is obtained locally rather than being obtained from distant servers along routed paths that may experience congestion. Caching has become a serious business on the Internet, as discussed under "Content Distribution."

Queuing and Congestion

Any discussion of congestion naturally involves queuing. Buffers on network devices are managed with various queuing techniques. Properly managed queues can minimize dropped packets and network congestion, as well as improve network performance.

The most basic technique is FIFO (first-in, first-out), where packets are processed in the order in which they arrive in the queue. Going beyond this, a priority queuing scheme uses multiple queues with different priority levels so that the most important packets are sent first. An important queuing technique is to assign flows to their own queues. This differentiates flows so that priorities can be assigned. Just as important, each flow is responsible for making sure that it does not overflow its own queue. Separating queues in this way ensures that each queue only contains packets from a single source.

Congestion Control in Frame Relay

While this topic is primarily about congestion problems in connectionless packet-switched networks, it is useful to examine the way congestion is handled in a connection-oriented network. Frame relay provides a good example. Frame relay subscribers negotiate a CIR (committed information rate) with the service provider. The CIR is the guaranteed level of service, but providers usually allow subscribers to burst over this level if network capacity is available. However, frames in excess of the CIR are marked as *discard eligible*. If a switch on the network becomes congested, it will drop discard eligible frames. This ensures that the service providers can meet their negotiated CIR levels for subscribers.

Dropping frames is never a good idea, so two congestion avoidance mechanisms are available:

BECN (backward explicit congestion notification) When a switch starts to experience congestion (i.e., the buffers/queues are getting full), it can send a frame in the backward direction to senders with the BECN bit set to inform senders to slow down.

FECN (forward explicit congestion notification) When a switch starts congesting, it can send a frame in the forward direction to receiving nodes with the FECN bit set. This informs the forward nodes that they should inform the sender to slow down. Note that sender or receiver do not need to respond to BECN or FECN, but eventually, network switches will drop frames as congestion continues to increase.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

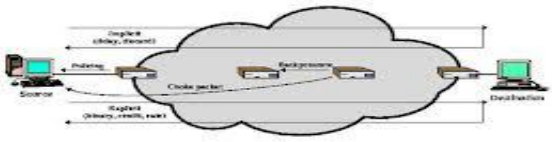
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

Congestion Control and Avoidance in TCP

Until the mid 1980s, the Internet was prone to a phenomenon called "congestion collapse." This would occur because there was little control over managing heavy network loads. Individual connections used flow controls between sender and receiver to prevent the sender from overwhelming the receiver. These are described in (Transmission Control Protocol, September 1981).But these early flow controls were designed to prevent overflowing the receiver's buffers, not the buffers of network nodes. However, the early Internet consisted of a large number of relatively slow links, so congestion was not the problem it is today. In the late 1980s, Van Jacobson developed the congestion control mechanisms that make TCP respond to congestion in the network. The basic "signal" is a dropped packet, which causes the host to stop or slow down.



Normally, when a host receives a packet (or set of packets), it sends an ACK (acknowledgement) to the sender. A window mechanism allows the host to send multiple packets with a single ACK as discussed under "Flow-Control Mechanisms." Failure to receive an ACK indicates that the receiving host may be overflowing or that the network is congested. In either case, the sender slows down or stops.

A strategy called *additive increase/multiplicative decrease* regulates the number of packets that are sent at one time. If you graphed the flow, you would see a saw tooth pattern where the number of packets increases (additive increase) until congestion occurs and then drops off when packets start to drop (multiplicative decrease). The window size is typically halved when a congestion signal occurs.

What the host is doing is finding the optimal transmission rate by constantly testing the network with a higher rate. Sometimes, the higher rate is allowed, but if the network is busy, packets start to drop and the host scales back. This scheme sees the network as a "black box" that drops packets when it is congested. Therefore, congestion controls are run by the end systems that see dropped packets as the only indication of network congestion. A sender that is transferring a large file will push for a higher rate until eventually it grabs all the bandwidth. Other hosts may have trouble getting packets through. Often, the host that has grabbed the bandwidth is transmitting the least important traffic. The effect is especially disruptive to real-time traffic such as voice. Even if a link has sufficient bandwidth to handle its load, ill-behaved hosts can saturate the link (although briefly) enough to disrupt voice traffic in a way that is perceptible to users.

Of course, the network can take an active role in managing congestion. That is where "active queue management" and congestion avoidance come into play, as discussed later. RFC 1254 (Gateway Congestion Control Survey, August 1991) describes congestion control and avoidance mechanisms that were reviewed by the IETF Performance and Congestion Control Working Group. The group divided congestion handling into the following:

Congestion recovery Restore the operating state of the network when demand exceeds capacity. **Congestion avoidance** Anticipate congestion and avoid it so that congestion never occurs.RFC 1254 states that the Internet would cease to operate without congestion recovery, but has operated a long time without congestion avoidance. Today, congestion avoidance is an important tool for improving the performance and QoS of the Internet.

RFC 2309 (Recommendations on Queue Management and Congestion Avoidance in the Internet, April 1998) states that router-based mechanisms for controlling congestion can be divided into "queue management" algorithms and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

"scheduling" algorithms. Refer to this RFC for useful information about congestion controls, congestion avoidance schemes, and queue scheduling techniques.

An important goal is to minimize the number of dropped packets. If a host is transmitting at a high rate and the network goes into congestion, a large number of packets will be lost. Congestion avoidance attempts to prevent this without putting limits on network throughput. RFC 2309 points out that it is better to accept the fact that there will be bursts that overflow queues rather than try to maintain queues in a non-full state. That would essentially translate to favoring low end-to-end delay over high throughput. The RFC also notes the following:

The point of buffering in the network is to absorb data bursts and to transmit them during the (hopefully) ensuing bursts of silence. This is essential to permit the transmission of burst data. It should be clear why we would like to have normally-small queues in routers: we want to have queue capacity to absorb the bursts. The counter-intuitive result is that maintaining normally-small queues can result in higher throughput as well as lower end-to-end delay. In short, queue limits should not reflect the steady state queues we want maintained in the network; instead, they should reflect the size of bursts we need to absorb. Keep in mind that bursts can disrupts multiple hosts. If single hosts fill a queue being used by multiple hosts, all of the hosts will need to back off. This results in a period in which the network is underutilized because hosts are sending packets at a lower rate. But eventually, they start building back up with a need to retransmit dropped packets. What happens then is that all the hosts that previously backed off try to resend at about the same time, causing another congestion state. This is called the "global synchronization" problem.

Keep in mind is that TCP handles congestion control. UDP is typically used for real-time audio and video streams because there is no need to recover lost packets. UDP is an unreliable transport protocol that does not send ACK signals back to the source. Since there is no ACK, UDP streams cannot be controlled with traditional TCP congestion controls. Lawrence G. Roberts, one of the early architects of the Internet, made some interesting comments about TCP and its congestion control schemes in a 1997 paper called "Explicit Rate Flow Control, A 100 Fold Improvement over TCP." His comments are paraphrased below. So long as TCP operates only in the end-stations its operation cannot be substantially improved . . . If TCP is not replaced, TCP will cause major overloads and outages on long haul networks like the Internet. Users are severely impacted by the slow start-up rate and high delay variance inherent with TCP. The IETF has not even considered revising TCP. In fact no study has been done by the IETF on flow control because everyone seems to believe, "if it worked in the past, it will continue to work". TCP must be replaced with a new flow control as good as explicit rate flow control as soon as possible.

RFC 2581 (TCP Congestion Control, April 1999) defines TCP's four intertwined congestion control algorithms: slow start, congestion avoidance, fast retransmit, and fast recovery. Each of these is discussed in the following sections.

Slow Start Congestion Control

Slow start reduces the burst affect when a host first transmits. It requires a host to start its transmissions slowly and then build up to the point where congestion starts to occur. The host does not initially know how many packets it can send, so it uses slow start as a way to gauge the network's capacity. A host starts a transmission by sending two packets to the receiver. When the receiver receives the segments, it returns ACKs (acknowledgements) as confirmation. The sender increments its window by two and sends four packets. This buildup continues with the sender doubling the number of packets it sends until an ACK is not received, indicating that the flow has reached the network's ability to handle traffic or the receiver’s ability to handle incoming traffic.

Slow start does not prevent congestion; it simply prevents a host from causing an immediate congestion state. If the host is sending a large file, it will eventually reach a state where it overloads the network and packets begin to drop. Slow start is critical in avoiding the congestion collapse problem. But new applications such as voice over IP cannot tolerate the delay caused by slow start and in some cases, slow start is disabled so the user can "grab" bandwidth. That trend will only lead to problems.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

Fast Retransmit and Fast Recovery (Reno)

Fast retransmit and fast recovery are algorithms that are designed to minimize the effect that dropping packets has on network throughput. The *fast retransmit* mechanism infers information from another TCP mechanism that a receiver uses to signal to the sender that it has received packets out of sequence. The technique is to send several duplicate ACKs to the sender.

Fast retransmit takes advantage of this feature by assuming that duplicate ACKs indicate dropped packets. Instead of waiting for an ACK until the timer expires, the source resends packets if three such duplicate ACKs are received. This occurs before the timeout period and thus improves network throughput. For example, if a host receives packet 5 and 7, but not 6, it will send a duplicate ACK for packet 5 when it receives packet 7 (but not packet 6).

Fast recovery is a mechanism that replaces slow start when fast retransmit is used. Note that while duplicate ACKs indicate that a segment has been lost, it also indicates that packets are still flowing since the source received a packet with a sequence number higher than the missing packet. In this case, the assumption is that a single packet has been dropped and that the network is not fully congested. Therefore, the sender does not need to drop fully back to slow start mode but to half the previous rate.

Note that the preceding mechanisms are called Reno. RFC 2582 (The New Reno Modification to TCP's Fast Recovery Algorithm, April 1999) describes a modification to Reno to cover situations in which ACKs do not cover all of the outstanding data when loss was detected.

Active Queue Management and Congestion Avoidance

Dropping packets is inefficient. If a host is bursting and congestion occurs, a lot of packets will be lost. Therefore, it is useful to detect impending congestion conditions and actively manage congestion before it gets out of hand.Active queue management is a technique in which routers actively drop packets from queues as a signal to senders that they should slow down.RFC 2309 lists the following advantages of active queue management:

Burst is inevitable. Keeping queue size small and actively managing queues improve a router's ability to absorb bursts without dropping excessive packets. If a source overflows a shared queue, all the devices sharing that queue will slow down (the "global synchronization" problem).Recovering from many dropped packets is more difficult than recovering from a single dropped packet.

Large queue can translate into delay. Active queue management allows queues to be smaller, which improves throughput.

Lock-out occurs when a host fills a queue and prevents other hosts from using the queue. Active queue management can prevent this condition. Several congestion avoidance schemes are described next. Keep in mind that the next step beyond these techniques involves traffic shaping, resource reservations, virtual circuits, and QoS techniques. More on this later.

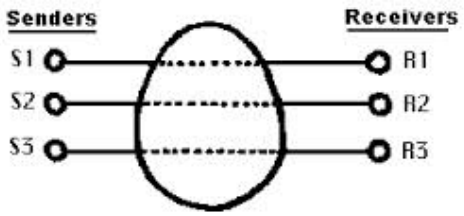


Fig 1. Shared subsystem.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

RED (Random Early Discard)

RED is an active queue management scheme that provides a mechanism for congestion avoidance. RFC 2309 , section 3 provides a description of RED. Sally Floyd and Van Jacobson wrote the basic paper describing RED gateways. That paper along with other papers and related resources may be found at Sally Floyd's Web site. The address is listed on the related entries page.

Unlike traditional congestion control schemes that drop packets at the end of full queues, RED uses statistical methods to drop packets in a "probabilistic" way before queues overflow. Dropping packets in this way slows a source down enough to keep the queue steady and reduces the number of packets that would be lost when a queue overflows and a host is transmitting at a high rate.

RED makes two important decisions. It decides when to drop packets and what packets to drop. RED keeps track of an average queue size and drops packets when the average queue size grows beyond a defined threshold. The average size is recalculated every time a new packet arrives at the queue. RED makes packet-drop decisions based on two parameters:

Minimum threshold (minth) specifies the average queue size *below which* no packets will be dropped.

Maximum threshold (maxth) specifies the average queue size *above which* all packets will be dropped

RED uses time-averaging, meaning that if the queue has recently been mostly empty, RED will not react to a sudden burst as if it were a major congestion event. However, if the queues remain near full, RED will assume congestion and start dropping packets at a higher rate.

RFC 2309 mentions that active queue management mechanisms in the Internet can have substantial performance benefits and that there appears to be no disadvantages to using the RED algorithm. RED (weighted RED) is a technique of dropping packets based on the type of traffic, where it is going, or other factors. WRED may also drop packets based on marking made to packets outside the network.

ECN (Explicit Congestion Notification)

The problem with RED is that it drops packets. A more efficient technique would be for a router to set a congestion notification bit in a packet, and then send the packet to the receiver. The receiver could then inform the sender to slow down via a message in the ACK. All the while, the receiver gets its packet and we avoid using packet drops to signal congestion.

ECN is an end-to-end congestion avoidance mechanism that adopts this technique. As the name implies, ECN provides direct notification of congestion rather than indirectly signaling congestion via dropped packets. ECN works when congestion is moderate. When congestion gets excessive, packet-drop techniques are used.

ECN-enabled routers set a CD (congestion experienced) bit in the packet header of packets from ECN-capable hosts when the length of a queue exceeds a certain threshold value. The packets are forwarded to the receiver, which then sends an ACK to the sender that contains the congestion indicator. This ACK is called an *ECN-Echo*. When the sender receives this explicit signal, it halves the rate at which it sends packets.

Note that ECN requires modifications to TCP. ECN is described in RFC 2481(A Proposal to Add Explicit Congestion Notification to IP, January 1999). Refer to RFC 2884 (Performance Evaluation of Explicit Congestion Notification in IP Networks, July 2000) for further information. The IETF Endpoint Congestion Management (ECM) Working Group has additional information (Web site listed on the related entries page).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

TCP Rate Control

TCP rate control is a technique in which endpoints can adjust their transmissions based on feedback from network devices that perform rate control. Packeteer is an advocate of rate control and this section describes how the company implements it in its Packet Shaper products. Packeteer's Web site has numerous papers on rate control and other congestion control topics.

TCP Rate Control is also known as ERC (explicit rate control). A form of ERC is implemented in ATM networks. The Lawrence G. Roberts paper mentioned earlier in this section describes ERC in both ATM and TCP networks.

Packeteer's Packet Shaper maintains state information about individual TCP connections. This allows it to send feedback to the source that controls its behavior. The primary goal is to control bursts by smoothing out a source's rate of transmission. With bursts reduced, traffic management becomes easier.

The rate control process is performed within the network between the end systems. Packet Shaper intercepts ACKs from receivers and holds them for an amount of time that is precisely calculated to make the sender transmit its next packet in a way that smoothes out the burst.

For example, a source sends a packet to the receiver. The receiver returns an ACK to the sender. Packet Shaper intercepts the ACK and changes some internal settings such as the TCP window size. At the precise moment, Packet Shaper sends the packet and the contents of the packet (ACK sequence number plus the window size) and tells the sender that it is time to transmit another packet.

This results in a steady flow of packets from sources and improved resource management. The downside is that routers must actively track each flow. In addition, changing the contents of in-transit packets may not be a good idea, depending on the network. Traffic management and QoS devices implement TCP rate control.

II. CONGESTION CONTROL PROTOCOLS

There are many ways to classify congestion control algorithms:

i. By the type and amount of feedback received from the network: Loss; delay; single-bit or multi-bit explicit signals
ii. By incremental deploy ability on the current Internet: Only sender needs modification; sender and receiver need modification; only router needs modification; sender, receiver and routers need modification.
iii. By the aspect of performance it aims to improve: high bandwidth-delay product networks; lossy links; fair-ness; advantage to short flows; variable-rate links
iv. By the fairness criterion it uses: max-min, proportion-al, "minimum potential delay"

2.1 Protocols based of Congestion Avoidance

a. TCP-Tahoe[23]

Tahoe uses 'Additive Increase Multiplicative De-crease' (AIMD) for congestion avoidance. In this case a packet loss is taken as a sign of congestion and TCP-Tahoe saves the half of the current window as a threshold value. It then set cwnd to one and starts slow start until it reaches the threshold value. After that it increments linearly until it encounters a packet loss. Thus it increase it window slowly as it approaches the bandwidth capacity. The problem with TCP-Tahoe is that to detect a packet loss it takes a complete timeout interval and in fact, in most implementations it takes even longer because of the coarse grain timeout. In some it sends cumulative acknowledgements in place of immediate ACK's, therefore it follows a 'go back n 'approach. Thus every time when a packet is lost it waits for a timeout and the pipeline is emptied. This offers a major cost in high band-width delay product links.

b. TCP-RENO [24]

TCP RENO adds some intelligence over Tahoe so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost. Reno suggest an algorithm called 'Fast Re-Transmit' in which when 3 duplicate ACK's are received, it is taken as a sign that the segment was lost, so retransmission of the segment without waiting for timeout is done. Reno performs very well over TCP when the packet losses are small. When there are multiple packet losses in one window then RENO doesn't perform too well and its performance is almost the same as Tahoe under conditions of

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

high packet loss.

c. NEW-RENO[25]

New-RENO is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient than TCP RENO in the event of multiple packet losses. Like TCP-RENO, New-RENO also enters into fast-retransmit when it receives multiple duplicate packets. New-RENO differs from TCP-RENO in that it doesn't exit fast recovery until all the data which was out standing at the time it entered fast-recovery is acknowledged.

d. TCP-SACK [26]

TCP with 'Selective Acknowledgments' is an extension of New-RENO. TCP-SACK works around the problems face by TCP RENO and TCP New-Reno, namely detection of multiple lost packets, and re-transmission of more than one lost packet per RTT. SACK retains the slow-start and fast-retransmit parts of RENO. The biggest problem with SACK is that currently selective acknowledgements are not provided by the receiver.

e. TCP-VEGAS[30]

Vegas is a TCP implementation which is a modification of Reno. It based on the fact that proactive measure to en-counter congestion is much more efficient than reactive ones. TCP-VEGAS overcome the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network. The main

Notification to the end-hosts. While doing this RED allows them to reduce their transmission rates before queues in the network overflow and packets are dropped. To detect Congestion RED maintains an exponentially weighted moving average of the queue length. One of the fundamental problems with RED is that they rely on queue length therefore it has an inherent problem in determining the severity of con-gestion. The AVQ algorithm maintains a virtual queue whose virtual capacity is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect a new arrival. Packets in the real queue are marked/dropped when the virtual buffer overflows. The virtual capacity at each link is then modified such that total flow entering each link achieves a desired utilization of the link. An important feature of the AVQ scheme is that in the absence of feedback delays the system is maximizes the sum of utilities of all the users in the network.

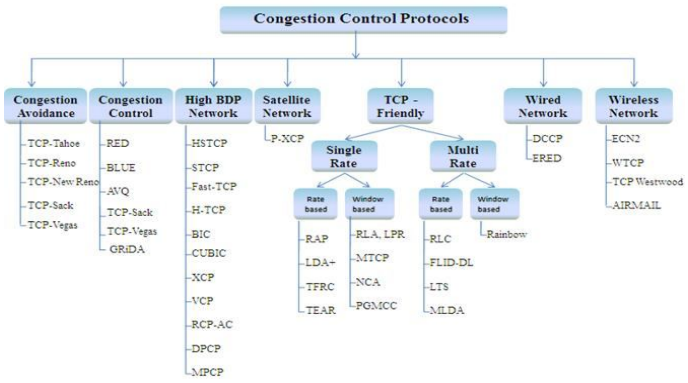


Figure 1: Types of Congestion-Protocols

d. GRiDA

GRiDA is a distributed online algorithm to reduce power consumption in backbone networks. GRiDA is used to selectively switch off links in an Internet Service Provider IP-based network to reduce the system energy consumption. Differently from other approaches, GRiDA does neither require a centralized controller node, nor the knowledge of the current traffic matrix. GRiDA saves energy up to 50% as compared to other existing centralized algorithms.

e. PID

In this case PID design is used with linear gain scheduling and normalized values that works very well under different network traffic conditions. The controller is tuned for the worst scenario and works properly in a wide range of situations. Thus our PID controller is determined only by one parameter, other than traditional PID controller is by three or more. The robust PID congestion controller can outper-form the existing controller, such as PI, RED, on

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

keeping the router queue size at the target value.

2.3 High Bandwidth Delay Product (BDP) Network

a. HSTCP[1]

High Speed TCP (HS-TCP) is an enhanced version of TCP that reacts better when using large congestion windows on high-bandwidth, high-latency networks. High Speed TCP focuses specifically on a change to the TCP response function, and its implications for TCP.

b. STCP[17]

The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

c. FAST TCP[2]

In FAST-TCP a link model is used which captures the queue dynamics when congestion windows of TCP sources change. By using this model FAST TCP is always linearly stable with a single bottleneck link. FAST-TCP extends the existing stability results on homogeneous FAST flows to cases with heterogeneous delays.

d. HTCP[1]

This protocol is used for deployment in high-speed and long-distance networks. In H- TCP window growth function is based on is based on real time. When this protocol is deployed in conventional networks, H-TCP behaves as a conventional TCP-variant.

e. BIC[15]

To solve the RTT unfairness problem, the Binary Increase Congestion control (BIC) protocol is used. BIC supports TCP friendliness and bandwidth scalability. It uses two window size control policies called additive increase and binary search increase.

f. CUBIC[16]

CUBIC extends the work of BIC TCP (Binary Increase Congestion Transmission Control Protocol). The linear window growth function of standard TCP to cubic function is modified in CUBIC**.** To enhance the scalability of TCP in long distance network scenarios, CUBIC is used.

f. XCP[13]

The explicit Control Protocol (XCP) is a multi-level net-work feedback mechanism for congestion control of Inter-net transport protocols. XCP is stable and efficient over high bandwidth-delay product paths, while being more scalable to deploy than mechanisms that require per-flow state in routers.

g. VCP

The Variable-structure Congestion -control Protocol (VCP) extends the work of XCP. VCP is a window-based protocol and is designed to regulate the cwnd with different conges-tion control policies according to the level of congestion in the network. VCP applies MI, AI and MD policies in three regions of congestion known as Low-load, High-load and over-load regions respectively.

h. RCP-AC

In comparison with other congestion control protocols like TCP Reno and XCP, to complete one to two orders of magnitude, Rate Control Protocol(RCP) enables typical Inter-net-sized flows as fast as possible. But RCP faces some problems. Rate Control Protocol with Acceleration Con-troll (RCP-AC) extends the work of RCP. RCP-AC allows the aggressiveness of RCP to be tuned, enabling fast completion of flows over a broad set of operating conditions.

i. DPCP[20]

Double Packet Congestion- control Protocol (DPCP) extends the work of VCP, in that it utilizes two ECN bits of a pair of packets in order to use the ECN bit a distributed way. In this case, for a given load-factor (LF) the packet that carries LSB of the LF is referred to as LSP and the packet that carries the MSB of the LF is referred to as MSP.

j. MPCP

The Multi Packet Congestion Control Protocol (MPCP) is a novel distributed ECN-based congestion control protocol. By utilizing only two ECN bits MPCP is able to relay a more precise congestion feedback. In MPCP each packet carries two of 2n bits in its ECN bits.

2.4 Satellite Network

a. P-XCP

Explicit Control Protocol (XCP) is a promising transport layer protocol for satellite IP networks. But XCP has some challenges while operating in satellite network. These challenges are low throughput under high link error rate

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

conditions, and output link underutilization in the presence of rate-limited connections. To address the these problems P-XCP protocol is used.

2.5 TCP-Friendly [22] i. Single Rate

a. RAP

The Rate Adaption Protocol (RAP) is a simple AIMD scheme. In this scheme each data packet is acknowledged by the receiver and the ACKs are used to detect packet loss and infer the RTT.

b. LDA+

Like RAP, LDA+ is essentially an AIMD congestion con-troll scheme, but it uses some interesting additional elements. The Loss-Delay Based Adaption Algorithm (LDA+) relies on the Real-Time Transport Control Protocol (RTCP).Feedback messages provided by the Real-Time Transport Protocol (RTP). LDA+ is designed only for unicast com-munication.

c. TFRC

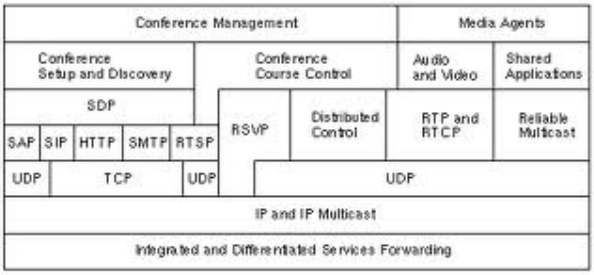
The TCP-Friendly Rate Control Protocol (TFRC) is de-signed for unicast as well as multicast communication. TFRC supports additional delay- based congestion avoid-ance by adjusting the inter packet gap to improve protocol performance in environments that do not fulfill the assumptions of the complex TCP equation, The main importance of TFRC is that is has a relatively stable sending rate while providing sufficient responsiveness to competing traffic.

d. TEAR

TCP Emulation at Receivers (TEAR) is a hybrid protocol because it combines aspects of window-based and rate-based congestion control. In this case sender adjusts the sending rate. TEAR protocol does not directly use the congestion window (cwnd) but calculates the TCP sending rate.

e. RLA & LPR

The Random Listening Algorithm (RLA) extends the work of TCP selective ACK (SACK) by introducing some en-hatchments for multicast. A TCP-like retransmission scheme with fast recovery is also included in RLA. RLA achieves the statistical long-term fairness. Linear Proportional Response (LPR), is a probabilistic loss indication filtering scheme that is an improvement over the corresponding RLA mechanism. As compare to RLA the LPR scheme achieves better fairness of multicast session's to-ward competing unicast sessions. LPR achieves good TCP friendliness in comparison with RLA when combined with the window adjustment mechanism.



f. MTCP

To achieve TCP friendliness, Multicast TCP (MTCP) is a reliable multicast protocol that uses window-based congestion control. In MTCP a logical tree structure is used where the root of the tree is the sender of the data. A parent in the logical tree structure stores a received packet until receipt is acknowledged by all of its children. Upon receiving a packet, a child transmits an ACK to its parent using unicast. The main problem of MTCP is its complexity.

g. NCA & PGMCC

Nominee-Based Congestion Avoidance (NCA) and pragmatic general multicast congestion control (pgmcc) these two protocols shares the same idea. In this approach congestion control and packet repair are treated independent of each other. This approach is used in reliable, as well as for unreliable data transmission. The most challenging aspect of NCA and pgmcc is how to select the group representative.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

II. MULTI RATE

a. RLC

Receiver-Driven Layered Congestion Control (RLC) protocol is used to dimension the layers so that the bandwidth consumed by each new layer increases exponentially. In case of RLC, ranularity at which the rate can be adapted to the network conditions is very coarse and may cause unfair behavior.

b. FLID-DL

To address some of the problems of RLC, Fair Layered Increase/Decrease with Dynamic Layering (FLID-DL).This protocol uses a digital fountain at the source. It introduces the concept of dynamic layering. The FLID-DL protocol extends the work of RLC.

c. LTS

The Layered Transmission Scheme (LTS) is used for the transmission of video. LTS is easy to implement but it suffer from a multitude of drawbacks.

d. MLDA

The Multicast Loss-Delay Based Adaption Algorithm (MLDA) is a congestion control protocol that uses layered multicast. It uses the combination of two protocols that are LDA+ and RTCP reports for the signaling between the sender and the receivers. Thus, it combines sender-based and receiver- based congestion control. The problem of MLDA is the added complexity of the application that has to distribute the data onto the dynamic layers.

e. Rainbow

Rainbow is a window-based congestion control scheme which is used for the reliable transfer of bulk data. In this case the data is encoded using a digital fountain. The main idea behind Rainbow is that receivers individually request the transmission of each data packet

2.6 Wired Network

a. Enhanced RED

In this approach ERQD algorithm is used for congestion avoidance in wired networks. The main idea behind this algorithm is to optimize the value of the average size of the queue used for congestion avoidance and to consequently reduce the total loss of packets at the queue and also reduces the Queue delay. This algorithm reduces the number of packet losses at the gateway and also reduces the queue delay.

2.7 Wireless Network

a. **DCCP**

DCCP, provide an efficient congestion control mechanism for heterogeneous wired-cum-wireless networks by using Congestion Control Identification (CCID) framework. DCCP evaluates a congestion control mechanism that implicitly discriminates congestion and wireless losses.

b. ECN2

Some of the protocols in network do not work well in wire-fewer networks because they take packet losses or timeout as the signal of congestion. The ECN2 protocol is based on expanded ECN mechanism to respond packet losses in wireless environment.

c. WTCP[10]

WTCP is a reliable transport protocol that addresses rate control and reliability over commercial WWAN networks such as CDPD. WTCP uses only end-to-end mechanisms and performs rate control at the receiver, and uses inter-packet delays as the primary metric for rate control. WTCP performs better than other protocols like TCP-RENO.

d. TCP_Westwood [11]

TCP Westwood (TCPW) is a sender-side modification of the TCP congestion window algorithm. TCPW improves upon the performance of TCP Reno in wired as well as wireless networks. An important distinguishing feature of TCP Westwood with respect to previous wireless TCP “ex-tensions” is that it does not require inspection and/or interception of TCP packets at intermediate nodes.

e. AIRMAIL[8]

Asymmetric Reliable Mobile Access In Link-layer (AIR-MAIL) is a link-layer protocol for indoor and outdoor wire-less networks. AIRMAIL is asymmetric to reduce the processing load at the mobile.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 7, July 2014

III. CONCLUSION

This paper discusses various congestion control protocols in network. Designing an efficient congestion control protocol which solves all problems of congestion is very difficult. Hence a survey of different congestion-control protocols, comparing the various features is absolutely essential to come up with new proposals for congestion–control in network. The performance of congestion-control protocols depend on various parameters. Thus this paper has come up with an exhaustive survey and comparison of different classes of congestion-control protocols.

REFERENCES

[1] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in *Proc. PFLDNet*, Feb. 2004.
[2] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, architecture, algorithms, performance," in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp.2490–2501.
[3] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One more bit is enough," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 37–48.
[4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 756–769, Dec.1997
[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," in Proc.10thACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2004, pp.168–177.
[6] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," in *Proc. 15th ICDCS*, Vancouver, BC, Canada, May 1995, pp. 136–143.
[7] Xiaolong Li, Homayoun Yousefi'zadeh,"Analysis, Simulation, and Implementation of VCP: A Wireless Profiling," IEEE/ACM Trans. Netw., vol. 18, no. 5, Oct.2010
[8] C. Parsa and J. Garcia-Luna-Aceves, "Improving TCP performance over wireless networks at the link layer," *Mobile Netw. Appl.*, vol. 5, no. 1, pp. 57–71, Mar. 2000.
[9] I. A. Qazi and T. Znati, "On the design of load factor based congestion control protocols for next-generation networks," in *Proc. of the IEEE INFOCOM 2008*, Apr. 2008.
[10] N. Vasic, S. Kuntimaddi, and D. Kostic, "One Bit Is Enough: a Framework for Deploying Explicit Feedback Congestion Control Protocols," in *Proc. of the First International Conference on communication Systems and Networks' (COMSNETS)*, Jan. 2009.
[11] M. Goutelle, Y. Gu, and E. He, "A Survey of Transport Protocols other than Standard TCP," 2004, https://forge.gridforum.org/forum/forum.php?forum id=410.
[12] L. Xu, K. Harfoush, and I. Rhee, "Binary Increase Congestion Con-trol (BIC) for Fast Long-Distance Networks," in *Proc. of the IEEE INFOCOM*, 2004.
[13] I. Rhee and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," in *Proc. of the PFLDNet'05*, Feb. 2005.
[14] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *Proc. ACM SIGCOMM*, Aug. 2002.
[15] H. Yousefi'zadeh, X. Li, and A. Habibi, "An End-to-End Cross-Layer Profiling Study of Congestion Control in High BDP Wireless Networks," in *Proc. of the IEEE WCNC, 2007*, Mar. 2007.
[16] Stevens W. TCP Slow Start, Congestion Avoidance, Fast Retrans-mit, and Fast Recovery Algorithms" RFC 2001, 1997.
[17] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in Proc. 8th Conf. Eur. Chap. Assoc. Comput. Linguist, Morristown, NJ: Assoc. Comput. Linguist. 1997, pp. 174–181.
[18] Wu-chang Feng, Kang G. Shin, Dilip D.Knadlur, Debanjan Saha. The BLUE active queue management algorithms. IEEE/ACM Transactions on Networking, 2002,10(4):513-528
[19] Athuraliya S., Low S. H., Li V. H., and et al. REM: Active Queue Management, IEEE Network, 2001, 15(3): 48-53.
[20] Srijith K.N., Jacob L., Ananda A.L.TCP Vegas-A: Improving the performance of TCP Vegas, Computer Communications,2005,28(4):429-440