Design of Congestion Prevention Algorithms in Computer Networks

Vinod Babu Vellank[1], Suresh Angadi[2]

[1]Student, Department of ECE, KL University, Andhra Pradesh,India
vvinod.08@gmail.com

[2]Associate Professor, Department of ECE, KL University, Andhra Pradesh, India

Abstract: ***Congestion is said to occur in the network when the resource demands exceed the capacity and packets are lost due to too much queuing in the network. A congestion avoidance scheme allows a network to operate in the region of low delay and high throughput. Such schemes prevent a network from entering the congested state. Congestion avoidance is a prevention mechanism while congestion control is a recovery mechanism. We compare the concept of congestion avoidance with that of flow control and congestion control. We model the network and the user policies for congestion avoidance as a feedback control system. The key components of a generic congestion avoidance scheme are: congestion detection, congestion feedback, feedback selector, signal filter, decision function, and in-crease/decrease algorithms. These components have been explained.***

Keywords-Increase/decrease algorithms, flow control, congestion control, feedback system.

1. Introduction

Recent technological advances in computer networks have resulted in a significant increasein the bandwidth of computer network links. The ARPAnet was designed in the 1970s usingleased telephone lines having a bandwidth of 50 Kbits/second. In the 1980s, local areanetworks (LAN) such as Ethernet and Token rings have been introduced with a bandwidth in the range of 10 Mbps. In this second half of the same decade, efforts are underway to standardize fiber optic LANs with a bandwidth of 100 Mbps and higher.The steadily increasing bandwidth of computer networks would lead one to believe thatnetwork congestion is a problem of the past. In fact, most network designers have found theopposite to be true. Congestion control has been receiving increased attention lately dueto an increasing speed mismatch caused by the variety of links that compose a computernetwork today. Congestion occurs mainly at routers (intermediate nodes, gateways, or IMPs)and links in the network where the rate of incoming traffic exceeds the bandwidth of thereceiving node or link.The problem of congestion control is more difficult to handle in networks with connectionlessprotocols than in those with connection-oriented protocols. In connection-oriented networks,resources in the network are reserved in advance during connection setup. Thus, one easyway to control congestion is to prevent new connections from starting up if congestion is sensed. The disadvantage of this approach, like any other reservation scheme, is thatreserved resources may not be used and may be left idle even when other users have beendenied permission. Rather than get in to the religious debate between followers of theconnection-oriented and connectionless disciplines, we simply want to point out the factthat the problem of congestion control in connectionless protocols is more complex. It is thisset of protocols that we are concerned with here.We are concerned with congestion avoidance rather than congestion control. The distinctionbetween these two terms is a rather subtle one. Briefly, a congestion avoidance schemeallows a network to operate in the region of low delay and high throughput. These schemesprevent a network from entering the congested state in which the packets are lost. We willelaborate on this point inthe next section where the terms flow control, congestion control,and congestion avoidance have been defined and their relationship to each other has beendiscussed.We studied a number of alternative congestion avoidance schemes. In this report, we discuss the goals, the metricsused to quantify the performance, and the fundamental components involved in the designof any congestion avoidance scheme. We address the issue of fairness in the service offeredby a network. The role of algorithms for increase/decrease of the amount of traffic a usermay place on the network is discussed, as well as the impact of the fairness of a range.

2. Concepts

In this section we define the basic concepts of flow control, congestion control, and congestionavoidance. These three concepts are related but distinct. They are related because all threesolve the problem of resource management in the network. They are distinct because theysolve resource problems either in different parts of the network or in a different manner. We also point out how decreasing cost of memory, or increasing link bandwidth and processorspeed are not sufficient to solve these problems.

2.1 Flow Control

In data communications, flow control is the process of managing the rate of data transmission between two nodes to prevent a fast sender from outrunning a slow receiver. It provides a mechanism for the receiver to control the transmission speed, so that the receiving node is not overwhelmed with data from transmitting node. Flow control should be distinguished from congestion control, which is used for controlling the flow of data when congestion has actually occurred. Flow control mechanisms can be classified by whether or not the receiving node sends feedback to the sending node. Flow control is important because it is possible for a sending computer to transmit

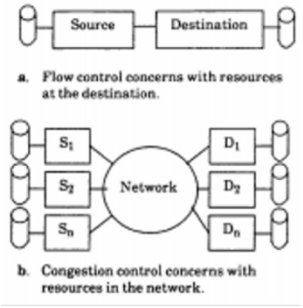
information at a faster rate than the destination computer can receive and process it. This can happen if the receiving computers have a heavy traffic load in comparison to the sending computer, or if the receiving computer has less processing power than the sending computer.

2.2 Congestion Control

Congestion results when one part of the subnet becomes overloaded. Because routers are receiving packets faster than they can forward them, one of two things must happen:

- The subnet must prevent additional packets from entering the congested region until those already present can be processed.
- The congested routers can discard queued packets to make room for those that are arriving.

A congestion control scheme protects the network from being flooded by its users. In connection oriented networks the congestion problem is generally solved by reserving theresources at all routers during connection setup. In connectionless networks it can be doneby explicit messages (choke packets) from the network to the sources or by implicitmeans such as timeout on a packet loss.



a. Flow control concerns with resources at the destination.

b. Congestion control concerns with resources in the network.

2.3 Flow Control vs Congestion Control

It is clear from the above discussion that the terms flow control and congestion control aredistinct. Flow control is an agreement between a source and a destination to limit the flow ofpackets without taking into account the load on the network. The purpose of flow control isto ensure that a packet arriving at a destination will find a buffer there. Congestion controlis primarily concerned with controlling the traffic to reduce overload on the network. Flowcontrol solves the problem of the destination resources being the bottleneck while congestioncontrol solves the problem of the routers and links being the bottleneck. Flow control isbipartite agreement. Congestion control is a social (network-wide) law. Different connections on a network can choose different flow control strategies, but nodes on the network shouldfollow the same congestion control strategy, if it is to be useful. The two parties in flowcontrol are generally interested in cooperating whereas the n parties (e.g., different users) incongestion control may be non cooperative. Fairness is not an issue for the two cooperatingpartieswhereas it is an important issue for n competing parties.

2.4 Congestion Avoidance

Traditional congestion control schemes help improve the performance after congestion hasoccurred. If the load is small, throughput generally keeps up with theload. As the load increases, throughput increases. After the load reaches the network capacity, throughput stops increasing. If the load is increased any further, the queues startbuilding, potentially resulting in packets being dropped. The throughput may suddenly dropwhen the load increases beyond this point and the network is said to be congested.

3. Design Requirements

Before we discuss the various schemes for congestion avoidance and compare them it is helpful to point out some of the design requirements that we followed. These requirementshelped us limit the number of schemes for further study. The key requirements are: no control during normal operation, no extra packets, a connectionless network layer, and configurationindependence. We describe these requirements below.

3.1 No Control during Normal Operation

Congestion is a transient phenomenon. Networks are configured in such a way that, on anaverage, the network is not overloaded. We therefore refrained from schemes that wouldgenerate extra overhead during normal (under loaded) conditions. This ruled out the use of such techniques as sending encouragement packets to users during under loadand indicatingoverload by the absence of these packets.

3.2 No New Packets

The processing overhead for network services depends upon the number of packets and thesize of those packets. Performance measurements of existing implementations have shown that the number of packets affects the overhead much more than the size. Short acknowledgment messages cost as much as 50% of the long data messages. This is why piggybacking(combining two are more messages) helps reduce the overhead.In summary, adding an extra packet causes much more overhead than adding a few bits inthe header. We therefore preferred schemes that did not require generation of new messagesand concentrated instead on adding only a few bits in the header.

3. 3 Distributed Control

The scheme must be distributed and work without any central observer. Thus, schemes where all routers send congestion information to a central network control center were considered unacceptable.

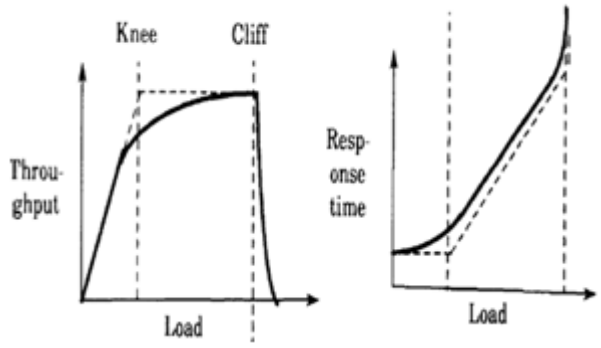
3.4 Connectionless Network Layer

The key architectural assumption about the networks is that they use connectionless networkservice and transport level connections. By this we mean that a router is not aware of thetransport connections passing through it, and the transport entities are not aware of thepath used by their packets. There is no prior reservation of resources at routers before anentity sets up a connection. The routers cannot compute the resource demands except byobserving the traffic owing through them.Examples of network architectures with a connectionless network layer are DOD TCP/IP,Digital Network Architecture (DNA), and ISO Connectionless Network Service.

4. Performance Metrics

The performance of a network can be measured by several metrics. The commonly usedmetrics are: throughput, delay, and power. Throughput is measured by the user bits transmitted per unit of time. Thus, protocol over-head, retransmissions, and duplicate packets are not considered in throughput computation.

Some of the more important applications of computer networks are: file transfer, mail, andremote login. The first two are throughput sensitive. The response time (time for the packetto reach the destination) is generally not so important. On the other hand, for remote login,response time is more important than throughput.The aforementioned goal, maximizing throughput and minimizing response time, are mutually contradictory in that all methods to increase throughput result in increased responsetime as well and vice versa. To resolve this contradiction, Giessler proposed the following metric:



Power $= \frac{Throughput^{\alpha}}{Response\ ime}$

Here α is a positive real number. Notice that by maximizing power, one tries to maximizethroughput and minimize response time. Normally, α= 1, i.e.increasing throughput anddecreasing response time are given equal weights. By setting α>1, one can favor file transfer by emphasizing higher throughput. Similarly, by setting α<1 one can favor terminal trafficby emphasizing lower response time.It must be pointed out that the throughput and response time used above are system-widethroughput (total number of packets sent for all users divided by the total time) and systemwide response time (averaged over all users) giving us system power. The operating point obtained in this manner is different from the one that would be obtained if each of the n user sites to maximize their own individual power (ratio of individual throughput and individualresponse time). Maximizing individual power leads to a number of undesirable effects.

5. Components of an Avoidance Scheme

The two key components of any congestion avoidance scheme, the feedback mechanism andthe control mechanism, have already been discussed earlier in this report. We call thesenetwork policies and user policies, respectively. This allows us to concentrate on one component at a timeand test various alternatives for that particular component. During the analysis, it can beassumed that other components are operating optimally. Of course, one would need to verifyat the end that the combined system worked satisfactorily under imperfect conditions.

The network policy consists of three algorithms: congestion detection, feedback filter, andfeedback selector. The user policy also consists of three algorithms: signal filter, decisionfunction, and increase/decrease algorithm. These generic algorithms apply to many different congestion avoidance schemes. For example, these six algorithms would apply whetherwe choose to implement network feedback in the form of source quench messages or we implement it via a field in the packet header.

5.1 Congestion Detection

Before the network can feedback any information, it must determine its state or load level.In a general case, the network may be in one of n possible states. The congestion detection function helps map these states into one of the two possible load levelsoverload or underload(above or below the knee). A k-ary version of this function would result in k levels of loadindications. A congestion detection function, for example, could work based on the processorutilization, link utilization, or queue lengths.

5.2 Feedback Filter

After the network has determined the load level, it may want to verify that the state lasts for a sufficiently long period before signaling it to the users. This is because a feedback ofstate is useful only if the state lasts long enough for the users to take action based on it.A state which changes very fast may lead to confusion. By the time users become aware ofthe state, it no longer holds and the feedback is misleading. Therefore, we need a (low-pass) filter function to pass only those states that are expected to last long enough for the useraction to be meaningful. Examples of feedback filters are exponential weighted average ormoving average of processor utilization, link utilization, or queue lengths.

5.3 Feedback Selector

After the network has determined that it is overloaded (or underloaded) and has ensuredthat the state is likely to last long enough, it needs to communicate this information to usersso that they may reduce (or increase) the traffic. A feedback selector function may be usedto determine the set of users to be notified. In other words, the network may want all usersto reduce the tra_c or it may selectively ask some users to reduce and others to increase the traffic. In the simplest case, it may give the same feedback signal to all users.

5. 4 Signal Filter

The users receiving the feedback signals from the network (routers) need to interpret the signal. The first step in this process is to accumulate a number of signals. Due to the probabilistic nature of the network, all these signals may not be identical. Some may indicate thatthe network is overloaded while others may indicate that it is underloaded. The user needsto combine these to decide its action. Some examples of received signal filter are majorityvoting (50%),

or three-quarter majority (75%), or unanimous (100%). The percentage maybe used after applying a weighting function, for example, giving higher weight to recentsignals.

5.5 Decision Function

Once the user knows the network load level, it has to decide either to increase its load ordecrease its load. The function can be broken down into two parts: the first part determines the direction and the other determines the amount. These parts are called decision functionand increase/decrease algorithms, respectively.The decision function takes feedback signals for the last T seconds, for instance, as inputparameter, and determines the load level of the network path. The key parameter is T the interval for which it should accumulate feedback. This determines the window updatefrequency. We will further discuss window update frequency later in this report.In its simplest form a decision function may be a 2-way function indicating whether theload should be increased or decreased. Some would argue that it may be a 3-way functionincluding a gray area where no action is taken.Another generalization often mentioned is to make a decision but not act on it unless wereach the same decision again, one or more times in the future. This may seem to increasethe probability of reaching the right decision.Both the generalizations mentioned above result in postponement of the action thereby causing the system to stay in the same state longer. This may be useful if the goal (knee) isstable but in a computer network the knee is a continuously moving target and it is helpful to reconfirm the state by perturbing the load, however slightly, one way or the other.

3 Increase/Decrease Algorithm

The key part of a control scheme is the control, i.e.the action taken as a result of thefeedback. For congestion avoidance schemes this part lies in the increase/decrease algorithmsused by the users. These algorithms are a key to achieving efficiency as well as fairness. Thechoice of other components of the congestion avoidance scheme depends upon the type offeedback chosen, whereas, the increase/decrease algorithms can be discussed and analyzedgenerically in great detail and apply to several feedback mechanisms.

6. Congestion Avoidance Schemes

Congestion control and congestion avoidance are dynamic system control issues. Like all other control schemes they consist of two parts: a feedback mechanism and a control mechanism. The feedback mechanism allows the system (network) to inform the users (source ordestination) of the current state of the system. The control mechanism allows the users toadjust their load on the system. The feedback signal in a congestion avoidance scheme tellsthe users whether the network is operating below or above the knee. The feedback signal ina congestion control scheme tells the users whether the network is operating below or above the cliff. The problem of congestion control has been discussed extensively in literature. A number of feedback mechanisms have been proposed. If we extend those mechanisms to signaloperations around the knee rather than the cliff, we obtain a congestion avoidance scheme.Of course, the control mechanism will also have to be adjusted to help the network operatearound the knee rather than the cliff. For the feedback mechanisms we have the followingalternatives:

1. Congestion feedback via packets sent from routers to sources.
2. Feedback included in the routing messages exchanged among routers.
3. End-to-end probe packets sent by sources.
4. Each packet contains a congestion feedback field that is filled in by routers in packets going in the reverse direction.
5. A congestion feedback field is filled in by routers in packets going in the forward direction.

The first alternative is popularly known as choke packet or source quench message inARPA net. It requires introducing additional traffic in the network during congestion,which may not be desirable. A complement to this scheme is that of encouraging sources toincrease the load during underload. The absence of these encouragement messages signalsoverload. This scheme does not introduce additional traffic during congestion. Nevertheless, it does introduce control overhead on the network even if there is no problem. The second alternative, increasing the cost (used in the forwarding database update algorithm) of congested paths, has been tried before in ARPAnet's delay-sensitive routing. Thedelays were found to vary too quickly, resulting in a large number of routing messages andstability problems. The third alternative, probe packets, also suffers from the disadvantage of added overheadunless probe packets had a dual role of carrying other information in them. If the latterwere the case, there would be no reason not to use every packet going through the networkas a probe packet. We may achieve this by reserving a field in the packet that is used by thenetwork to signal congestion. This leads us to the last two alternatives.The fourth alternative, reverse feedback, requires routers to piggyback the signal on thepackets going in the direction opposite the congestion. This alternative has the advantage in that the feedback reaches the source faster. However, the forward and reverse traffic are not always related. The destinations of the reverse traffic may not be the cause of or even theparticipant in the congestion on the forward path. Also, many networks (including DNA)have path splitting such that the path from A to B is not necessarily the same as that fromB to A. The fifth alternative, forward feedback, sends the signal in the packets going in the forwarddirection (direction of congestion). The destination either asks the source to adjust the loador returns the signal back to the source in the packets (or acknowledgments) going in thereverse direction. This is the alternative that we finally chose for further study.The minimal forward feedback requires just one bit of feedback signal with every packet.

Although at first, one bit may not appear to be able to carry enough information, we show in the second part of this report series that there is considerable performance gain evenby single-bit feedback.Most of the discussions in this and associated reports center around window-based control mechanisms. However, we must point out that this is not a requirement. Thecongestion avoidance algorithms and concepts can be easily modified for other forms of flow

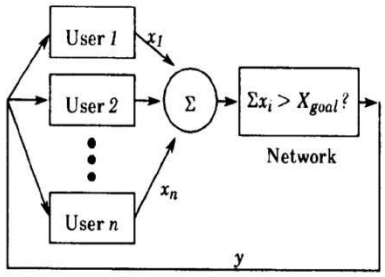
control such as rate-based flow control in which the sources must send below a rate(packets/second or bytes/second) specified by the destination. In this case, the users would adjust rates based on the signals received from the network.

7. Increase/Decrease Algorithm

The key part of a control scheme is the control, i.e., the action taken as a result of thefeedback. For congestion avoidance schemes this part lies in the increase/decrease algorithmsused by the users. These algorithms are a key to achieving efficiency as well as fairness. Thechoice of other components of the congestion avoidance scheme depends upon the type offeedback chosen, whereas, the increase/decrease algorithms can be discussed and analyzedgenerically in great detail and apply to several feedback mechanisms. We discuss some ofthese alternatives in the next section.

In this section we compare a number of alternative algorithms for window increase anddecrease. We show that an additive increase, multiplicative decrease algorithm provides fairand stable operation and that it is important to keep windows as real valued variables which are rounded of to the nearest integer.

We assume that the source and destination transport entities are using a window-based flow control. Thus, increasing the window increases the load on the network and decreasingthe window decreases the load. It must be pointed out, however, that all the argumentsapply equally well to other forms of flow control such as rate based flow-control, in which the destination permits the source to send data at a pre-specified rate (bits/second or pack-ets/second). In this case, it is obvious that increasing the rate increases the load and viceversa.



A control system model of n users sharing a network.

$$x_i(t+1) = x_i(t) + f_i(x_i(t), y_i(t))$$

A general increase (or decrease) algorithm would take the current control (flow-control window) and feedback signals as input arguments and produce the new control as an outputargument. However, as discussed above, we assume that the feedback signals have beenanalyzed by other components of the congestion avoidance scheme and the decision providedto this component is to increase or decrease the traffic. Thus, the key parameter to theincrease/decrease algorithms is the current window. We considered two types of increase/decrease algorithms:

1. Additive - The window is increased or decreased by a fixed amount.

$$x_i(t+1) = a_I + x_i(t)$$

$$x_i(t+1) = a_D - x_i(t)$$

2. Multiplicative - The window is increased or decreased by a fixed multiple.

$$w\ x_i(t+1) = b_I x_i(t); r1 > 0$$

$$w\ x_i(t+1) = b_D x_i(t);\ 0 < r2 < 1$$

Here, we concentrate on choosing one of the following four combinations:

1. Multiplicative Increase, Multiplicative Decrease

$$x_i(t+1) = \begin{cases} b_I x_i(t) & if\ y(t) = 0 => Increase, \\ b_D x_i(t) & if\ y(t) = 1 => Decrease \end{cases}$$

2. Multiplicative Increase, Additive Decrease

3. Additive Increase, Additive Decrease

$$x_i(t+1) = \begin{cases} b_I x_i(t) & if\ y(t) = 0 => Increase, \\ a_D + x_i(t) & if\ y(t) = 1 => Decrease \end{cases}$$

$$x_i(t+1) = \begin{cases} a_I + x_i(t) & if\ y(t) = 0 => Increase, \\ a_D + x_i(t) & if\ y(t) = 1 => Decrease \end{cases}$$

4. Additive Increase, Multiplicative Decrease

$$x_i(t+1) = \begin{cases} a_I + x_i(t) & if\ y(t) = 0 => Increase, \\ b_D x_i(t) & if\ y(t) = 1 => Decrease \end{cases}$$

In all these alternatives we assume that the computed value is rounded to an integer valueand that the window is never allowed to go below 1.The two key requirements of the increase/decrease policy are that it should allow a single usernetwork to operate as close to optimality as possible and that it should allow a multi-usernetwork to operate as fairly as possible. In comparing the above alternative we will assume a simplified model of the network in which all users share the same path and therefore receivethe same feedback. If i^{th} user has a window w_i, the network gives the signal to go up if and
only if:

$$\sum_{i=1}^{n} w_i \leq w_{knee}$$

Here, $w_{knee}$is the window at the knee of the throughput (or response time) curve for the given network configuration.The fairness goal dictates that regardless of any starting point all n users should converge to the same final window w_{knee}/n. While going down, the users with higher windows shouldgo down more than those with lower windows, i.e., the decrease should be proportional (multiplicative).

8. Conclusion

We have seen the problems due to congestion and various methods to overcome congestion through various congestion policies. We have discussed about various metrics used in congestion control. We examined the user increase/decrease policies under the constrain of binary signal feedbackWe formulated a set of conditions that any increase/decrease policy should satisfy to ensure

convergence to efficiency and fair state in a distributed mannerWe show the decrease must be multiplicative to ensure that at every step the fairness either increases or stays the same

References

[1] V. Ahuja, \Routing and Flow Control in Systems Network Architecture," IBM SystemsJournal, Vol. 18, No. 2, 1979, pp. 298 - 314.

[2] K. Bharat-Kumar, \A New Approach to Performance-Oriented FlowControl," IEEE Transactions on Communications, Vol. COM-29, No. 4, April 1981, pp.427 - 435.

[3] W. Bux and D.Grillo, \Flow Control in Local Area Networks of Interconnected TokenRings," IEEE Transactions on Communications, Vol. COM-33, No. 10, October 1985,pp. 1058-66.

[4] Dah-Ming Chiu and Raj Jain, \Congestion Avoidance in Computer Networks with aConnectionless Network Layer. Part III: Analysis of Increase/Decrease Algorithms,"Digital Equipment Corporation, Technical Report #TR-509, August 1987.

[5] David Clark, \NETBLT: A Bulk Data Transfer Protocol," Massachusetts Institute ofTechnology, Lab for Computer Science, RFC-275, February 1985.

[6] Digital Equipment Corp., \DECnet Digital Network Architecture NSP Functional Specification, Phase IV, Version 4.0.0," March 1982.

[7] M..Gerla and L. Kleinrock, \Flow Control: A Comparative Survey," IEEE Transactionson Communications, Vol. COM-28, No. 4, April 1980, pp. 553 - 574.