Volume 6, Issue 4, April 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Congestion Control Issues & Trends

Akshay Mishra
CSE, ACTS, Satna, Madhya Pradesh,
India

Nirmala Sinha
CSE, PSIT, Kanpur, Uttar Pradesh,
India

Abstract— Popular myths that cheaper memory, high-speed links, and high-speed processors will solve the problem of congestion in computer networks are shown to be false. A simple definition for congestion based on supply and demand of resources is proposed and is then used to classify various congestion schemes. The issues that make the congestion problem a difficult one are discussed, and the architectural decisions that affect the design of a congestion scheme are presented. It is argued that long, medium-, and short-term congestion problems require different solutions. Some of the recent schemes are briefly surveyed, and areas for further research are suggested.

Keywords— Introduction, Myths, problems difficult, polices, fundamental principle, summary, references.

I. INTRODUCTION

Congestion control is concerned with allocating the resources in a network such that the network can operate at an acceptable performance level when the demand exceeds or is near the capacity of the network resources. These resources include bandwidths of links, buffer space (memory), and processing capacity at intermediate nodes. Although resource allocation is necessary even at low load, the problem becomes more important as the load increases because the issues of fairness and low overhead become increasingly important. Without proper congestion control mechanisms, the throughput (or net work) may be reduced considerably under heavy load.

In this paper, we begin with several myths about congestion and explain why the trend toward cheaper memory, higher-speed links, and higher-speed processors has intensified the need to solve the congestion problem. We then describe a number of proposed solutions and present a classification of congestion problems as well as their solutions. In Section 4 we explain why the problem is so difficult. In Section 5, we discuss the protocol design decisions that affect the design of a congestion control scheme. Finally, we describe our recent proposals and suggest areas for future research.

II. MYTH ABOUT CONGESTION CONTROL

Congestion occurs when the demand is greater than the available resources. Therefore, it is believed that as resources become less expensive, the problem of congestion will be solved automatically. This has led to the following myths:

1. Congestion is caused by a shortage of buffer space and will be solved when memory becomes cheap enough to allow infinitely large memories.
2. Congestion is caused by slow links. The problem will be solved when high-speed links become available.
3. Congestion is caused by slow processors. The problem will be solved when the speed of the processors is improved.
4. If not one, then all of the above developments will cause the congestion problem to go away.

The congestion problem can not be solved with a large buffer space. Cheaper memory has not helped the congestion problem. It has been found that networks with infinite-memory switches are as susceptible to congestion as networks with low-memory switches.On the other hand, with infinite-memory switches, the queues and the delays can get so long that by the time the packets come out of the switch, most of them have already timed out and have been retransmitted by higher layers. In fact, too much memory is more harmful than too little memory since the packets (or their retransmissions) have to be dropped after they have consumed precious network resources.

The congestion problem can not be solved with high-speed links. In the beginning, the telephone links connecting computers had a speed of a mere 300 bits per second. Slowly, the technology improved, and it was possible to get dedicated links of up to 1.5 Mbits per second. Then came the local area networks (LANs), such as Ethernet, with a speed of 10 Mbits per second. It was precisely at this point that the interest in congestion control techniques increased. This is because the high-speed LANs were now connected via low-speed, long-haul links, and congestion at the point of interconnection became a problem.

The following experiment, although a contrived one, shows that introducing high-speed links without proper congestion control can lead to reduced performance. The time to transfer a particular file was five minutes. After the link between the first two nodes was replace by a fast 1 Mbits per second link, the transfer time increased to seven hours! With the high-speed link, the arrival rate to the first router became much higher than the departure rate, leading to long queues, buffer overflows, and packet losses that caused the transfer time to increase.

Congestion occurs even if all links and processors are of the same speed. Our arguments above may lead some to believe that a balanced configuration with all processors and links at the same speed will probably not be susceptible to congestion. This is not true. The conclusion is that congestion is a dynamic problem. It cannot be solved with static solutions alone. We need protocol designs that protect networks in the event of congestion. The explosion of high-speed networks has led to more unbalanced networks that are causing congestion. In particular, packet loss due to buffer shortage is a symptom not a cause of congestion.

III. A CLASSIFICATION OF CONGESTION PROBLEMS AND SOLUTIONS

In simple terms, if, for any time interval, the total sum of demands on a resource is more than its available capacity, the resource is said to be congested for that interval. Mathematically speaking:

∑Demand > Available Resources

In computer networks, there are a large number of resources, such as buffers, link bandwidths, processor times, servers, and so forth. If, for a short interval, the buffer space available at the destination is less than that required for the arriving traffic, packet loss occurs. Similarly, if the total traffic wanting to enter a link is more than its bandwidth, the link is said to be congested.

The above definition of congestion, although simplistic, is helpful in classifying congestion problems as well as solutions. Depending upon the number of resources involved, a congestion problem can be classified as a single resource problem or a distributed resource problem, as shown in Figure 4. The single resource involved may be a dumb resource, such as a LAN medium, in which case, all the intelligence required to solve the congestion problem has to be provided by the users. Various LAN access methods, such as CSMA/CD (Carrier Sense Multiple Access with Collision Detection), token access, register insertion, and so on, are examples of solutions to the problem of single, dumb resource congestion. If the resource is intelligent, for example, a name server, it can allocate itself appropriately. The problem is more difficult if the resource is distributed as in the case of a store and forward network. For example, considerint the links as the resources, the user demands have to be limited so that the total demand at each link is less than its capacity. It is this set of problems dealing with distributed resource congestion that we are concerned with in this paper.

The simple definition of congestion above also allows us to classify all congestion schemes into two classes: those that dynamically increase the available resource, and those that dynamically decrease the demand. Some examples of both these types of schemes are described below.

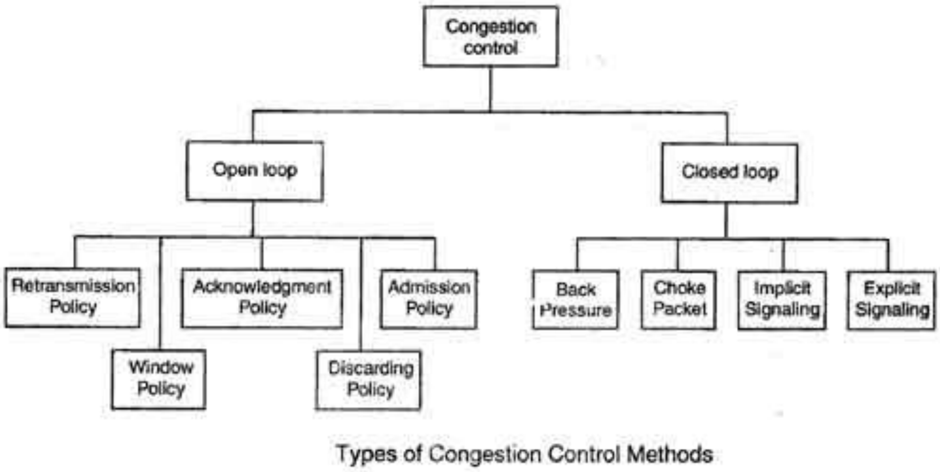

Figure 1: Types of congestion control methods.

1. Resource Creation Schemes: Such schemes increase the capacity of the resource by dynamically reconfiguring them. Examples of such schemes are:
   - Dial-up links that can be added only during high usage.
   - Power increases on satellite links to increase their bandwidths.
   - Path splitting so that extra traffic is sent via routes that may not be considered optimal under low load.

   With all of the above schemes, users of the resource do not need to be informed, as they may not even be aware of the congestion in the network. The network is solely responsible for solving the congestion problem.
2. Demand Reduction Schemes: These schemes try to reduce the demand to the level of the available resources. Most of these schemes require that the user (or other control points) be informed about the load condition in the network so they can adjust the traffic. There are three basic classes of such schemes:
   - Service Denial Schemes: These schemes do not allow new sessions to start up during congestion. The busy tone provided by the telephone company is an example of such a scheme. Connection-oriented computer networks also use similar schemes where congestion at any intermediate node would prevent new sessions from starting up.
   - Service Degradation Schemes: These schemes ask all users (existing as well as new users) to reduce their loads. Dynamic window schemes in which the users increase or decrease the number of packets outstanding in the network based on the load are examples of this approach.

- Scheduling Schemes: These schemes ask users to schedule their demands so that the total demand is less than the capacity. Various contention schemes, and polling, priority, and reservation schemes are examples of this approach. It must be pointed out that all scheduling schemes are a special case of the service degradation approach.

In connectionless networks, starting a new session does not require that all intermediate resources be informed, so the service denial approach cannot be effectively used. Such networks generally use service degradation and scheduling techniques. All congestion control schemes, resource creation as well as demand reduction schemes, require the network to measure the total load on the network and then to take some remedial action. The first part is often called feedback, while the second part is called control. Depending upon the load, a feedback signal is sent from the congested resource to one or more control points, which then take remedial action. In demand reduction schemes, the control point is generally the source node of the traffic, while in resource creation schemes, the control points may be other intermediate nodes (or sources) on the network. A number of feedback mechanisms have been proposed, for example:

- Feedback Messages: Explicit messages are sent from the congested resource to the control point. Such messages have been called choke packets, source quench messages, or permits. The sources reduce their loads upon the receipt of choke packets [24] or source quench messages and increase it if these are not received. In the isarithmic scheme [6], the sources have to wait to receive a permit before sending a packet. Critics of this approach argue that the extra traffic created by the feedback messages and permits during heavy load may worsen the congestion.
- Feedback in Routing Messages: Each intermediate resource sends its load level (typically in terms of queue length or delay) to all neighboring nodes who then adjust the level of traffic sent to that resource. The delay adaptive routing used in ARPAnet at one time is an example of this approach. This method was found to generate too many routing messages, since the rate of change of delay through a node was much faster than the rate at which control could be affected.
- Rejecting Further Trafic: In this approach, no explicit messages are sent. However, incoming packets are either lost or not acknowledged, thereby, creating a backpressure. This results in queues being built at other nodes, which then backpressure their neighbors. The backpressure slowly travels towards the source. This technique is useful only if the congestion lasts for a very short duration. Otherwise, the traffic that is not even using the congested resources is unfairly affected by the backpressure propagating throughout the network.
- Probe Packets: This requires sources to send probe packets through the network and to adjust their loads depending upon the delay experienced by the probe packets.

A number of alternatives for the location of control have also been proposed:

- Transport Layer: The traffic is generated by the end systems, therefore, they are in the best position to adjust the load in an efficient manner. Dynamic window schemes are an example of such controls at the transport layer. If the network and the end systems are under different administrative control, such as in public networks, the control may be exercised between the first and the last intermediate systems (entry-to-exit or DC% to-DCE) instead of between the end systems.
- Network Access: Like traffic lights at the entrance ramps of some highways, the access controls at the network layer of the source node allow new traffic to enter the network only if the network is not congested.
- Network Layer: The routers and gateways, if congested, can take immediate action by reducing service to the sources that are sending more than their fair share. The fair queueing scheme, various buffer class schemes, and the leaky bucket algorithm are examples of this approach. These schemes are particularly useful for public networks, which may not be able to ensure that the end systems will reduce the load on a congestion feedback signal.
- Data Link Layer: The control can also be exercised at the data link level at each hop using data link level flow control mechanisms. Backpressure on buffer exhaustion is one such scheme.

IV. A CLASSIFICATION OF CONGESTION PROBLEMS AND SOLUTIONS

Despite the fact that a number of schemes have been proposed for congestion control, the search for new schemes continues. The research in this area has been going on for at least two decades [lo]. There are two reasons for this. First, there are requirements for congestion control schemes that make it difficult to get a satisfactory solution. Second, there are several network policies that affect the design of a congestion scheme. Thus, a scheme developed for one network may not work on another network with a different architecture. In this section, we elaborate on the first issue of requirements. The second issue of network policies is discussed in the next section.

The scheme must have a low overhead. In particular, it should not increase traffic during congestion. This is one of the reasons why explicit feedback messages are considered undesirable. Some researchers have suggested that feedback be sent only during low load, thus, the absence of feedback would automatically indicate a high load. Even such schemes are not desirable, since the network resources are also used for nonnetworking applications. Therefore, resources consumed to process these additional messages could have been better used by these other applications. The scheme must be fair. Fairness may not be important during low load when everyone’s demands can be satisfied. However, during congestion when the resources are less than the demand, it is important that the available resources be allocated fairly. Defining fairness is not trivial. A number of definitions have been proposed. However, no one definition has been widely accepted. For example, some researchers consider starvation of a few users to be unfair. Not allocating any resources to a user is called starvation. By this definition, if all users get a nonzero share of the resources, the scheme is fair. Others argue that a scheme without starvation can still be unfair if the resources are allocated unevenly. The key problem is defining what is an even distribution of resources in a wide-area network where different users are traveling different distances. Some want to give preference to traffic that has traveled a long distance (more hops), while others want to give equal throughput to all users. The definition of users is also not clear. Some researchers treat each source-destination pair as a user. Giving equal throughput to all source-destination pairs passing through an intermediate node does not automatically guarantee that all connections from a single source will be treated fairly. Finally, the scheme must be socially optimal. That is, the scheme must allow the total network performance to be maximized. Schemes that consider each user in isolation may be individually optimal, but not socially optimal. For example, if each user attempted to maximize its throughput, it may lead to an unstable situation where total network load keeps increasing.

It should be clear from the above list of requirements that designing a congestion control scheme is not a trivial problem.

V. POLICIES THAT AFFECT THE CONGESTION CONTROL SCHEME

Any architectural or implementation decision that affects either side of Equation 1 affects the design of a congestion control scheme. Thus, any design decision affecting the load (demand) or resource allocation can be considered a part of the overall congestion control strategy of the network. These decisions are called policies in this paper. A list of such policies is presented in Table I.

The most important network policy is the connection mechanism. There are two types of networks: connection-oriented and connectionless.

1. Network Layer:
   - Connection mechanism
   - Packet queuing and service policy
   - Packet drop policy
   - Packet routing policy
   - Lifetime control policy

2. Transport Layer:
   - Round-trip delay estimation algorithm
   - Timeout algorithm
   - Retransmission policy
   - Out-of-order packet caching policy
   - Acknowledgment policy
   - Flow control policy
   - Buffer management policy

3. Data Link Layer:
   - Data link level retransmission policy
   - Data link level queuing and service policy
   - Data link level packet drop policy
   - Data link level acknowledgment policy
   - Data link level flow control policy

Packet queuing and service policies in the intermediate nodes affect resource allocation among users. An intermediate node may have separate queues for each output link, each input link, or a combination of the two. In some networks, there is a separate queue for each source and, thus, fairness among all sources can be guaranteed. The packet drop policy deals with the issue of which packet is dropped if there is insufficient buffer space in a queue. Some of the alternatives are the first packet in the queue, the last packet in the queue (the arriving packet), or a randomly selected packet. The choice depends upon the type of application. For real-time communications, the older the message, the less valuable it is. Therefore, it is better to drop packets at the head of the queue. This type of traffic has been called ‘milk’ and is contrasted with file and terminal traffic, which has been called ‘wine’ because older messages are more valuable than newer ones. To ensure fairness, some have proposed random dropping, but others have argued its effectiveness .

The route selection policy, in general, and the path splitting policy, in particular, affect the resource allocation and, hence, congestion in the network. In most networks today, a low-speed path will be totally unused even if a parallel high-speed path is congested. Path splitting is performed only across paths of the same speed or across parallel links connecting the same nodes (one hop).

Lifetime control policies affect the length of time a packet stays in the network before being dropped. There may be too many unnecessary retransmissions (and, hence, load) if the lifetime is either too short or too long.

The round-trip delay estimation and the timeout interval computation algorithms used by the transport protocol also have a significant impact. In fact, finding a good algorithm for estimating round-trip delay in the presence of packet loss has been the first step towards finding a solution for congestion control. Reducing the probability of false timeout alarms using the mean as well as the variance of the round-trip delay also improves the efficiency of congestion control mechanisms using timeouts .

The number of packets retransmitted on a packet loss affects the stability of timeout-based congestion schemes. The optimal number may depend upon the out-of-order packet caching policy at the destination.

The packet acknowledgment policy affects the feedback delay in congestion information reaching back to the source. If every packet is acknowledged, there may be too much traffic but the congestion feedback is fast. If some acknowledgments are withheld, the load due to acknowledgments is less, but the congestion feedback is delayed more.

The data link level policies are similar to the transport layer policies except that they apply to each hop in the network. For example, the intermediate systems in the network may have their own packet caching, acknowledgments, retransmission, and flow control policies. All of these will affect the design of the congestion control scheme.

In summary, there are a large number of architectural decisions that affect the design of a congestion control scheme. This is why analysts comparing the same set of alternatives may reach different conclusions.

VI. A FUNDAMENTAL PRINCIPLE OF CONTROL

As the name indicates, the problem of congestion control is basically a control problem. Most congestion control schemes consist of a feedback mechanism and a control mechanism. In control theory, it is well known that the control frequency should be equal to the feedback frequency. As shown in Figure 5, if the control is faster than the feedback, the system will have oscillations and instability. On the other hand, if the control is slower than the feedback, the system will be tardy and slow to respond to changes. In designing congestion schemes it is important to apply this principle and to carefully select the control interval. In many existing schemes this is ignored, and although a feedback mechanism such as the source quench is specified, the issue of how often to send feedback and how long to wait before acting is left unspecified. This leads to schemes that are later found ineffective.

VII. AREAS FOR FURTHER RESEARCH

Although congestion control is not a new problem, there are considerable opportunities for research. In this section, we point out several issues that need to be resolved.

Path splitting among long paths of differing capacities is not well understood. In most networks today, all traffic from a given source to a given destination either passes through the same path or is split equally among different paths of equal capacities. Thus, if the optimal path is congested and a slower path is available, the slower path is not used. Designing a scheme that allows slower paths to be used depending upon the load levels on all paths is a topic for further research.

Insulating one level of network hierarchy from congestion in other levels is another area for research. Most large networks are organized hierarchically into several levels. Schemes are required that prevent congestion at one level from affecting the traffic at other levels. Thus, congestion of a backbone network should not affect other networks and vice versa.

Congestion control in integrated networks with voice, data, and several other types of traffic is also an interesting research problem. Giving higher priority to voice traffic, a commonly proposed solution, does not suite all environments. In some cases, such as real-time applications, the delay and throughput requirements are complex, and accommodating them in a congestion control scheme is nontrivial. As the telecommunication industry is moving towards asynchronous transfer mode (ATM), which uses short, fixed-size packets (cells), the congestion control schemes for such networks are being heatedly debated in several standards committees.

Heterogeneous networks consisting of networks using several different architectures need implicit feedback schemes for congestion control and avoidance. This problem was mentioned earlier.

Dynamic link creation schemes that require the dialing up of a new link need to be developed. When a link should be dialed up or disconnected depends upon the tariff structure. Now that high-speed, dial-up links are becoming available, it would be interesting to have guidelines regarding their usage.

Server congestion is a recent problem that started occurring with the introduction of distributed systems. After a power failure, all nodes in a building need access to the name server, boot server, and so on. Unless the access is regulated properly, the server can get congested with requests and may be so late in responding that the requests are retransmitted, thus causing an unnecessary additional load on the servers. Schemes to solve this problem need to be developed.

REFERENCES

[1] K. Bharat-Kumar and J. M. Jaffe, “A New Approach to Performance-Oriented Flow Control,” IEEE Transactions on Communications, Vol. COM-29, No. 4, April 1981, pp. 427-435.

[2] W. Bux and D. Grille, “Flow Control in Local-Area Networks of Interconnected Token Rings,” IEEE Transactions on Communications, Vol. COM-33, No. 10, October 1985, pp. 1058-66.

[3] D. Cheriton, ‘Sirpent: A High Performance Internetworking Approach,” Proc. ACM SIGCOMM’89 Symposium on Communications Architectures and Protocols, Austin, TX, September 1989, pp. 158169.

[4] D. M. Chiu and R. Jain, “Analysis of Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” Computer Networks and ISDN Systems, Vol. 17, 1989, pp. 1-14.

[5] D. Cohen, “Flow Control for Real-Time Communication,n Computer Communication Review, Vol. 10, No. 1-2, January/April 1980, pp. 41-47.

[6] D. W. Davies, “The Control of Congestion in Packet-Switching Networks,” IEEE Trans. Commun., Vol. COM-20, No. 6, June 1972.

[7] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” Proc. ACM SIGCOMM’89 Symposium on Communications Architectures and Protocols, Austin, TX, September 1989,

[8] B. T. Doshi and H. Q. Nguyen, “Congestion Control in ISDN Frame-Relay Networks,” AT&T Technical Journal, November/December 1988, pp. 3546.

[9] F. D. George and G. E. Young, “SNA Flow Control: Architecture and Implementation,” IBM System Journal, Vol. 21, No. 2, 1982, pp. 179-210.

[10] M. Gerla and L. Kleinrock, “Flow Control: A Comparative Survey,” IEEE Transactions on Communications, Vol. COM-28, No. 4, April 1980, pp. 553574. PI.] M. Gerla, H. W. Ghan, and J. R. Boisson de Marca, “Fairness in Computer Networks,” Proc. IEEE International Conference on Communications ICC’85, Chicago, IL, June 23-26, 1985, pp. 43.5.1-6.

[11] E. L. Hahne and R. G. Gallager, URound Robin Scheduling for Fair Flow Control in Data Communications Networks,” Proc. IEEE International Conference on Communications ICC’86, Toronto, Canada, June 22-25, 1986, pp. 4.3.1-5.

[12] M. Irland, “Buffer Management in a Packet Switch,” IEEE Trans. on Commun., Vol. COM-26, March 1978, pp. 328-337.

[13] V. Jacobson, ‘Congestion Avoidance and Control,” Proc. ACM SIGCOMM’88, Stanford, CA, August 1988, pp. 314-329. [15] J. M. Jaffe, “Bottleneck Flow Control,” IEEE Transactions on Communications, Vol. COM-29, No. 7, July 1981, pp. 954-962.

[14] R. Jain, D. M. Chiu, and W. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems, Digital Equipment Corporation, Technical Report DEC-TR-301