CONTROL OF NETWORKS: Basic Concepts and Fundamental Techniques

S.Georgievska

Institute of Informatics, Faculty of Natural Sciences and Mathematics
Sts. Cyril and Methodius University
Arhimedova bb, P.O.Box 162, Skopje, Macedonia
sonja@ii.edu.mk

Abstract: In present high-performance communication networks, the control strategies play a central role in the quality of services. In a public network that sells services to users, better control leads to greater revenue. In a private network, it results in lower costs for services.

There are four basic means of control: admission control, routing, flow and congestion control, and allocation control. The means available depend on whether the transport method used by the network is circuit, datagram, or virtual circuit switching.

Keywords: datagram networks, virtual circuit-switched networks

1. Introduction

A network that combines an ATM transport layer, or an IP layer with some form of quality of service or class of service, with a high-speed physical layer such as SONET can potentially provide the large range of quality of service (QoS) necessary to support most applications. However, in order to provide this range of QoS, the network resources (bandwith and buffers) must be properly managed or controlled.

In this text we take a look at the concepts and fundamental techniques used to control datagram and virtual circuit-switched networks in order to achieve efficient use of network resources. We shall see how different control techniques affect different network performance measures. There are four principal methods: admission control, routing, flow and congestion control, and resource allocation. Flow-control procedures for datagram networks attempt to prevent a source from overwhelming a destination. Tipically, the destination informs the source when it is becoming congested, and the source stops transmitting. The congestion-control mechanism is used to prevent some internal nodes of the network from becoming congested. Datagram networks (such as the Internet), generally, do not guarantee delay or troughput. Their congestion control attempts to reduce the average delay per packet for any given troughput. Admission control determines which virtual circuit connection requests are accepted by the network. This is similar to the admission of telephone calls by the telephone network. Normally, a datagram network always accepts the packets (datagrams) submitted by a user and does not exercise admission control. Lastly, in virtual circuit switching, the network can control the bandwith and buffers allocated to each virtual circuit. This is called (resource) allocation control.

In this article we pay attention to control of datagram networks.

2. Control of Datagram Networks

In datagram networks, the routes taken by successive packets may be different, even if they go from the same source to the same destination. Also, because the packet sizes may be different, the transmission times of the packets may be different, since they are equal to the packet lengths divided by the transmission rate. The queuing model of Figure 1 can be used by the designer to predict the transmission delays and to design good routing and flow-control algorithms.

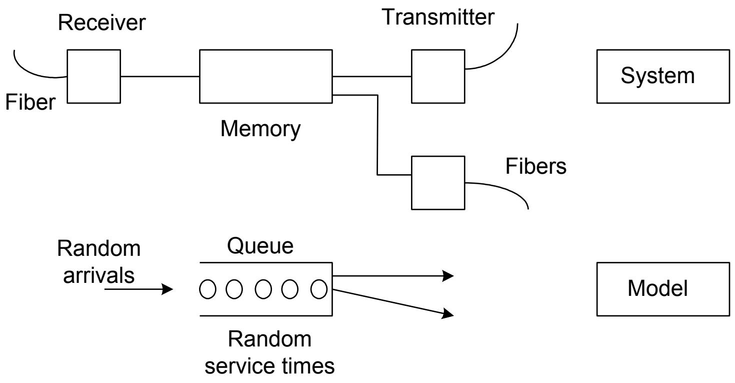

Figure 1:

Here, the service time is the length of the packet in bits divided by the transmission bit rate. The service time is random when the packet length is random. If the packets all have the same size, as in ATM, the service time is deterministic. Thus, the fluctuations of the arrival times and of the transmission times are modeled by random variables. The specific assumptions about the distributions of these random variables depend on the precise model being used.

The most useful result of queuing theory for the analysis ofa datagram networks concerns the network shown in Figure 2. That result is the formula for the average delay per packet in such a network.

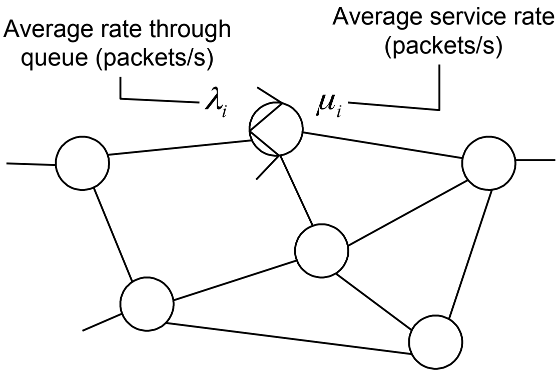
Average rate through queue (packets/s)
Average service rate (packets/s)
λ_i
μ_i

Figure 2:

The network consists of a collection of nodes connected by links. The packets that arrive from outside of the network are assumed to form Poisson processes, and the packet transmission times in the variuous nodes are assumed to be independent and exponentially distributed. We use the following notation. At each node j, packets arrive at rate λ_j packets/s, and the service rate is μ_j packets/s. Thus μ_j is the transmission rate in bits per second divided by the average packet size in bits. It is assumed that $\mu_j > \lambda_j$ for all j. Then the average delay faced by a packet entering the network is

$$T = \frac{1}{\gamma} \sum_j \frac{\lambda_j}{\mu_j - \lambda_j} \qquad (1)$$

where γ is the total rate at which packets arrive into the network.

The assumptions are not exactly satisfied in actual networks. For instance, the packet lengths are not exponentially distributed. Typically, the packets that travel on the Internet tend to have a bimodal distribution: most packets are either short or long, and the fractions of short and long packets are not consistent with an exponential distribution. Also, since the length of a packet does not change as the packet travels trough the network, the transmission times of a packet at the different nodes are not independent. In fact, if one knows the transmission time of a packet at one node, then one can determine the length of that packet and therefore its transmission times in all the other nodes.

Although these assumptions are not always valid, the formula for the average delay per packet provides a reasonably good estimate of the actual value of that average delay in a real network. This simple formula is the starting point for the construction of routing and flow-control algorithms.

2.1 Routing optimization

The optimal static routing problem for a general network is to minimize the average delay per packet T with respect to all routing probabilities.

$$\lambda_i = \gamma_i + \sum_j \lambda_j p_{ji}, \text{ for all i.} \quad (2)$$

The network sends a packet leaving node i to node j with probability p_{ij}. Given these routing probabilities p_{ij}, we can calculate the average rates of flow λ_i through the nodes i by solving the following flow-conservation equations:

In these equations, γ_i denotes the rate of arrivals of packets from outside the network into node i. The equations say that, for each i, the rate of flow λ_i through node i is equal to the external arrival rate into that node γ_i plus the sum over all nodes j of the fraction p_{ji} of the rate λ_j of flow leaving that node and being sent to node i. If the network is open, that is, if all the packets that enter the network can eventually leave it, then the flow-conservation equations have a unique solution $\{\lambda_i, i=1,...J\}$.

Thus, given the external rates $\{\gamma_i, i=1,...J\}$, the flow-conservation equations enable us to calculate the rates $\{\lambda_i, i=1,...J\}$ as a function of the routing probabilities p_{ij}.

Once these rates are determinated, we can use our formula (1) to compute the average delay T.

The delay T is a complicated function of the routing probabilities p_{ij}. The minimization of T does not result in a closed-form expression for the optimum routing probabilities. Instead, one must use a numerical minimization algorithm.

Instead of selecting (fixed) probabilities to route the packets when they leave the nodes, we can devise a dynamic routing algorithm that bases the routing decisions on the actual backlogs of the nodes. In this case the probabilities p_{ij} might depend on the queue lengths of the successive nodes.

Generally, the derivation of the optimal dynamic routing algorithm for a network with many nodes is a formidable problem that is still beyound the reach of current approaches. Consequently, approximations are necessary. Moreover, the state of the network is not known instantaneously, so that, even if it could be derived, the optimum dynamic routing agorithm would not be implementable. As a result of these limitations, simple heuristics have been developed that can be implemented. For example, consider the model of network given in Figure 3.

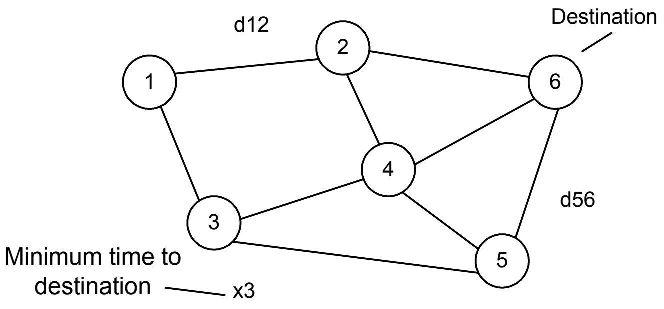

Figure 3:

The average delay on each link is estimated by the corresponding transmitter. One possible estimation method is for the transmitter on each link to keep track of the backlog in its buffer and to calculate the average delay by dividing the total number of bits stored in the buffer by the transmission rate. The propagation time of signals along the link can be added for improving the estimate.

$$x_i = \min_j \{d_{ij} + x_j\} \qquad (3)$$

Let us assume that the delay on the link from node i to node j has been estimated for all pairs of nodes. Let us denote by x_i the minimum delay between node i and some fixed destination. The minimum delay must satisfy the equation

These equations are of the form

$$x = F(x) \qquad (5)$$

where x designates the vector with components x_i. Thus, the vector x satisfies fixed-point equations. These fixed-point equations can be solved by the recursion

$$x^{n+1} = F(x^n) \quad (6)$$

The resulting algorithm is called the Bellman-Ford algorithm.

In IP, the routing actually uses a hierarchy. The network is decomposed into autonomous systems (ASs). An AS is a group of networks or subnetworks under a single administration. Within each AS, the routing algorithm in each area calculates the shortest path to all the other routers in the areas to which it belongs using some shortest path algorithm as the one presented above. The routing between ASs uses the Border Gateway Protocol. Each AS contains a router, called border gateway speaker, that implements BGP. BGP speakers use TCP to exchange routing tables and their updates.

2.2 Congestion Control

Congestion control is the name of control procedures that throttle the flow of packets along a path to keep points of the network from becoming congested.

In addition, the window congestion control mechanism is explained. The algorithm is called additive increase and multiplicative decrease. The TCP congestion-avoidance mechanism is of this type.

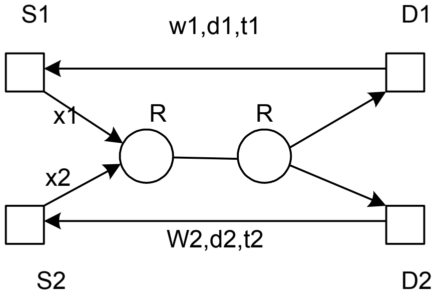

Figure 4:

Consider a network given in Figure 4. This is a set of interconnected routers and hosts. We assume that the links all have the same transmission rate of 1.Two connections share the network: from source S_1 to destination D_1 and from source S_2 to destination D_2. Figure 4 shows transmissions of acknowledgements from each destination D_i to the source S_i. The following quantities are defined for the connection from S_i to D_i: the propagation time d_i, the transmission rate x_i, the round-trip time t_i, and the number of bits in transit w_i. The router can store a finite number of bits. If bits arrive at the router when it is full, the bits are dropped. Since these bits do not reach the destination, they are not acknowledged and the source can realize, after some delay, that the bits were dropped.

Source S_i implements a window congestion control that limits the number of bits in transit to w_i. The transmission links that the packets from S_i to D_i and their acknowledgements from D_i to S_i go through have a total propagation time equal to d_i. That is, when there is very little trafic in the network, the time between the transmission of a packet and the reception of its acknowledgement by S_i is d_i. This delay d_i is called the propagation time of the connection. The round-trip time t_i of the connection is equal to the propagation time plus the queuing time of the packets in the routers. If the links of the network are used efficiently, then x1+x2=1. Also, if the two connections have a fair share of the links, then x1=x2=0.5. The window congestion control algorithm is a mechanism to adjust the window size w_i based on the observed quantities x_i,t_i. The difficulty in designing an algorithm of this type is that it must work for a source that does not know the network topology, the link rates, nor the set of other connections with which it is sharing the network.

Tha main idea of the algorithm follows: the sources increase their transmission rate until they detect dropped bits. When a source detects that some of its bits were dropped, it reduces its transmission rate by some factor. But, an analysis of this algorithm reveals that even for two connections that share a single bottleneck, the additive increase and multiplicative decrease window congestion algorithm is biased in favor of the connection with a smaller propagation time.

To correct this bias, hte *random early drop* (RED) mechanism has been designed. A router that implements RED drops incoming packets with a probability that is a function of the average recent buffer occupancy. If that average value is below a low threshold, then the router accepts the packet. If the average is larger than a high threshold, then the router drops the packet. If the average is between these thresholds, then the router drops the packet with a probability that increases linearly with the average. The effect of this mechanism is that sources learn on the congestion before the router buffer is full and so get a chance to slow down before facing multiple succesive losses. Moreover, RED is more likely to drop packets from faster connections (since they send more packets). Consequently, faster connections are more likely to slow down than slow connections, which somewhat corrects the TCP bias.

3. Conclusion

Routing optimization and congestion control algorithms for datagram networks are complex problems which still have not been solved completely. Modelling computer networks requires many assumptions on the behaviour of the network and implementation of heuristic algorithms for control.

4. References:

1. Warland J., Varaiya P.: *"High-performance Communication Networks"*, Morgan Kaufmann 2000
2. Gelenbe E., Pujolle G.: *"Introduction to Queuing Networks"*, Wiley, 1998.