컴퓨터 통신망의 경로 및 흐름제어 효과

°이 광 재* 김 정 선**
*, ** 한국항공대학 전자공학과

Effect of Routing and Flow Control in Computer Communication Networks

Kwang Jae Lee* Jung Sun Kim **
*, ** Dept. of Avionics, HanKuk Aviation College.

ABSTRACT

Successful operation of a computer network is dependent on the adaquate routing and flow control. Routing and flow control algorithms are studied together rather than as isolated mechanism. In this paper, we discuss the effect of the flow control and routing algorithms.

1. Introduction

Computer communication networks which defined packet-switched or store-and-foward networks typically have some problems:

(1) Packet loss and delay due to its speed matching at the nodes.

(2) Decrease the throughput when the network load is heavy.

Thus, from these problems, the overall congestion control strategy of a computer networks specifies how that network protect itself from potential overload. and how network resources are allocated to network users. And successful operation is dependent on the adaquate routing and flow control.

Routing and flow control algorithms are studied together rather than as isolated mechanism, since to do otherwise can lead to dangerous conclusions about differences in performance in real networks.[1].

This paper deals with several properties of the effect of routing and flow control algorithms. The principle performance evaluation employed are average packet delay as a function of throughput and throughput as a function of the offered load.

2. Network model

The results discussed here are for the 19 node network shown Fig.1 and Fig.2 shows the system model for this network.

- (i,k) : a link from node i to node k.
- f_{ik} : traffic flow on (i,k).
- C_{ik} : traffic capacity on (i,k).
- r_i^j : expected traffic from node i to node k.
- t_i^j : total expected traffic.
- ϕ_{ik}^j : routing variable.

The end-to-end flow control scheme is acting on source traffic with rate R_i^j by limiting the input capacity to C_{si}^j for traffic destined for node j. Then r_i^j =

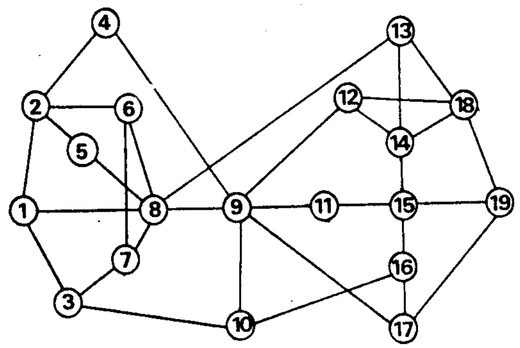

Fig. 1. The 19 node network.

$\min[R_i^j\ C_{si}^j]$. The queue in source is a function of R_i^j/C_{si}^j, where the input capacity C_{si}^j is a time varying quantity.[4]

Let $D_{ik}(f_{ik})$ be the expected number of packet/s transmitted on link (i,k) times the expected delay/packet. Also define the total delay number of packet arrivals/s. This is given in terms of $D_{ik}(f_{ik})$ as

$$D_T = \sum_{ik} D_{ik}(f_{ik}) \qquad (1)$$

The expected end-to-end delay per packet is simply Davg $= D_T/\gamma$, where γ is the expected number of packet/s entering the networks. Thus, Davg is minimized by minimizing D_T.[2][3]

Define the marginal delay, M_{ik}^j, as the increase in D_T due to an incremantal increase in traffic from node i to node j that is routed via link (i,k), i.e.,

$$M_{ik}^j = D'_{ik}(f_{ik}) + \partial D_T/\partial r_k^j \qquad (2)$$

where the incremental delay, $\partial D_T/\partial r_k^j$, can be computed as

$$\partial D_T/\partial r_k^j = \sum_m \phi_{km}^j \left[D'_{km}(f_{km}) + \partial D_T/\partial r_m^j \right] \qquad (3)$$

Assume that $D_{ik}(f_{i,k})$ is convex$\cup$ and continuousiy differentiable for $0 \le f_{ik} < C_{ik}$ for all links (i,k).

If a set of ϕ_{ik}'s exist for which all link flows satisfy $f_{ik} < C_{ik}$, then a sufficient

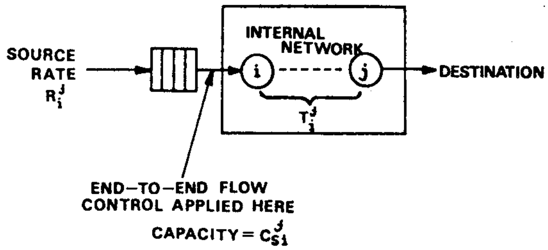

Fig. 2. System model.

condition for minizing D_T is that

$$M_{ik}^j \ge \partial D_T/\partial r_i^j \qquad (4)$$

for all $i \ne j$ and all links(i,k). This should be obvious since if a link (i,k) exists for which (4) is not satisfied, then one could increase slightly and reduce ϕ_{im}^j by the same amount on a link (i,m) which satisfies (4). This would result in a further decrease in D_T. Equation (4) is equivalent to

$$M_{ik}^j - B_i^j \ge 0, \qquad (5)$$

for all $i \ne j$ and all link (i,k) with equality for $\phi_{ik}^j > 0$ where

$$B_i^j = \min_m \left[M_{im}^j \right] \qquad (6)$$

The value of m corresponding to the minimum M_{im}^j is denoted by k_i^j.

3. Window flow control with routing

In this section, we examine the fixed window flow control with shortest path routing. To determine the shortest path route we use an shortest path algorithm.[7]

For a given set of (R_i^j), we can determine the steady-state set (r_i^j) by an iterative method utilizing Little's theorem to calculate approximate steady-state throughput. Little's theorem states that the expected number of packets in the network between source and destination is $N_i^j = r_i^j T_i^j$, where T_i^j is the

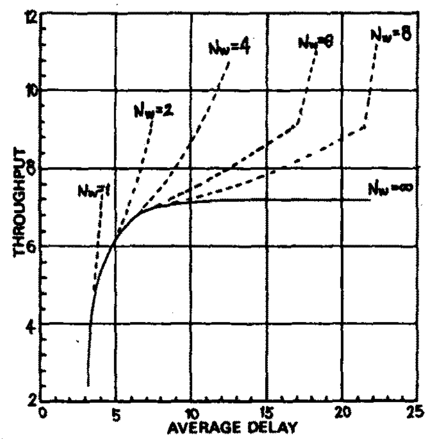

Fig.3. Average delay versus throughput for fixed window flow control and shortest path routing.

expected internal network end-to-end delay which can be calculated using the link utilization. In the iterative algorithm, we utilize the window size as a bound on N_i^j. The algorithm initially selects $r_i^j = R_i^j$ (although the algorithm will also work any initial value $0 \le r_i^j < R_i^j$) On each iteration we compute the delays T_i^j and the allowable throughput, $\gamma_i^j = Nw/T$ predicted by Little's theorem. The incremental adjustment of r_i^j are proportional to $(\gamma_i^j - r_i^j)$ with the constant that $r_i^j \le R_i^j$. This iterative adjustment continues until T_i^j satisfies Little's theorem for all S-D pairs, at which point the steady-state T and γ are found. This algorithm produces reasonably accurate results when load exceeds throughput. In addition, when the network is operating in the light flow control region the algorithm correctly sets $r_i^j = R_i^j$. Effective flow control is provided because early saturation of a few links does not occur, as with Nw=∞.

The model assumes infinite queues at the internal network nodes and buffers of length 5 at the network inputs. Short acknowledgment delays are modeled by making ACK's 1/16 the size of a normal data packet. Zero processing and propagation delays are also assumed. Using this model a different values Nw and load were simulated, and the results are also shown in Fig.3. One can see in Fig.3 the points along the knee of the Nw=∞ curve from which the finite window curves depart. These are boundaries between the light and heavy flow control region. For the larger window sizes, these boundaries are quite close. In the light control region, all of the offered load is carried by the network. In the heavy control region, one or more of the traffic rources is limited to a share of network capacity that is less than its traffic requirement.

4. Gallager routing and flow control

If the network traffic load is known and fixed, then one can solve for the optimum routing variables during network design.[5] Note that $(M_{ik}^j - \min_m [M_{im}^j])$ provides the direction in which to change ϕ_{ik}^j, as well as the proper magnitude. On each iteration of the algorithm, the (ϕ) set as well as incremental delay change by small amount until the optimum routes are finally obtained. The routing variables are updated according to

$$\phi_{ik}^{j}(new) = \begin{cases} \phi_{ik}^{j}(old) - \Delta\phi_{ik}^{j}, & k \neq k_i^j \\ \phi_{ik}^{j}(old) + \sum_{k \neq k_i^j} \Delta\phi_{ik}^{j}, & k = k_i^j \end{cases} \qquad (7)$$

The flow control algorithm just described produces steady-state solution that satisfy an upper bound on network conges-

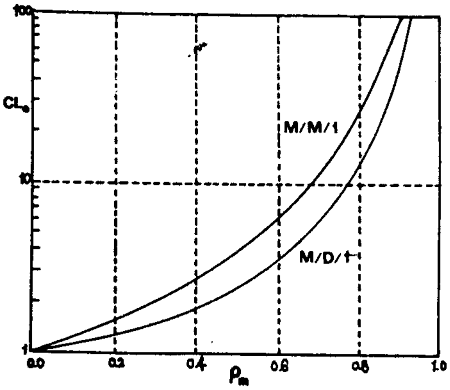

Fig.4. Required CL_0 to ensure that no link utilization exceeds

tion.For two of the most common models (M/M/1 and M/D/1)

$$D'(f) = d(\rho)/C \qquad (8)$$

where C is the link capacity and $d(\rho)$ is convexU. Then for every link

$$D'(f) \le L_0 \qquad (9)$$

This implies that every link throughout the network has utilizatoin $\rho<\rho_m$, where ρ_m is the solution of

$$d(\rho_m) = CL_0 \qquad (10)$$

Thus the flow control scheme can be designed to ensure that any desired ρ_m is never exceeded by any link in the network. The required value of CL_0 as a functuon of ρ_m is shown Fig.4 for the both M/M/1 and M/D/1 models.

Under heavy loading, the user closest to the destination can monopolize that link by requesting the entire capacity. In these cases, it is desirable to allow a disadvantaged user a small amount of capacity, even at the cost of more detrimental effects on network congestion. This can be accomplished with the same flow control concept by replacing the L_0 bound with a penalty function $L(r_i^j)$ that depends on the input rate r_i^j. We will use the normalized function

$$L(r) = \frac{a/C}{(b+r/C)^n} + d/C, \quad 0\le r<\infty \qquad (11)$$

where the constants a, b, d, and n may be chosen to give the desired shape.

In the selecting an appropriate $L(r)$ function, setting $L(0)=CL_0$, the penalty function constants must satisfy

$$a/b^n + d = CL_0 \qquad (12)$$

At no load the smallest incremental delay is that of a single hop path. For both the M/M/1 and M/D/1 models

$$D'(0)=1/C \qquad (13)$$

A desired zero-load source capacity C_0 is chosen in order to provide a specified zero-load source delay. Then to achieve this constraint for the single hop paths, parameters must be chosen such that

$$L(C_0)=1/C \qquad (14)$$

This gives

$$(a/(1-d))^{1/n} - b = C_0/C \qquad (15)$$

Of course, any nonzero load will produce an input capacity of $C_{s_i}^j < C_0$. For a path of h hops, the parameters must be chosen such that

$$(a/(h-d))^{1/n} - b = C_0/C \qquad (16)$$

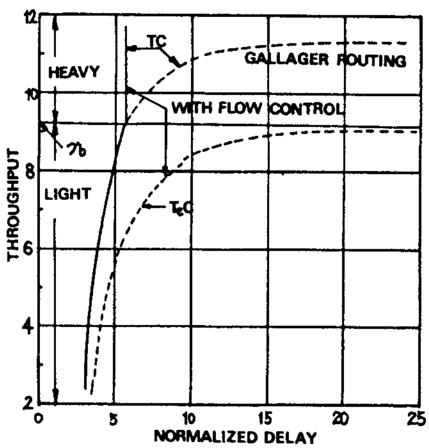

Fig.5. Influence of Gallager routing and flow control.

to guarantee all sources a zero-load input capacity of C_0.

Fig.5 shows comparision of normalized delay for the Gallager routing scheme with and without flow control. Note that one of the curves is the normalized end-to-end delay CT_e for the region of light flow control. It is only larger than the internal delay except near the boundary of heavy flow control.

There are two very useful quantities obtained by this model: the amount of internal network congestion and the boundary between the region of light and heavy flow control. This boundary is the throughput at which the first source bottleneck appears.

5. conclusion and further study

There is no single definition of an optimum routing and flow control algorithm. The adaptive algorithms described here have some useful features. An obvious point of the fixed window results is a tradeoff between throughput and network congestion, and one of the features of Gallager routing and flow control scheme is the ability to upper bound the expected amount of network congestion. But in actual environment, a number of quantities must be estimated such as link flows, system times and D'(f), also considered buffer management schemes with deadlock avoidance properties.

References

[1] H. Rudin and H. Mueller, "Dynamic routing and flow control," IEEE Trans. Commun., vol. COM-28, pp. 1030-1039, July 1980.

[2] J. B. Cain and G. H. Thaker, "Interactions between quasi-static routing and flow control algorithms," in Int. Conf. Commun Proc., Denver, co, 1981.

[3] J. B. Cain, D. B. Bradliy, and G. H. Thaker, "Simulation results for a quasi-static routing algorithm for packet-switched networks," in Nat. Telecommun. Conf. Proc., Houston, TX, 1980.

[4] G. H. Thaker and J. B. Cain, "Interaction between routing and flow control algorithms," IEEE Trans. Commun., vol. COM-34, pp. 269-277, March 1986.

[5] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," IEEE Trans. Commun., vol. COM-25, pp. 73-85, Jan. 1977.

[6] L. Kleinrock, Queueing Systems, Vol. II: Computer Applications. NewYork: Wiley, 1976.

[7] J. F. Hayes, Modeling and Analysis of Computer Communitions Networks. NewYork: Plenum Press. 1984.