Chapter 5

Networks with Tiny Buffers

Enachescu et al. [25] show that under TCP traffic, a single router can achieve close-to-peak throughput with buffer size $O(\log W_{\max})$. The crucial assumption in this result is a non-bursty input traffic (see Section 2.1.3).

In a network with arbitrary traffic matrix and topology, however, the traffic pattern of flows may change across the network. Although there has been no thorough analytical work on how an arbitrary network of FIFO buffers changes the traffic pattern, there are examples that show the traffic may become bursty as it traverses the network [27].

In this chapter, we explore whether a network can maintain high throughput when all its routers have tiny congestion buffers. The running assumption is that traffic is smooth at ingress ports of the network. To answer this question, we will first study networks with tree structure (Section 5.2), where there is no cross traffic. We will show that the buffer occupancy of a router in a tree-structured network does not exceed the buffer occupancy of an isolated router into which the ingress traffic is directly fed. Therefore, the single-router results can be applied to all routers in this network. For general-topology networks (Section 5.3), we will propose an active queue management policy (BJP) which keeps the packet inter-arrival times almost unchanged as the flows traverse the network. Therefore, if traffic is smooth at ingress ports, it remains so at every router inside the network, and hence, the single-router results hold for all routers in the network.

5.1 Preliminaries and assumptions

Throughout this chapter, we assume that flows go through only one buffering stage inside each router. Hence, we set the number of buffering stages equal to the number of routers and refer to them interchangeably.

For a given network and traffic rate matrix, we define the offered load on each link to be the aggregate injection rate of flows sharing that link. The *load factor*, ρ, of the network is the maximum offered load over all the links in the network. We assume that the network is over-provisioned (i.e., $\rho < 1$), all packets have equal sizes, and all buffers in the network have equal service rates of one packet per unit of time.

5.2 Tree-structured networks

In a tree-structured network, routers form a tree, where the traffic enters the network through the leaves of the tree. After packets are processed at a hop (router), they are forwarded to the next hop towards the root of the tree, which is the exit gateway of the network.

Consider a router R_m at distance m from the leaves in a tree-structured network (Figure 5.1, top). This router is the root of a sub-tree that receives traffic on a subset of the ingress ports. The queue size of router R_m (i.e., number of packets queued in the router) can be compared to that of an isolated router R, which directly receives the same ingress traffic (Figure 5.1, bottom). At a given time t, there is an arrival at router R if and only if there is one at the input ports of the sub-tree.

Assume that the propagation delay on the links is negligible. By comparing the queue sizes of these two routers, it can be shown that the drop rate at R_m with buffer size B is upper bounded as stated in the following lemma.

Lemma 5.1. With Poisson ingress traffic and a load factor $\rho < 1$, the drop rate at router R_m is smaller than or equal to ρ^B.

The proof is in Appendix B.

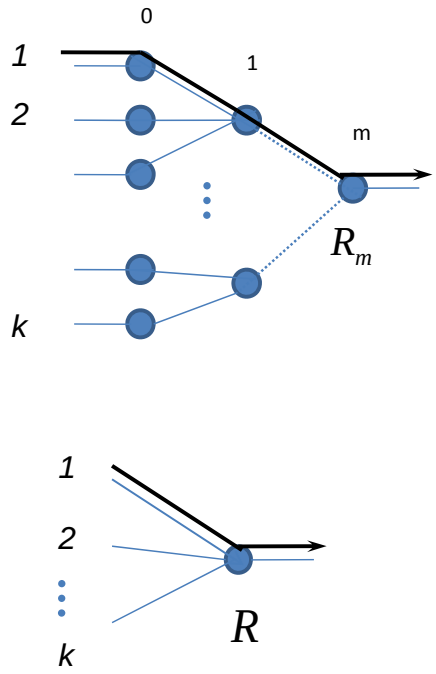

Figure 5.1: Tree-structured network (top). The buffer occupancy of router R_m in this network does not exceed the buffer occupancy of an isolated router R into which the ingress traffic is directly fed (bottom).

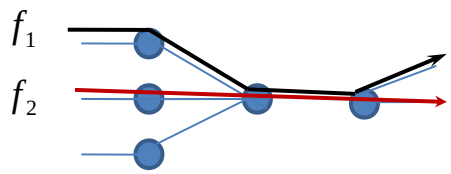

Figure 5.2: An example of a general-topology network.

This lemma implies that the overall packet drop rate will be less than $n\rho^B$ if each packet goes through at most n routers. The following theorem immediately follows.

Theorem 5.1. In a tree-structured network with Poisson ingress traffic and a load factor $\rho < 1$, packet drop rate of ϵ can be achieved if each router buffers

$$B \geq \log_{1/\rho}(\frac{n}{\epsilon})$$

packets, where n is the maximum number of routers on any route.

5.3 General-topology networks

Figure 5.2 shows an example of a network with general topology where flows can share a part of their routes and then diverge. The results obtained in Section 5.2 depend crucially on the tree-structured nature of the network (see Appendix B for details) and cannot be applied to a general network.

But what happens if routers in a general-topology network delay packets by exactly D units of time? Clearly, in this network the inter-arrival times of packets in each flow will remain intact as the flow is routed across the network. In particular, if the ingress traffic of the network is Poisson, the input traffic at each intermediate router will continue to be Poisson. However, delaying every packet by a fixed amount of time may not be feasible. Packets arriving in a burst cannot be sent out in a burst if

they arrive faster than the transmission capacity of the output interface. Therefore, some small variations should be allowed in the delay added by each router. This is the basic idea behind the Bounded Jitter Policy, which is explained below.

5.3.1 Bounded Jitter Policy (BJP)

Figure 5.3 illustrates the scheduling of BJP. This policy is based on delaying each packet by D units of time but also allowing a cumulative slack of Δ, $\Delta \leq D$. We call Δ the *jitter bound* of the scheme.

In order to implement BJP, packets need to be time stamped at each router. When a packet first enters the network, its time stamp is initialized to its actual arrival time. At each router on the route, the time stamp is updated and incremented by D units of time, regardless of the actual departure time of the packet.

Consider a router that receives a packet with time stamp t. The router tries to delay the packet by D units of time and send it at time $t + D$. If this exact delay is not possible, i.e., if another packet has already been scheduled for departure at time $t + D$, then the packet will be scheduled to depart at the latest available time in the interval $[t + D - \Delta, t + D]$. If there is not any available time in this interval, the packet will be dropped.

The packet scheduling of BJP is based on time stamps rather than actual times. It follows that if a packet leaves a router earlier than its departure time stamp $(t+D)$, it will likely be delayed more than D units by the next hop, and this can compensate the early departure.

The following theorem shows that if the ingress traffic is Poisson, BJP can achieve a logarithmic relation between the drop rate and the buffer size.

Theorem 5.2. In a network of arbitrary topology with Poisson ingress traffic, packet drop rate ϵ can be achieved if each router buffers

$$B \geq 4 \log_{1/\alpha}(\frac{n}{(1-\alpha)\epsilon})$$

packets, where n is the maximum number of routers on any route, $\alpha = \rho e^{1-\rho}$, and

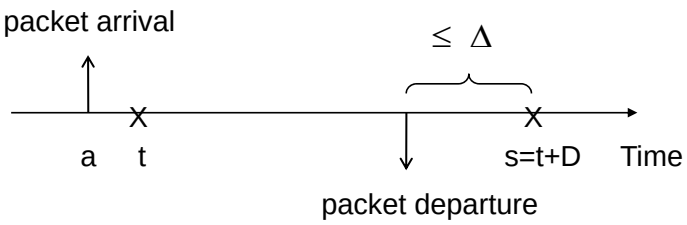

Figure 5.3: Packet scheduling under BJP. A packet that arrives at time a is scheduled based on its arrival time stamp $t \geq a$. The packet is scheduled to depart the buffer as close to $t + D$ as possible, where D is a constant delay.

$\rho < 1$ is the load factor of the network.

The proof is in Appendix C.

5.3.2 Local synchronization

Although BJP requires a time stamp to be carried with each packet inside the network, it only needs synchronization between the input and output linecards of individual routers, not a global synchronization among all routers. In fact, all that is needed is the slack time of the packet, not its total travel time. Consider a router that receives a packet at time a with a slack $\delta \leq \Delta$. Ideally, the router sends out the packet at time $a + D + \delta$. When the packet leaves the router, the difference between this ideal departure time and the actual departure time will be the updated time stamp of the packet.

5.3.3 With TCP traffic

As we explained in Section 2.1.3, the burstiness in TCP is not caused by the AIMD dynamics of its congestion control mechanism. Spreading out packets over a round-trip time can make the traffic smooth without needing to modify the AIMD mechanism.

To analyze the throughput of the network under smooth TCP traffic, we take an approach similar to [25] and assume that packet arrivals of each flow (at the rate dictated by the AIMD mechanism of TCP) follow a Poisson model. This lets us use the upper bound on the drop rate derived in the previous section.

Consider an arbitrary link l with bandwidth C packets per unit of time, and assume that N long-lived TCP flows share this link. Flow i has time-varying window size $W_i(t)$, and follows TCP's AIMD dynamics. In other words, if the source receives an ACK at time t, it will increase the window size by $1/W_i(t)$. If the flow detects a packet loss, it will decrease the congestion window by a factor of two. In any time interval $[t, t')$ when the congestion window size is fixed, the source will send packets as a Poisson process at rate $W_i(t)/RTT$. We also assume that the network load factor ρ is less than one. This implies that $\rho_l = \frac{N \times W}{RTT} < C$, where ρ_l is the offered load on link l. The effective utilization, θ_l, on link l is defined as the achieved throughput divided by ρ_l.

Under the above assumptions, the following theorem holds.

Theorem 5.3. To achieve an effective utilization θ on every link in the network, a buffer size of

$$B \geq 4 \log_{1/\alpha}(\frac{nW}{(1-\alpha)(1-\theta)})$$

packets suffices under the BJP scheme.

The proof is in Appendix D.