CMPSCI 754 Multimedia Systems **Spring 2001**

Lecture 24: May 1, 2001

Lecturer: Prashant Shenoy *Scribe: Puru Kulkarni*

24.1 Background

Following are some of the design issues of integrated services networks we have studied:

- Difference between traditional best-effort data applications and real-time audio and video services.
- The fact that the existing internet supports only best-effort services.
- Two possible approaches to designing integrated services networks:
  - Change the network and application implementation, but not the arhitecture. In this case the entwork delivers the same class of service to all applications, i.e. with no QoS guarantees and application development becomes more complex. One approach is to use packet scheduling algorithms in routers that eliminate interference of real-time applications with data applications. Can also employ admission control at routers to ensure guarantees are met by the network.
  - Change the architecture. In this case application development is easy, but requires fundamental changes to the existing networks. This is difficulut to deploy and can be done over a period of time.
- Flows for video are VBR and lead to packet losses due to burstiness. Packet loss can be avoided by *smoothing* (traffic shaping) at the source or regulate the rate of flow using network feedback.

Types of scheduling algorithms are:

- Non-work conserving: The algorithm can keep the resource idle in presence of work.
- Work conserving: The algorithm always keeps the resource busy as long as their is work to do.

It is desirable for the scheduling algorithm to be work conserving to get better utilzation. But non-work conserving algorithms can have some useful properties that are not provided by work conserving algorithms.

24.2 Delay Guarantee

Source sends packets with some flow characteristics. The source can specify that it wants to send p packets/sec and requires that a packet be scheduled before d seconds. The network performs admission control and provides guarantess(contract) to the source. The upper bound on the end-to-end packet delay depends on source traffic charateristics and scheduling algorithm used at the intermediate nodes.

24.2.1 Traffic shaper

A traffic shapper or rate regulator ensures that the incoming flow to the scheduler meets the characteristics specified according to the contract. Examples of traffic shapers are:

- CBR (Constant Bit Rate) regulator
- Leaky bucket shaper (allows bounded burstiness)

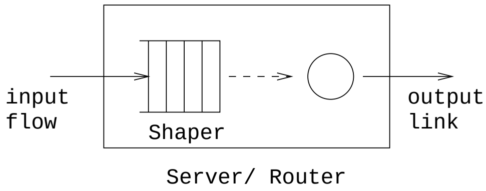

Figure 24.1: A source with a traffic shaper

The combination of a rate regulator and a packet scheduling algorithm is a non-work conserving algorithm. If the flow does not meet the characteristics specified (suppose it sends a burst of packets), the regulator regulates the flow and allows only the number of packets fixed in the contract even id the output link is free. This clearly is not good for providing high utilization. So the question is can we use a traffic shaper so that flow characteristics are maintained according to contract and still have a work conserving packet scheduling algorithm.

24.2.2 Fair Airport(FA) Algorithms

Fair Airport algorithms use a rate regulator for each flow and are also work conserving.

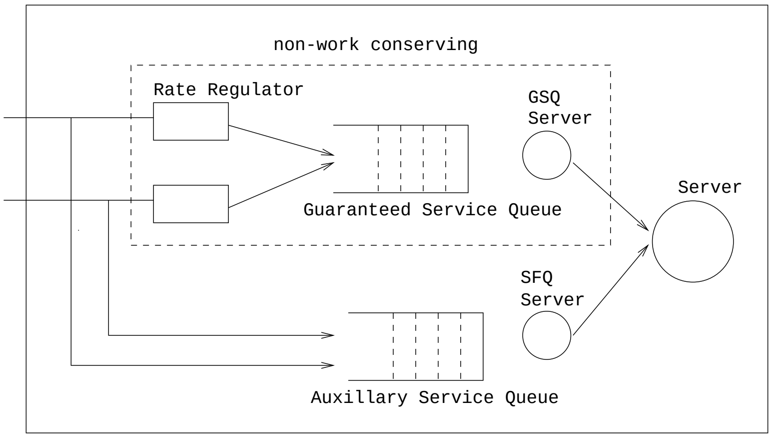

Figure 24.2: FA algorithm

- Each switch(router) that uses a FA algorithm for scheduling uses two schedulers internally to schedule packets on the output link. The two schedulers are GSQ and SFQ. The GSQ schedules packet from the Guaranteed Service queue and the SFQ schedules packets in the Auxillary service queue. Only one of the two schedulers is in operation at any given time.

- Each packet of a flow that arrives at the regulator is also placed in the auxillary queue. The rate regulator sends packets of flows to the guaranteed service queue according to the contract of the flows.

- If the GSQ is not empty, the GSQ server schedules packets and meets service guarantees. As packets are scheduled on the output line, corresponding packets in the auxillary queue are removed.

- If GSQ is empty as all packets sent by the regulator are already scheduled, but regulator does not output more packets as that will violate the contract with the network. In this case the FA algorithm switches to the SFQ scheduler. The SFQ sceduler has the packets that are arriving at the inputs of the regulators. As long as the GSQ is empty the SFQ schedules packets from the auxillary queue. Each packet that is scheduled by the SFQ is removed from the rate regulator input.

- Every arriving packet is serviced by one of the schedulers and the output line in never idle as long as we have packets to schedule. By using a combination of schedulers and a rate regulator, the FA algorithm guarantees that it will meet the contract and if a flow sends more packets it tries to schedule as many more packets it can. Thus we have a work conserving scheduling algorithm and one which has better utilization than the non-work conserving scheduler.

The GSQ scheduler to be used can be any scheduler that guarantess:

$$L_{GSQ}(p^j) \leq EAT^{RC}(p^j) + \beta^j$$

where:
$L_{GSQ}(p^j)$ is the departure time for packet p^j
$EAT^{RC}(p^j)$ is the expected arrival time of packet p^j
β^j is a contant for a given flow and is independent of other flows

Examples of GSQ scedulers are: Delay EDD, Virtual clock etc.

24.2.3 Delay Guarantees provided by FA algorithms

Assumptions:

- routers along the end-to-end path do no use the same scheduling algorithm.

- scheduling algorithms used belong a class of scheduling algorithms (GR class) as we cannot have an arbitrary combination of algorithms and still provide good delay guarantees.

  A scheduling algorithm belongs to the GR class if it guarantees a deadline for a packet based on the expected arrival time of the packet - referred to as delay guarantee. Large class of algorithms belong to this class. A GR scheduling algorithm guarantees that packet p_f^j will be transmitted by $GRC^i(p_f^j) + \beta^j$
  where:
  β^j is a constant for each flow and
  $GRC^i(p_f^j)$ is the guaranteed rate clock value of packet p_f^j at server i and is independent of the behavior of other flows. This property facilitates end-to-end delay determination.

The objective is to provided bounded end-to-end packet delays and loss service in heterogeneous networks.

24.3 Determining End-to-End Delay Bounds

To determine the end-to-end delay bounds along a path, we analyze a path as a whole assuming that the scheduling algorithm at each server belongs to the GR class.

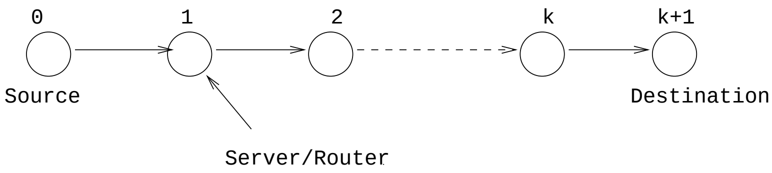

Figure 24.3: Methodology to determine end-to-end delay bound

$$\begin{aligned} d_f^j &= \text{end-to-end delay of packet } p_f^j \\ &= \text{Arrival time at destination - Departure time at source} \end{aligned}$$

End-to-end delay of p_f^j:

$$d_f^j \quad \leq ArrivalTime^{k+1} - ArrivalTime^1$$

$= D^k + \tau^{k,k+1} - A^1$

D^k is departure time at server k and $\tau^{k,k+1}$ is transmission time from server k to $k+1$

$$= GRC^k(p_f^j) + \underbrace{\beta^k + \tau^{k,k+1}}_{\alpha^k} - A^1(p_f^j)$$

$$d_f^j \quad \leq (GRC^k(p_f^j) - A^1(p_f^j)) + \alpha^k$$

An observation from the above inequality, $GRC^k(p_j^i)$ depends on $A^k(p_f^j)$, which in turn depends on $GRC^{k-1}(p_f^j)$ and so on ... till in the end $A^2(p_f^j)$ depends on $GRC^1(p_f^j)$. Thus, if we write $GRC^k(p_f^j)$ in terms of $GRC^1(p_f^j)$, end-to-end delay can be computed by computing only the values of GRC^1 and A^1 for each flow. So the end-to-end delay problem is reduced to a single node problem. A simplified expression using the above observation is:

$$d_f^j \leq (GRC^1(p_f^j) - A^1(p_f^j) + 2 \text{ constants}$$

24.3.1 End-to-end delay bounds: Example

The factor $GRC^1(p_f^j) - A^1(p_f^j)$ is different for different flow characteristics.
For a Leaky bucket constrained flow:

$$AP_f(t_1, t_2) \leq \sigma + \rho(t_2 - t_1)$$

σ : burstiness (maximum allowed burst in any time interval)
ρ : average rate
AP_f : arrival rate of packets of flow f

For the above flow charateristics:

$$GRC^1(p_f^j) - A^1(p_f^j) \leq \frac{\sigma}{\rho}$$

The complete inequality for d_f^j for a end-to-end path with multiple servers(routers) is:

$$d_f^j \leq (GRC^1(p_f^j) - A^1(p_f^j) + (k-1) \times max_{n \in (1 \ldots j)} \frac{l_f^n}{r_f} + \sum_{n=1}^{k} \alpha^n$$

α^n has a term β which is a constant, but is different for different routers as they might be using differetn scheduling algorithms.
l_f^n is the length of the packet n of flow f
r_f is the rate associated with flow f

All the terms in the formula are deterministic and thus end-to-end belay bound for each flow can be specified by computing values for a single node and knowing corresponding constants.