CMPSCI 754 Multimedia Systems	**Spring 2001**

Lecture 21: April 24, 2001

Lecturer: Prashant Shenoy — *Scribe: Jiang Lan*

The first part of this lecture continues the discussion of the packet scheduling algorithms, then the second part resumes the talk of smoothing techniques (slides 221-223).

21.1 Selecting a Packet Scheduling Algorithm

21.1.1 Application Requirements

Integrated services networks are required to support a variety of applications with with a wide range of QoS requirements. For example, continuous media applications require the network to provide QoS guarantees with respect to bandwidth, packet delay and loss; interactive applications such as telnet require low packet delay and loss; throughput intensive applications like ftp require network resources to be allocated such that throughput is maximized. A network meets these requireemnts primarily by appropriately scheduling its resources.

Below is a summary of the requirements of some of the principal applications envisioned for integrated services networks:

- Video
  Variable bit rate (VBR) video sources have unpredictable and highly variable bit rate requirement at multiple time-scales. These imposes the following requirements on network resource managements:
  - Unpredictability implies that video channels may utilize more than its reserved bandwidth. As long as the aditional bandwidth used is not at the expense of other channels (i.e., the channel utilize idle bandwidths), it should not be penalized in the future by reducing its bandwidth allocation.
  - Variabile bit rate requirement suggests that to achieve efficient utilization of resources, a network will have to over-book available bandwidth. The network should provide some QoS guarantees even in the presence of of overload (caused by over-booking).
  - Unfair scheduling algorithms such as Virtual Clock, Delay EDD, etc., penalize channels for the use of idle bandwidth and do not provide any QoS guarantee in the presence of congestion. Hence fair scheduling algorithms are desired for video applications.
- Audio
  To maintain adequate interactivity for audio application such as audio conferencing, scheduling algorithms must provide low average and worst-case packet delay.
- Data
  - Interactive data application (e.g., telnet): need low average packet delay.
  - Throughput intensive application:
    - In heterogeneous, large-scale, decentralized environments: bandwidth must be allocated fairly.
    - Due to the coexistence of VBR video sources and data sources in intergated services networks, the bandwidth available to data applications may vary significantly over time. Consequently, the fairness property of the scheduling must hold regardless of variation in server capacity.

21.1.2 Network Requirements

- Due to the heterogeneity in services and protocols, the scheduling algorithm should facilitate hierarchical link bandwidth allocation. In Figure 21.1, different service classes are arranged in a hierarchical tree structure, each service class is assigned a weight. At each level, link bandwidth is allocated to each service class in proportion to its weight.

  Advantage of this scheme is isolation between different service classes:

  - Different service classes can be designed independently
  - Different scheduling algorithms can be applied to different service classes

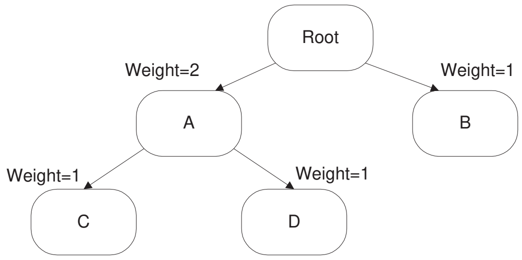

Figure 21.1: Hierarchical link bandwidth allocation among different service classes

- Hierarchical link bandwidth allocation can also be applied to multiple administrative domains (Figure 21.2)

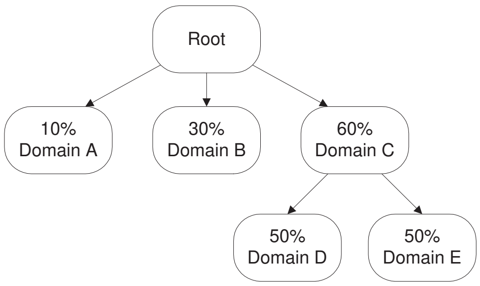

Figure 21.2: Hierarchical link bandwidth allocation among multiple administrative domains

- Finally, to facilitate its implementation in high-speed networks, a scheduling algorithm should be computationally efficient.

21.1.3 Requirements Summary

In summary, a suitable scheduling algorithm for integrated services network should:

1. Achieve low average and worst-case packet delay for low throughput applications
2. Provide fairness for VBR video
3. Provide fairness, regardless of variation in server capacity, for throughput-intensive, flow-based applications
4. Utilize hierarchical link bandwidth allocation

5. Computationally efficient

The objective is to develop a scheduling algorithm that meets all the above requirements.

In previous work, several fair scheduling algorithm has been proposed, next we introduce Weighted Fair Queuing and show its limitation, then we'll propose a new fair scheduling algorithm that meets the above objective.

21.1.4 Weighted Fair Queuing (WFQ)

WFQ was designed to emulate a hypothetical bit-by-bit weighted round robin server in which the number of bits of a flow served in a round is proportional to the weight of the flow. This emulation is done by scheduling packets in the increasing order of their departure times in the hypothetical server. To compute this departure order, WFQ associates two tags, a start tag and a finish tag, with every packet of a flow. Specifically, let p^j and l^j denote the j^{th} packet of a flow and its length, respectively, and if $A(p^j)$ denotes the arrival time of packet p^j at the server, then start tag $S(p^j)$ and finish tag $F(p^j)$ of packet p^j are defined as

$$S(p^j) = max\{v(A(p^j)), F(p^{j-1})\}; j \geq 1$$

$$F(p^j) = S(p^j) + \frac{l^j}{r}$$

where $F(p^0) = 0$ and $v(t)$ is defined as the round number that would be in progress at time t if the packets were being serviced in a bit-by-bit weighted round robin manner. Formally, $v(t)$ is defined as

$$\frac{dv(t)}{dt} = \frac{C}{\sum_{i \in B(t)} r_i}$$

where C is the capacity of the server and $B(t)$ is the set of back-logged flows at time t in the bit-by-bit round robin server.

WFQ then schedules packets in the increasing order of their finish tags.

21.1.4.1 Limitation of WFQ

- Computation of $v(t)$ requires simulation of bit-by-bit round robin server in real time, and is therefore computationally expensive.

- WFQ is based on the assumption that the server capacity is constant, it is unfair over variable rate servers. Intuitively, this is because $v(t)$ depends on the server capacity C, flutuation in C affects the finish tag and deviates the bit-by-bit simulation. An example can be easily constructed to illustrate the unfairness of WFQ over variable rate servers.

21.1.4.2 Limitation of Existing Algorithms

In order for hierarchical link allocation to be useful, a scheduling algorithm must provide fairness over variable rate servers. Algorithms like WFQ fails to provide fairness over variable rate server, thus are unsuitable for integrated services networks. Other existing algorithms either are unfair over variable rate servers or have unsuitable delay properties.

21.1.4.3 Some Key Observations

1. Algorithms should be defined independent of server capacity to achieve fairness over variable rate servers
2. Schedule packets in start tag order to provide low delay to low throughput flows. In WFQ, finish tag is inversely proportional to weight
$$F(p^j) = S(p^j) + \frac{l^j}{r}$$
scheduling in finish tag order thus provides low delay to high throughput flows. Intuitively, doing the contrary may give us a better algorithm. In the next section, we introduce Start-time Fair Queuing (SFQ) algorithm [1] which is based on this idea and meets the requirements of an integrated services network.

21.1.5 Start-time Fair Queuing (SFQ)

Like WFQ, in SFQ each packet has a start tag $S(p^j)$ and a finish tag $F(p^j)$

$$S(p^j) = max\{v(A(p^j)), F(p^{j-1})\}; j \geq 1$$

$$F(p^j) = S(p^j) + \frac{l^j}{r}$$

$A(p^j)$: arrival time of packet

$v(t)$: virtual time

r : weight of the flow

Unlike WFQ, packets in SFQ are scheduled in the increasing order of the start tags. Furthermore, $v(t)$ is defined as the start tag of the packet in service at time t, thus the computation of $v(t)$ is much more simplified in SFQ compared to WFQ and can be done efficiently.

It can be proven that SFQ provides fairness over variable rate servers. Intuitively this is because the virtual time equals the start tag of the packet in service at time t, thus won't be affected when server capacity fluctuates. An example could be easily constructed to illustate the fairness.

In short, SFQ is proven to meet all the proposed requirements for integrated services networks.

Next the lecture goes back to slides 221-223.

21.2 Congestion Control and Admission Control Protocols for Stored Video

21.2.1 Congestion Control Protocol for Stored Video

Compressed vedio streams are not resilient to packet loss. In stored video, the sender has full knowledge of the characteristics of the video stream prior to transmission, thus it can exploit predictablity of the bit rate requirement to completely avoid any packet loss in the network. In the proposed protocol [2], the source sends bit rate requirement to the network during the connection setup, then the network sends feedback rate back to the source to avoid congestion. Upon receiving the network feedback, the source will send packets at the feedback rate. This protocol migrates the functionality of discarding packets in the event of congestion to the source, which understand the semantics of data thus can ensure minimum degradation of video quality during congestion.

21.2.2 Multimedia Servers: Review

Due to the periodic nature of the video playback, the server can service retrieval requests from clients by proceeding in fixed duration *rounds*. In a round, a server retrieves a fixed number of frames, sufficient to sustain playback for the duration of the round for each client. The data retrieved by the server in one round are then transmitted to clients over the network in the next round. This technique is called double buffering.

Since the end-to-end delay requirements for stored video are not very stringent, a server can make the rate changes infrequent, regular, and predictable by transmitting the video information accessed during a round at a constant rate during the next round. This allows the server to easily characterize the variation in the bit rate requirements of a video stream which can then be utilized by network admission control algorithm.

21.2.3 Admission Control

A histogram-based traffic characterization is computed at the time of storage. Because in stored video, packet loss is a more important factor compared to end-to-end delay. The proposed admission control algorithm will base on the packet loss probability.

The algorithm:

Consider a network switch that is multiplexing N video sources originating from distributed multimedia servers onto a link of capacity C (bits/sec). Let random variable χ_i denote the bit rate requirement of the i^{th} video source, and let q_i denote the maximum acceptable loss probability. Without loss of generality, we assume $\forall j \in [i..N] : q_i \leq q_j$. Let us define the QoS requirement as priority, and associate this priority to packet loss probability. Specifically, the higher the packet loss probablity, the lower is the priority, i.e., its QoS requirement. Essentially, the sources are ordered in decreasing order of their QoS requirement. Then the admission control algorithm must ensure that at each QoS level, the probability of cumulative bandwidth of sources, belonging to QoS levels equal to or higher than that level, exceeding the link capacity is within the tolerable packet loss at that level. I.e., the following must hold:

$$\forall j \in [i, N] : P\left(\sum_{i=1}^{i=j} \chi_i > C\right) \leq q_j$$

Z-transform or CLT can be utilized to evaluate the above inequalities.

References

[1] P. GOYAL,H.M. VIN, and H. CHENG, Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks, *In Proceedings of ACM SIGCOMM'96, San Francisco*, Pages 157-168, August 1996

[2] P. GOYAL and H.M. VIN, Network Algorithms and Protocol for Multimedia Servers, *In Proceedings of INFOCOM'96, San Francisco*, Pages 1371-1379, March 1996