A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks

by

Abhay K. Parekh
Laboratory for Information Decision Systems
MIT

1992

- Communication and computation speeds are continuing to increase dramatically.
- Integrated Services networks that can support multimedia applications over wide areas are becoming realizable.
  - –Video On Demand –Scalable Video
  - –Digital Libraries –Parallel Processing
  - –Virtual Reality –Scientific Visualization

Caution: Buzz words abound!!

What are the architectural issues?

Integrated Services Networks will offer services that:

- span a wide range of bit rates (1-10^9 bits/sec)
- span a wide range of message lengths
- may have to offer delay guarantees to each message (voice and video)

As more transmission bandwidth becomes available so will the demand for high bandwidth services.

Traditional networks trade off flexibility with performance guarantees. Need **both** in integrated services networks.

Primary goal of the thesis: To investigate the extent to which these apparently conflicting requirements can be reconciled in a packet switched network, when the short-term demand for link bandwidth frequently exceeds link capacity.

Difficulties—Questions of Modelling

- What do the sources look like?
  - Complicated or Unknown Stochastic Processes!
- How do we use queueing theory to provide *hard* performance guarantees?

Issues—Questions of Strategy

- Is it necessary to manage packet delay at each hop of the route?
  - Tobagi and Peha, Ferrari.
- Does the server have to be non-work conserving?
  - Golestani.
- Is it necessary to perform rate enforcement at each hop of the route?
  - Zhang.

Approach taken:

1. Use Leaky Buckets admission control to upper bound the the inflow of traffic into the network for all intervals of time. This allows us to talk meaningfully about *worst-case* delay. Inspired by Turner, Cruz.

   Packet Delay = Delay in Bucket + Delay in Network

2. Assume a fluid model of traffic; use a non-packet based, multiplexing discipline at the nodes, GPS, that is

   - flexible
   - efficient
   - *analyzable.*

3. Devise a practical packet-based service discipline, PGPS, that closely approximates GPS.

   Demers, Shenker and Keshav.

Major Contributions

A framework for flow control within which

- worst-case performance guarantees can be given on packet delay and backlog to a wide range of co-existing session types (CBR and VBR).
- these bounds can be computed efficiently for arbitrary topology networks.

Our results have been used by Clark, Shenker and Zhang in a recent proposal to support real-time traffic.

Demonstration of the fact that it is not necessary for the service discipline to be deadline-based or non-work conserving.

Price to pay:

- small delay $\Rightarrow$ small packets
- high flexibility $\Rightarrow$ more computational overhead at session set-up time

We have also developed a set of analytical tools and ideas that can be used to understand the behavior of other service disciplines.

Stages in the Research

1. Single Server GPS
2. Single Server PGPS
3. Arbitrary Topology GPS Networks
4. Arbitrary Topology PGPS Networks

A Generalized Processor Sharing (GPS) server is work conserving and operates at a fixed rate r.
It is characterized by positive $\phi_1, \phi_2, ..., \phi_N$.
Let $S_i(\tau, t)$ be the amount of session i traffic served in an interval $[\tau, t]$. Then a GPS server is defined as one for which

$$\frac{S_i(\tau, t)}{S_j(\tau, t)} \geq \frac{\phi_i}{\phi_j}, \quad j = 1, 2, ..., N$$

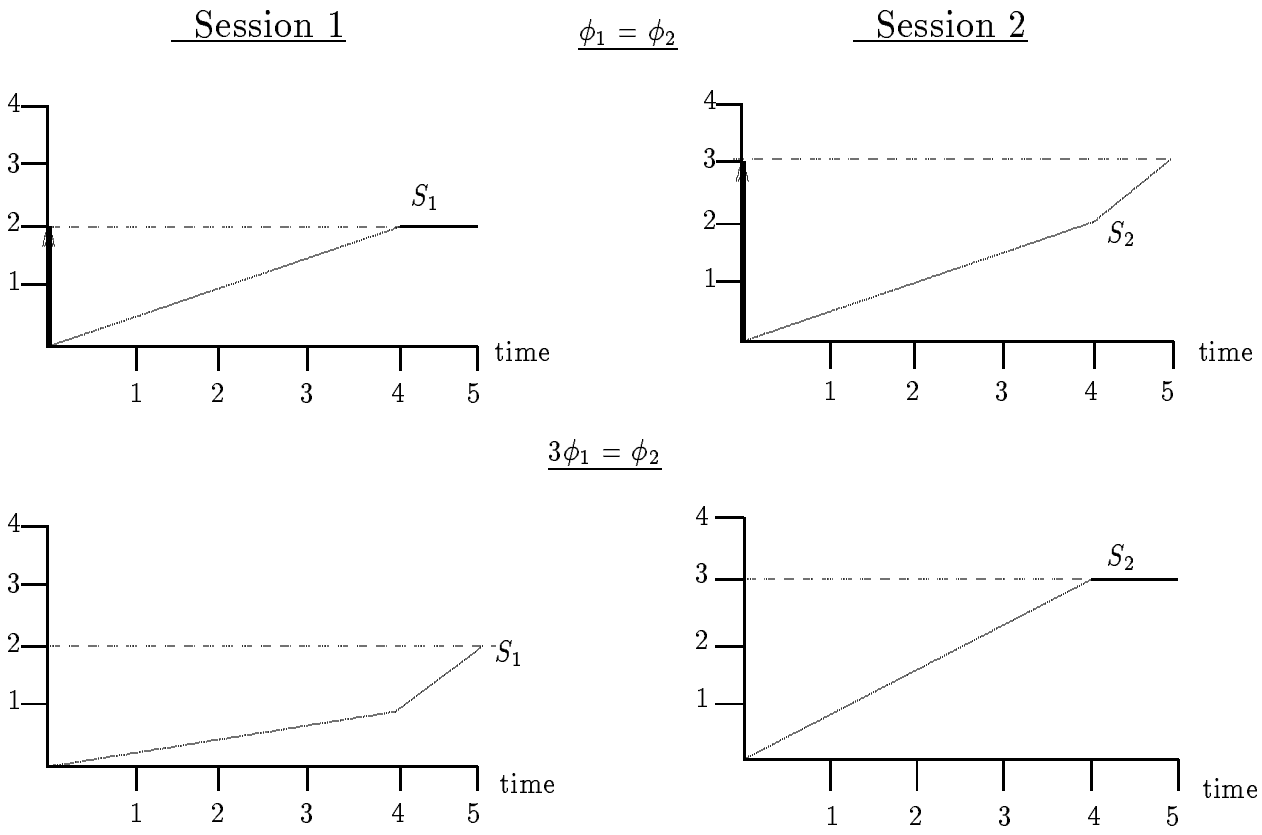
for any session i that is backlogged in the interval $[\tau, t]$.

Session i is guaranteed a backlog clearing rate (BCR) of

$$g_i = \frac{\phi_i}{\Sigma_j \, \phi_j} r.$$

If a session i arriving bit sees a session backlog of q, then it will be served in at most $\frac{q}{g_i}$ time units.
If $\hat{g}_i$ $(\forall i)$ is desired guarantee then can set $\phi_i = \hat{g}_i$ as long as $\Sigma \, g_i < r$.
Also, if combined average rate is constrained to be less than r, then every assignment of the ϕ's results in a stable system.

A Simple Example of GPS



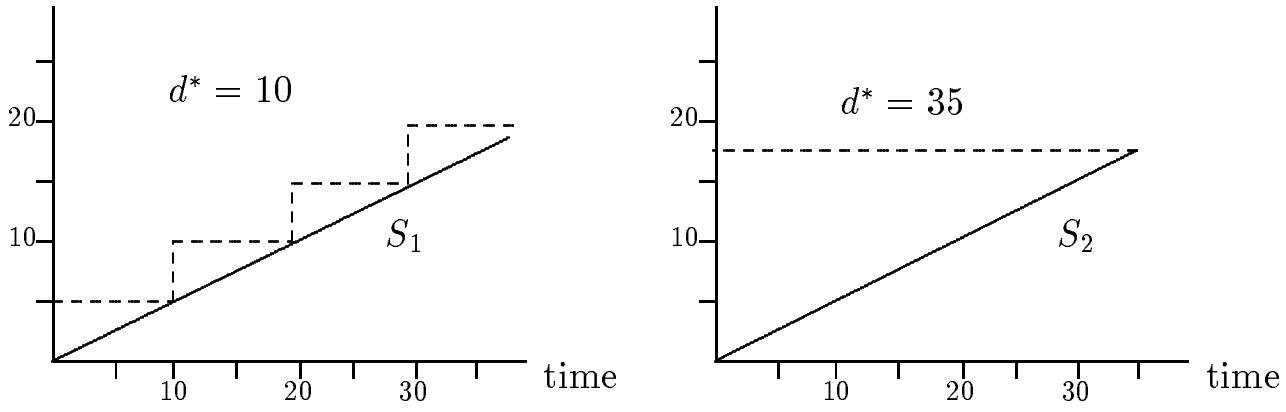
Higher $\phi_i \Rightarrow$ lower session i delay at the the expense of the other sessions.

Increasing ϕ for Steady Sessions

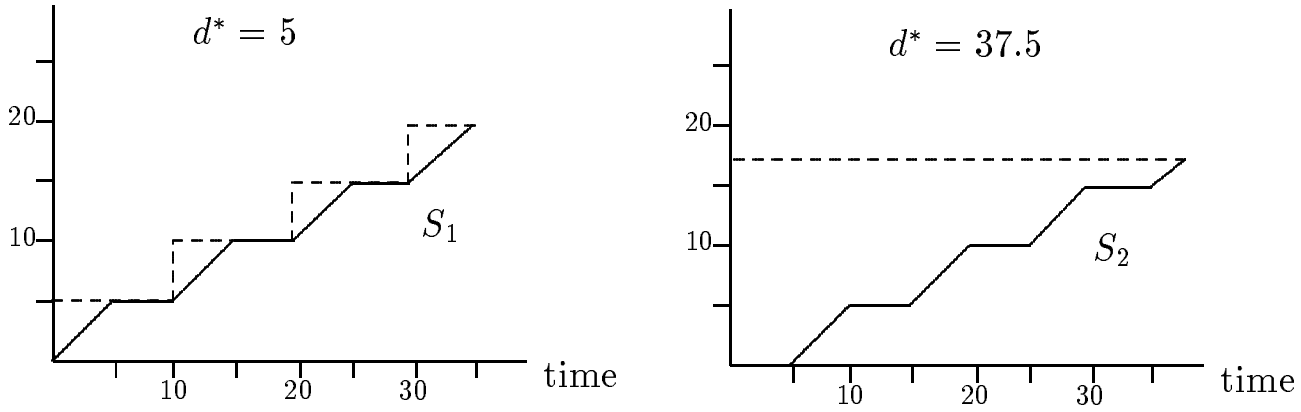
A very steady session, such as a video session, may also be extremely delay sensitive. Can improve its performance by increasing its value of ϕ_i, with minimal performance degradation to other sessions.

In the example, ϕ_1 is increased to infinity, but session 2 delay goes up by only 2.5 time units.

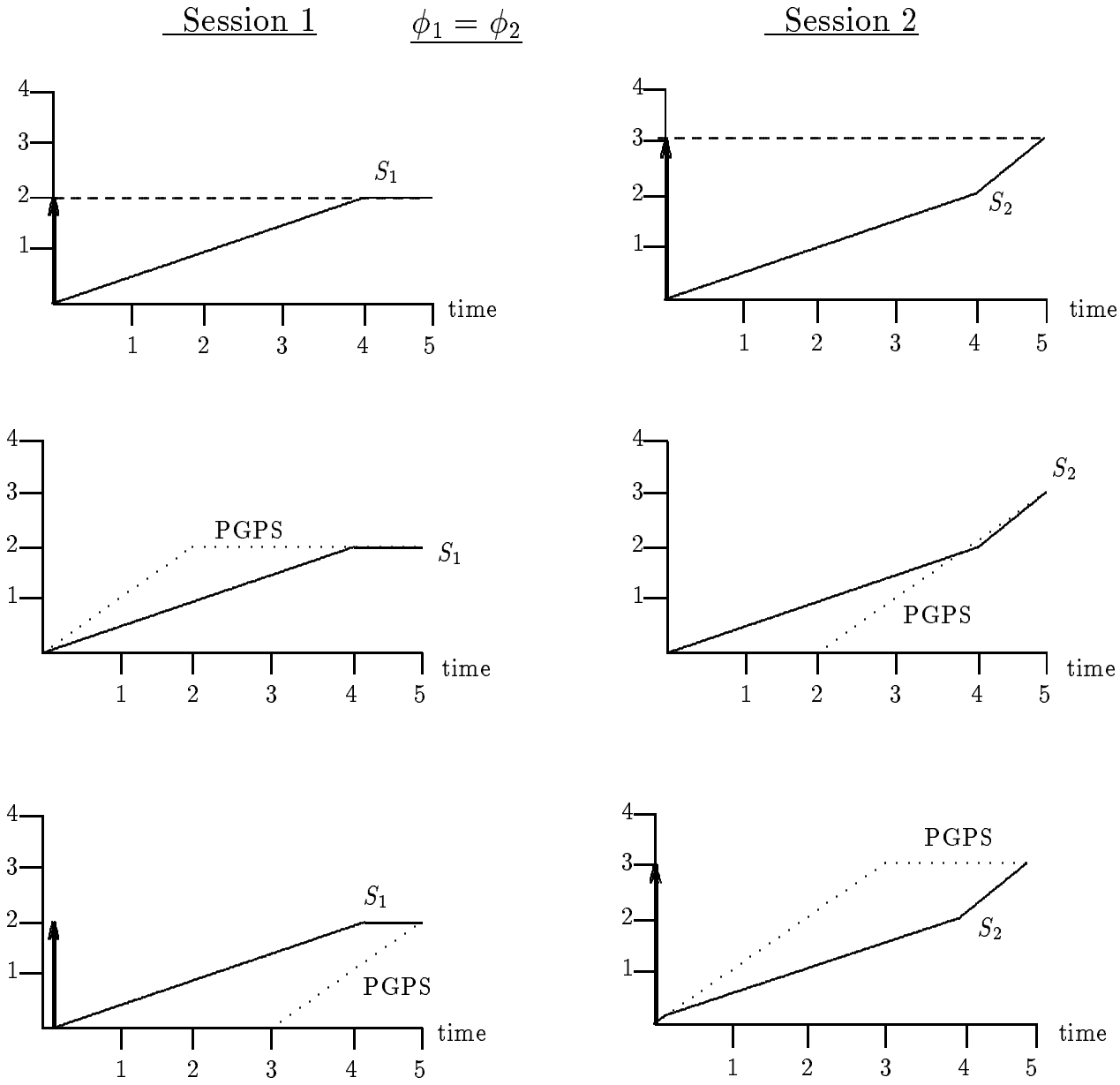
$\phi_1 = \phi_2$



$\phi_1 >>> \phi_2$



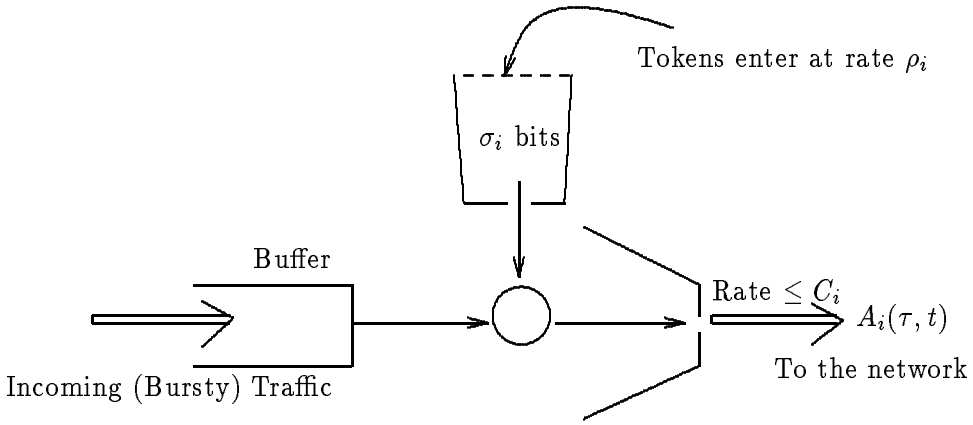
Packet-by-Packet GPS (PGPS) attempts to schedule the packets in the same order as they would depart the system under GPS.



Most of the time packets get out faster under PGPS. Also,

$$\hat{F}_p - F_p < \frac{L_{max}}{r}, \quad \text{for all} \quad p.$$

Admission Control—Leaky Buckets (Turner)



The traffic entering the network is characterized by

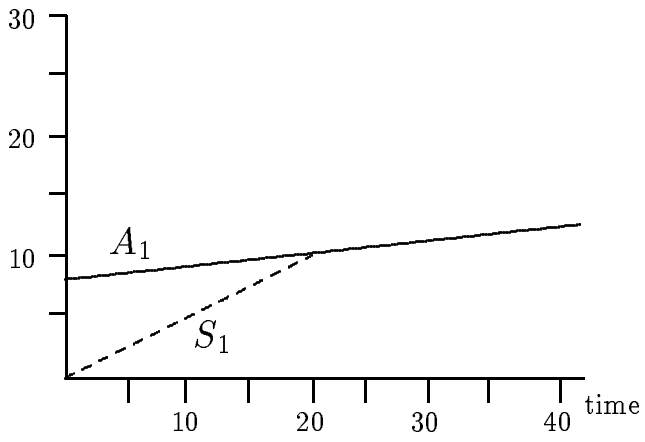
$$A_i(\tau, t) \leq \min\{(t-\tau)C_i, \sigma_i + \rho_i(t-\tau)\}, \quad \forall t \geq \tau \geq 0,$$

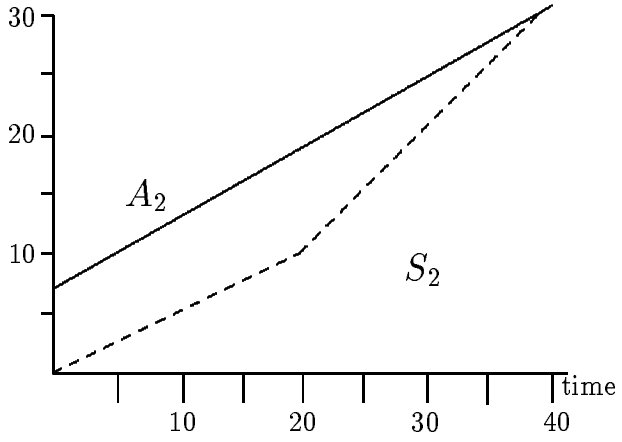
for every session i (Cruz, Kumar).

Session i conforms to (σ_i, ρ_i, C_i), or $A_i \sim (\sigma_i, \rho_i, C_i)$.

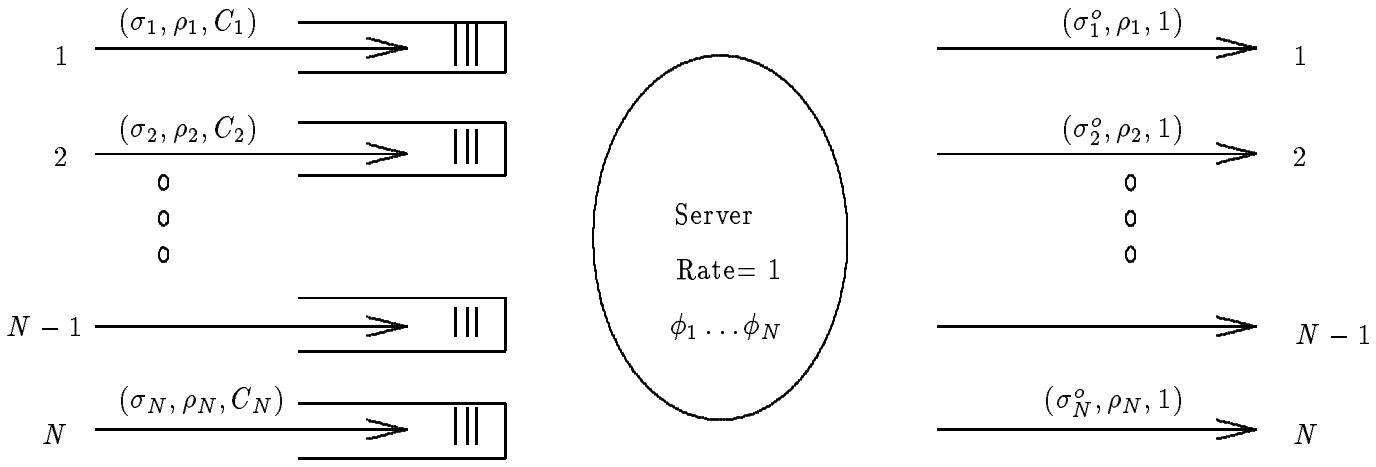
Packet Delay = Delay in Bucket + Delay in Network

$A_1 \sim (8, .1, \infty)$ $\quad$ $\underline{\phi_1 = \phi_2}$ $\quad$ $A_2 \sim (6, .6, \infty)$





A Single Server System



Incoming Traffic

Outgoing Traffic

$$\sum_{i=1}^{N} \rho_i < 1$$

$$C_i \geq \rho_i$$

$$\sigma_i^o \geq \sigma_i \qquad i = 1, ..., N$$

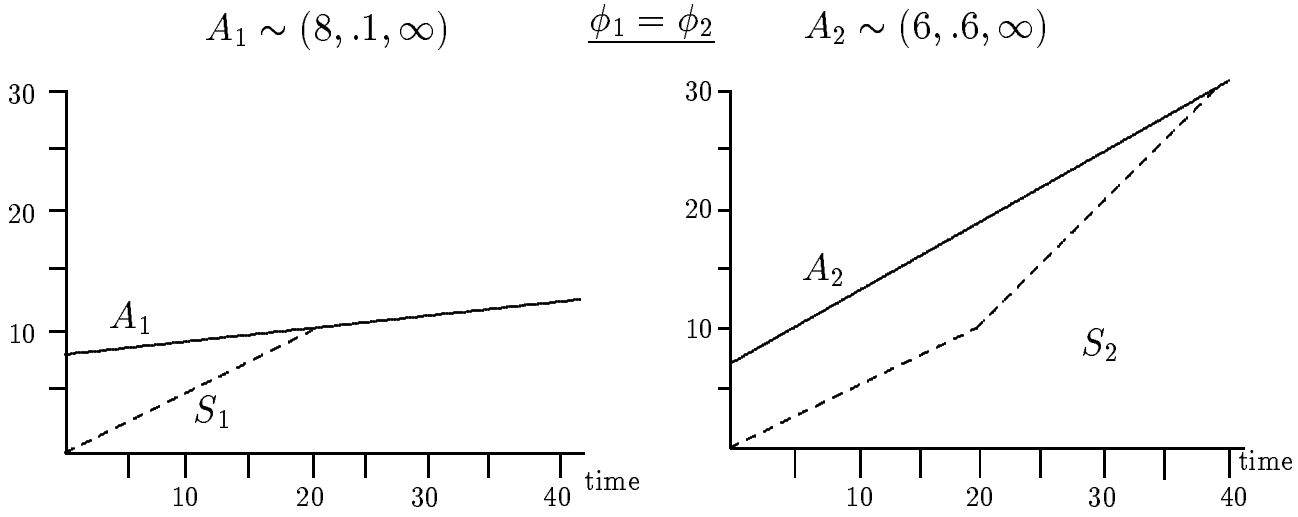
$A_i \sim (\sigma_i, \rho_i, C_i)$ for every session i.

D_i^* is the worst case delay for session i bits.

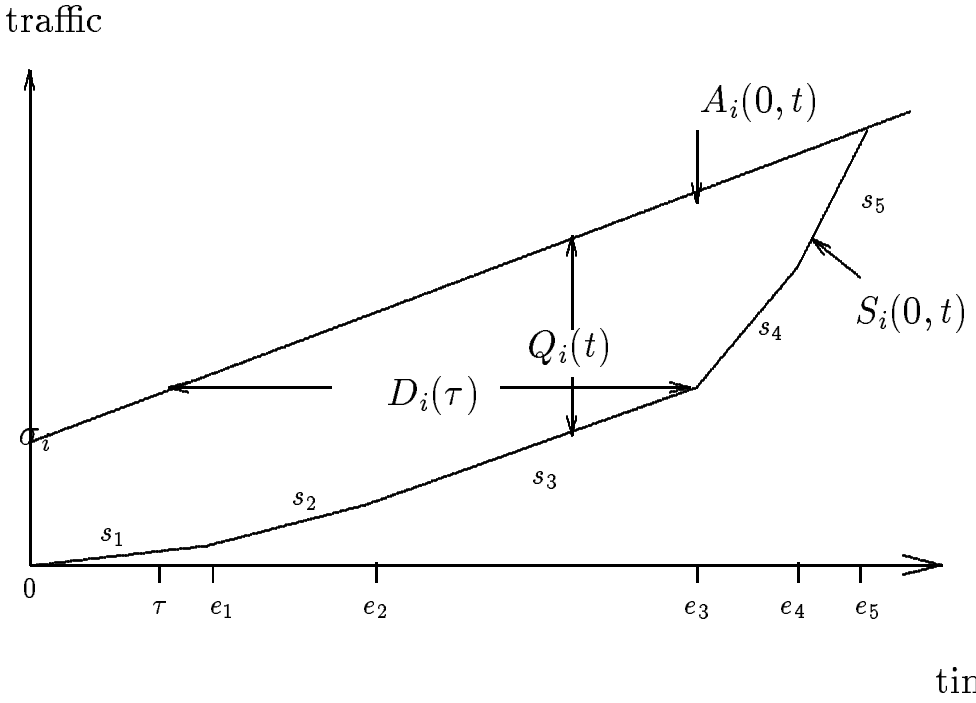
Q_i^* is the worst case session i backlog.

A session is greedy from time τ if it sends traffic as fast as it can from time τ—i.e. it attempts to exhaust all of its tokens from time τ on.
It turns out that for every session i, D_i^* and Q_i^* are achieved when all the sessions are greedy starting at time zero.

Also, $\sigma_i^o = \max\{\sigma_i, Q_i^*\}$.



Session i behavior for equal ϕ's and infinite C's



In an all greedy system, once the server has cleared a session's backlog, the session is never backlogged again. Suppose that these backlogs are cleared in the order $1, 2, ..., N$:

Then the slopes of S_i are easily determined:

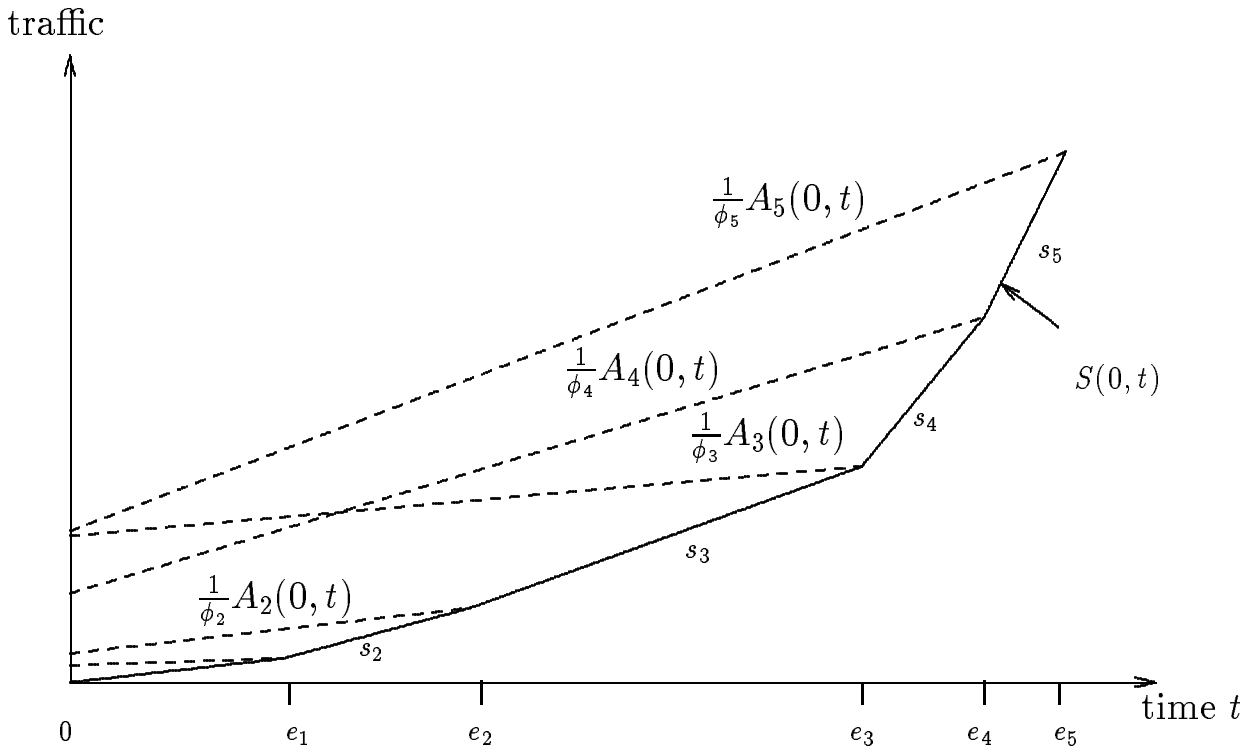
$$s_1 = \frac{1}{N}$$

$$s_2 = \frac{1-\rho_1}{N-1}$$

$$s_k = \frac{1-\sum_{i=1}^{k-1}\rho_i}{N-k+1} \quad k = 1, 2, ..., N.$$

Also, $\rho_k < s_k$ for $k = 1, 2, ..., N$.

A Universal Service Curve for General ϕ



The worst case delay D_i^* can be computed directly for each session.

The worst case backlog Q_i^* can be computed by scaling distances by ϕ_i.

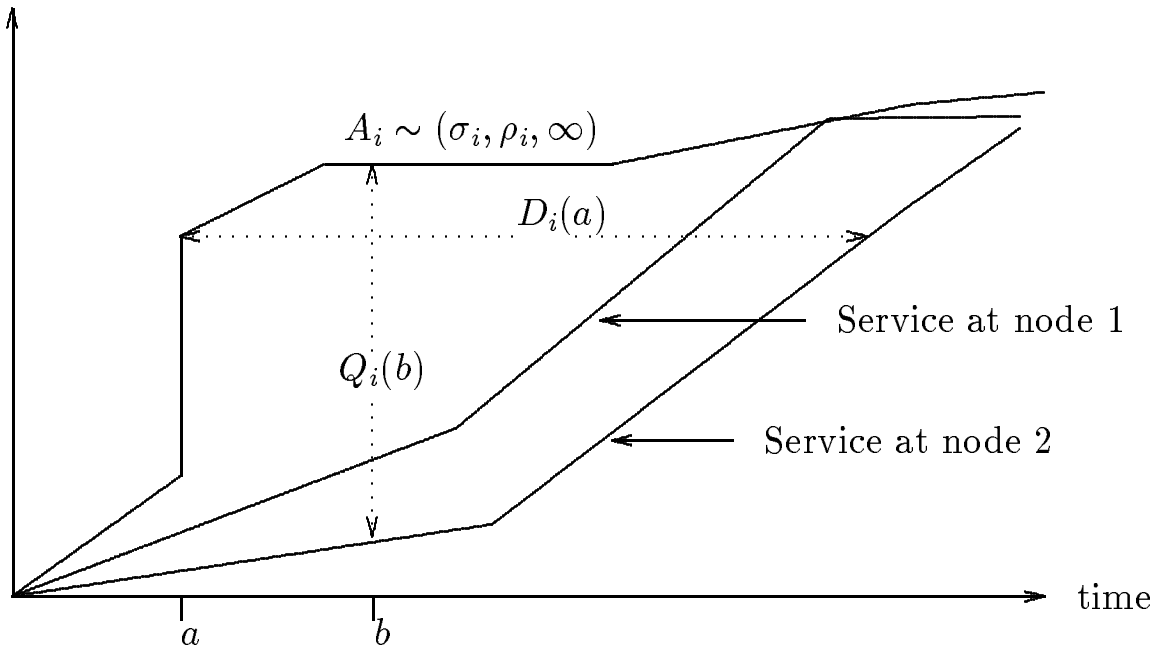
The output traffic, i.e. σ_i^o can be computed from Q_i^*.

Networks of GPS servers

$A_i \sim (\sigma_i, \rho_i, C_i)$ for every session i.

D_i^* is the worst case delay for session i packets.

Q_i^* is the worst case session i backlog.



We can compute bounds on D_i^* and Q_i^* for every session i under a broad class of allocations:

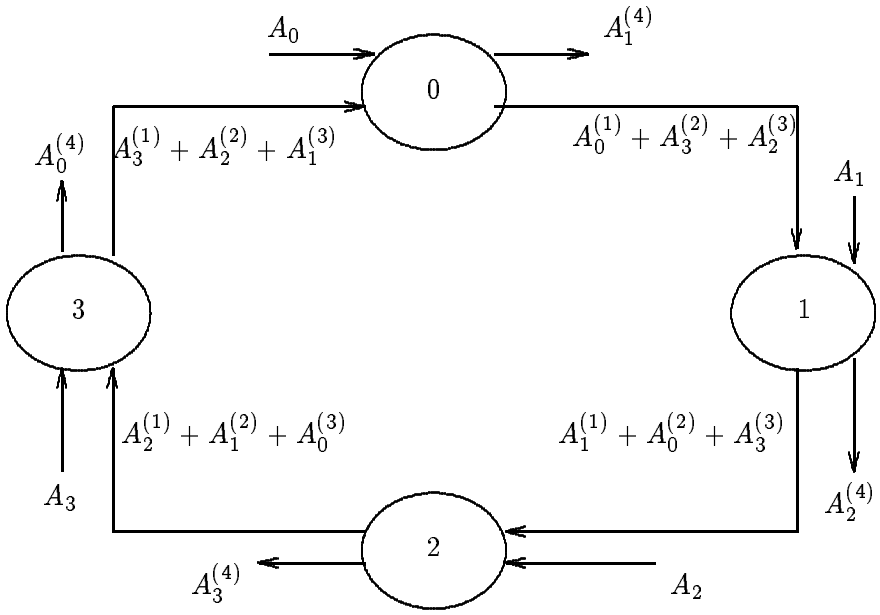
Our analysis treats the session i route as a whole—rather than to add up worst-case delays at each node of the session i route.

Estimating D_i^* and Q_i^*

Two steps:

- Estimate internal network traffic in terms of σ's ρ's and C's. If session i uses node k then $A_i^k \sim (\sigma_i^k, \rho_i, C_i^k)$.
- Analyze the session i route.

The Problem with Cycles



$A_0^{(1)}$ depends on $A_2^{(2)}$, which depends on $A_0^{(3)}$
$\Rightarrow$ Virtual Feedback! This is quite difficult to deal with for general service disciplines.

- When through traffic is given priority, Cruz provides a bound that implies instability when link utilizations are $\approx \frac{1}{2}$.
- For GPS, we show stability for link utilizations < 1, when the GPS server parameter assignments lie in a broad class called Consistent Relative Session Treatment Assignments.

Consistent Relative Session Treatment Assignments

Session j is said to impede a session i, at a node m if

$$\frac{\phi_i^m}{\phi_j^m} < \frac{\rho_i}{\rho_j}.$$

A Consistent Relative Session Treatment GPS assignment (CRST) is one for which there exists a strict ordering of the sessions such that for any two sessions i, j, if session i is less than session j in the ordering, then session i does not impede session j at any node of the network.

The class of assignments that yield CRST is quite broad:

Uniform Relative Session Treatment Assignment:

$$\frac{\phi_i^m}{\phi_j^m} = \frac{\phi_i}{\phi_j} \quad \forall m \text{ s.t. } i, j \in I(m).$$

Alternatively: For every session i, and node m that is on the session i route: $\phi_i^m = \phi_i$.

Rate Proportional Assignment: $\phi_i^m = \rho_i$ for every session i.

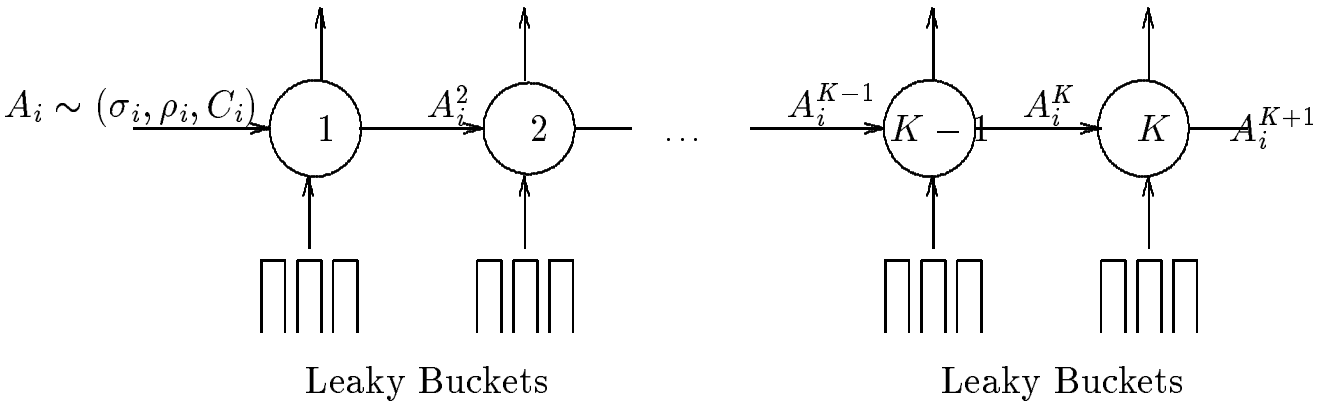
All CRST systems are stable. σ_i^m's can be efficiently and accurately computed.

The Independent Sessions Assumption

Suppose the session i route is $1, 2, ..., K$:

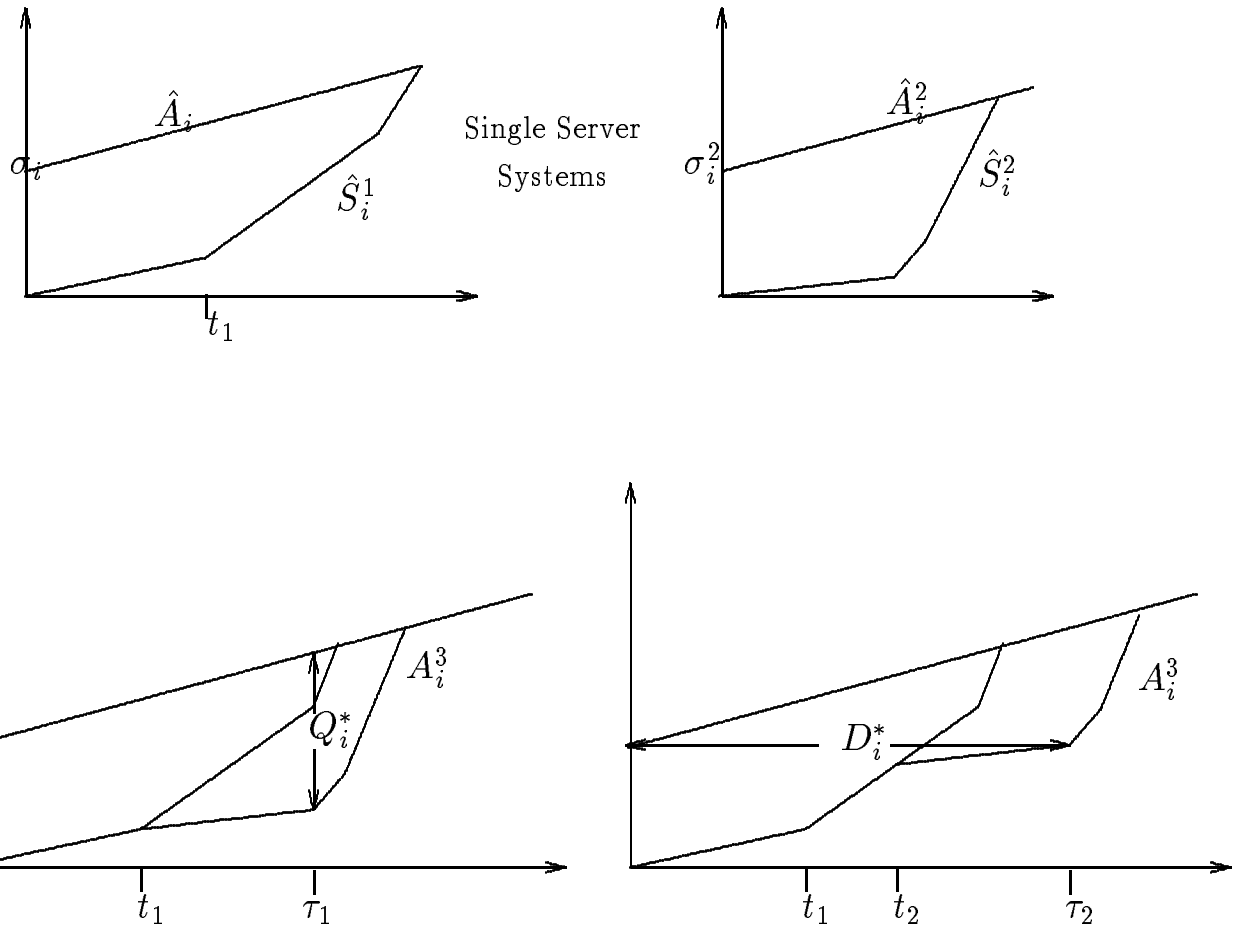
A session $j \neq i$ at any of these nodes is called an independent session.

We assume that each independent session at a node k is free to send traffic in any manner it chooses as long as $A_j^k \sim (\sigma_j^k, \rho_j, C_j^k)$.
Performance under assumption upper bounds true WC performance.



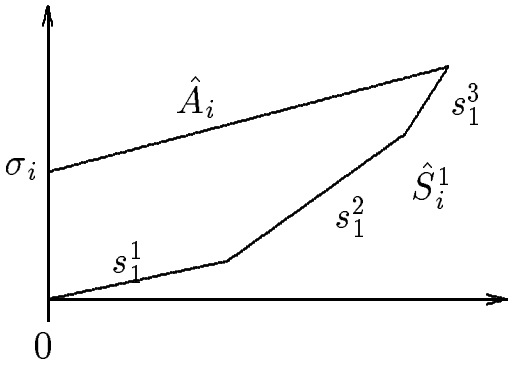
We can analyze the session i route exactly given the independent sessions assumption.

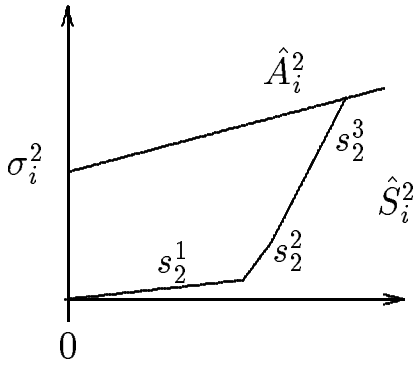
The independent sessions at a node k become simultaneously greedy, but only after the independent sessions at node $k-1$ have simultaneously become greedy. Thus, the worst case behavior follows a staggered greedy pattern.

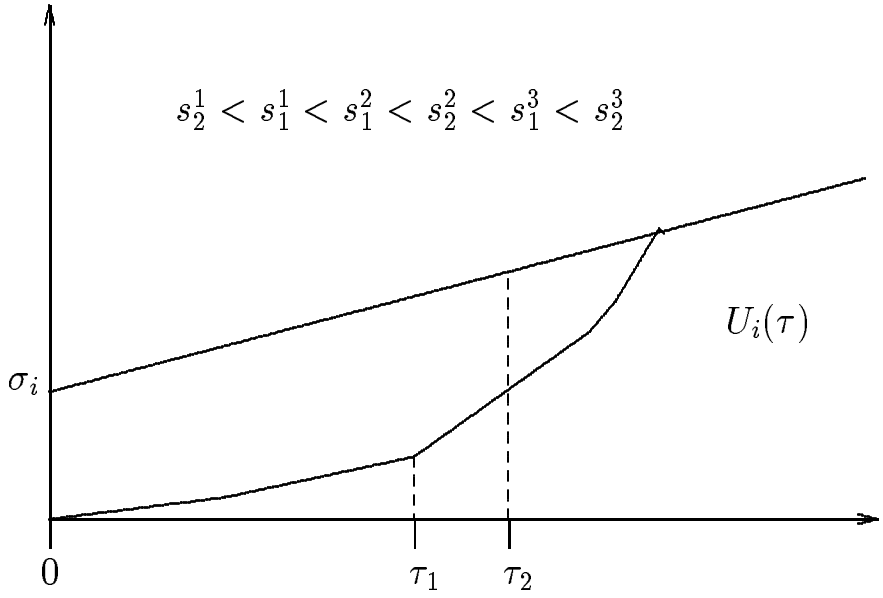


Different regimes maximize backlog and delay!

The Universal Service Curve for Session i







Now "read off" D_i^* and Q_i^*!!

Special Assignments

Bounds on D_i^* and Q_i^* can be easily calculated efficiently for a wide range of allocations (eg all CRST allocations). However closed form solutions are messy.

Special cases: Define (the min BCR along a route)

$$\hat{g}_i = \min_{m \in P(i)} \frac{\phi_i^m r^m}{\Sigma_{j \in I(m)} \phi_j^m}.$$

If $\hat{g}_i \geq \rho_i$ for session i:

$$D_i^* \leq \frac{\sigma_i}{\hat{g}_i}.$$

Bound is independent of the topology of the network and number of links in the route taken by the session. Also, it is independent of the σ_j, $j \neq i$. Other sessions do not even have to be leaky bucket constrained.

Now if $\phi_i = \rho_i$ for each session i then Rate Proportional Processor Sharing

$$D_i^* \leq \frac{\sigma_i}{\rho_i}, \quad Q_i^* \leq \sigma_i.$$

PGPS Networks–A Special Case

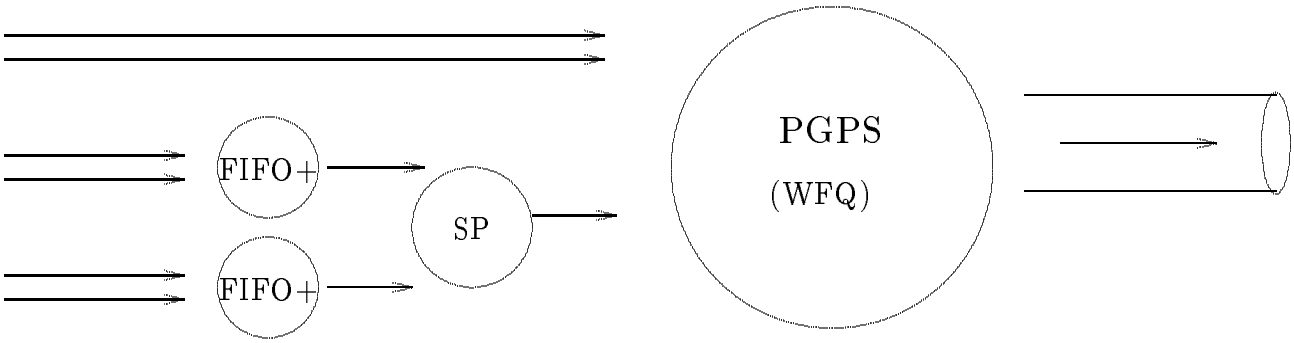
Suppose the BCR, g_i, for session i is greater than ρ_i at each node in the path. Then

$$D_i^{*,\mathrm{PGPS}} \leq \frac{\sigma_i + (K-1)L_{\max}}{g_i} + \sum_{m=1}^{K} \frac{L_{\max}}{r^m}.$$

Under Rate Proportional Processor Sharing, this holds for each session i.

Can incorporate other service classes:

Unified Scheme of Clark, Shenker and Zhang



Sessions 1 and 2 carry guaranteed traffic: $\phi_i = \rho_i$.

Also, $\phi_{3'} < r - \phi_1 - \phi_2$

Main Conclusions

We provided a framework for flow control in which

- worst-case performance guarantees can be given on packet delay and backlog simultaneously to a wide range of session types (CBR and VBR).
- network buffer requirements can be estimated.

Demonstrated that it is not necessary for the service discipline to be "deadline-based" or non-work conserving.

Price to pay:

- small packets;
- computational overhead at session set-up time;
- bounds may be overly conservative if the network is very flexible (many CRST classes);

We have also developed a set of analytical tools and ideas that can be used to understand the behavior of many other service disciplines.

Extensions and Further Work

- Call Admission/Connection Management
- Algorithmic issues.
- Simulation of PGPS to check how tight the bounds are for "typical" network traffic.
- PGPS applied to time-sharing systems.
- A stochastic analysis of GPS — not reversible.
- Determining stability of non-acyclic networks for arbitrary schemes—some connection with work in manufacturing networks by Kumar et al.