

A Flow Control Strategy for Connectionless Traffic over an ATM Network

Lorne G. Mason

INRS-Telecommunications

University of Quebec, Nuns Island, Quebec H3E 1H6

e-mail: lgmason@telusplanet.net

Felisa J. Vázquez-Abad

Department of Computer Science and Operations Research

University of Montreal, Montreal, Quebec H3C 3J7

e-mail: vazquez@iro.umontreal.ca

and

Bernard Kamté

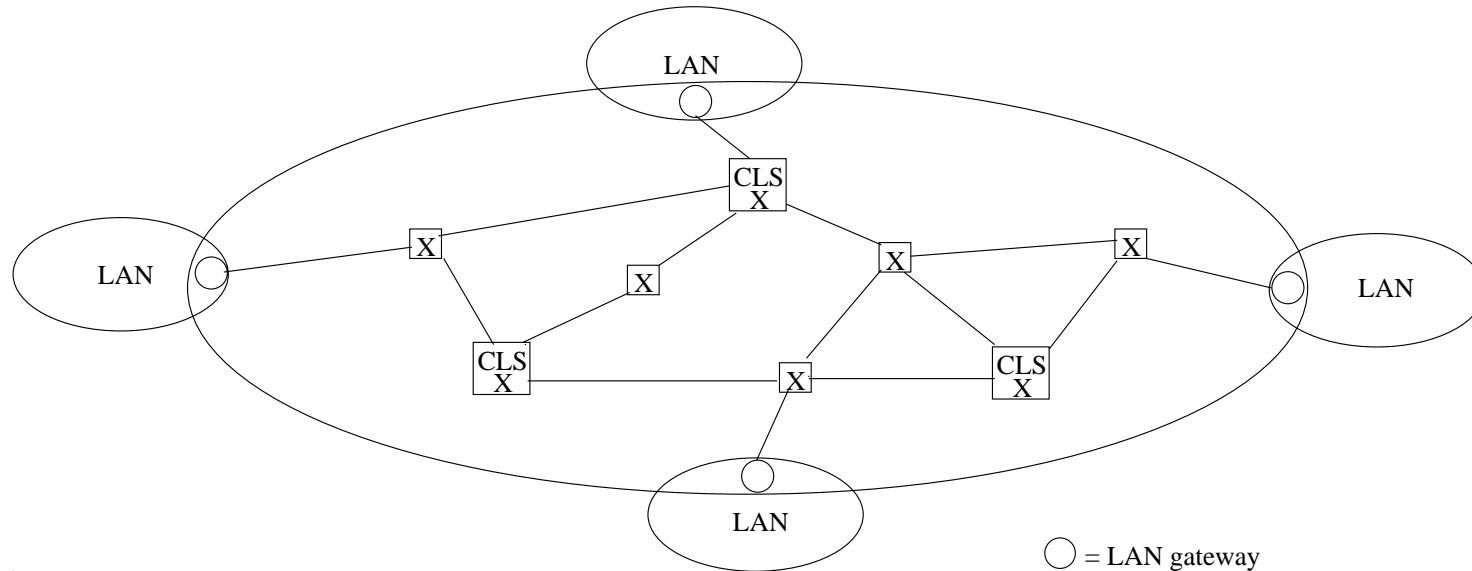
INRS-Telecommunications

University of Quebec, Nuns Island, Quebec H3E 1H6

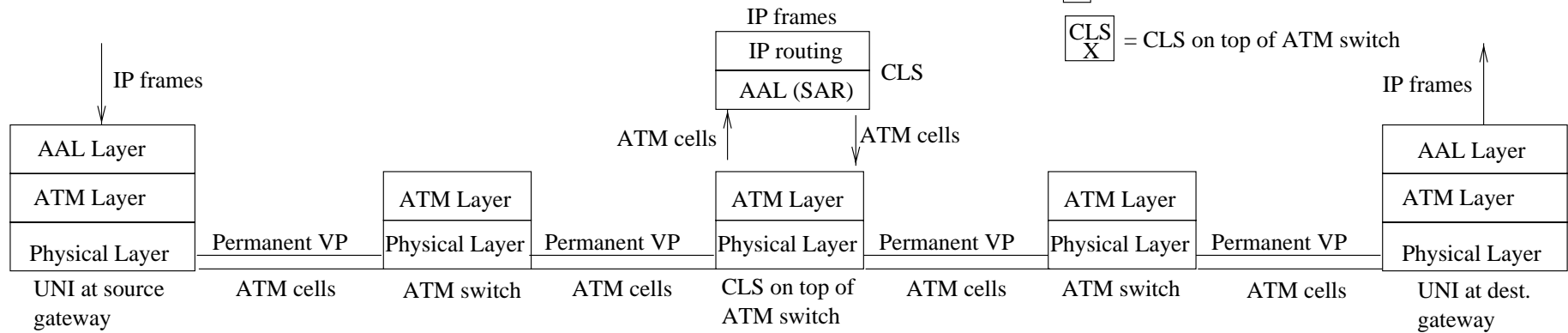
e-mail: kampte@inrs-telecom.quebec.ca

Annual Research Conference *Canadian Institute for Communications Research*, Montebello, Quebec, August 1998.

Network Architecture: Large Scale Network with Connectionless Servers



○ = LAN gateway
 ◻X = ATM switch
 ◻
CLS
X = CLS on top of ATM switch



- At the gateway of the source LAN, IP frames are segmented into cells.
- CLS: IP frames are reassembled (AAL5) for IP routing and further (re)segmented for transmission.
- The CLS network is a virtual network on top of the ATM network.

CLS: Packet Forwarding (AAL5) and Cell Forwarding (AAL3/4)

Connectionless Server: Located at the Gateways, the CLS are responsible for segmentation and routing of IP-frames within the ATM Network, called “Forwarding”.

We shall consider two different Forwarding Techniques: packet forwarding, which uses the AAL5, and cell forwarding, which uses AAL3/4.

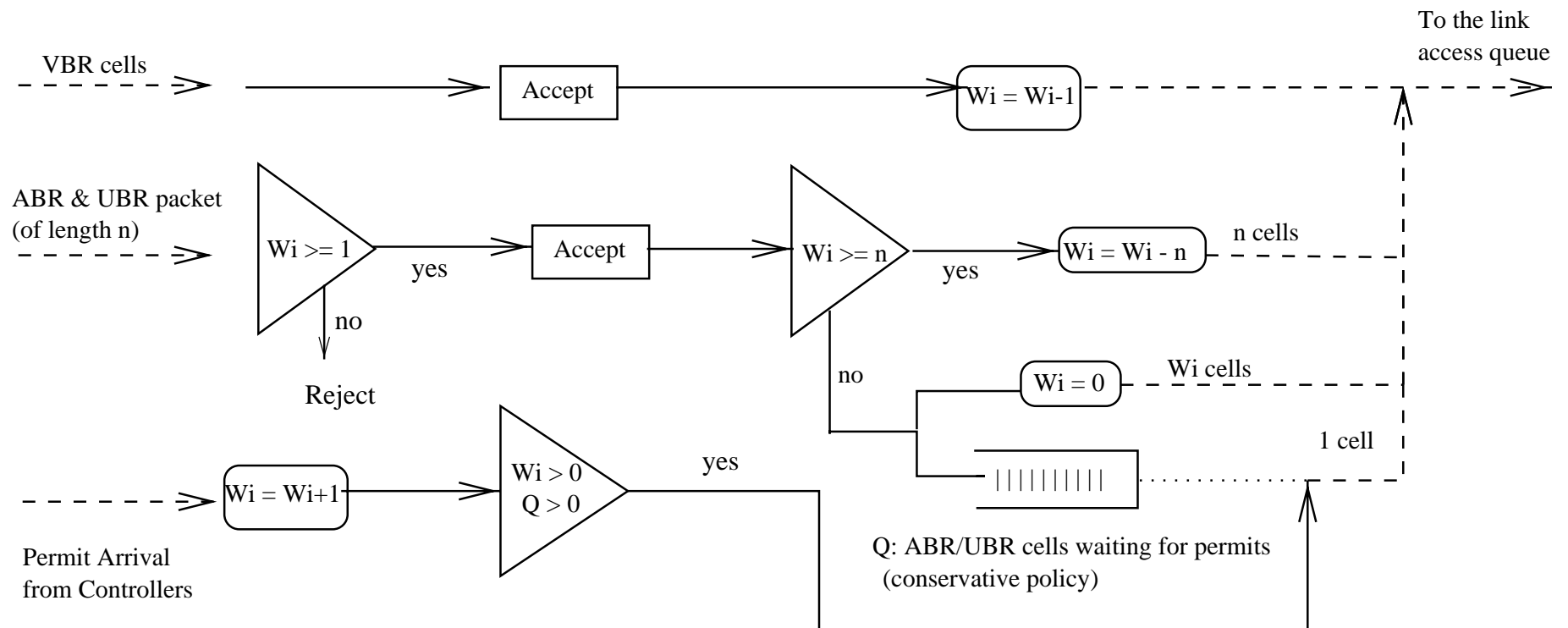
AAL5

- IP-frames are reassembled at the CLS, routed and (re)-segmented into cells for transmission by the ATM network.
- The packet ID is not required: only the first and the last cells of the IP-frame need to be known.
- Since packet ID is not required, there is more room for information in cells from IP-frames.
- Since segmentation and re-assembly are required, delays are introduced for IP-frames.

AAL3/4

- Each cell has a packet ID identifying the IP-frame to which it belongs.
- Reassembly and segmentation are not necessary at CLS, since cells are routed according to:
 - Identification of the IP-frame to which they belong
 - If first cell from a given IP-frame, a route is chosen by the CLS and kept in memory.
 - Subsequent cells from the same IP-frame are identified and routed as the first cell.
- Since packet ID is required, there is less room for information in cells from IP-frames.
- Since cells are routed immediately at the CLS without waiting for other cells from the same IP-frame, end-to-end delays of IP-frames may be reduced.

Network Access Structure: Packet Selective Window (PSW)

Picture Depicting LAN Gateway at node i .

W_i : number of virtual permits at node i . Since real-time traffic is always accepted, W_i can be negative.

Conservative: Upon arrival of a ABR/UBR packet with n cells, if $0 < W_i < n$, then W_i cells are routed towards the link access queue at that node and $n - W_i$ wait in queue for permits.

Aggressive: Upon arrival of a ABR/UBR packet with n cells, if $W_i > 0$, all cells are accepted and routed towards the link access queue at that node. There is no queue to wait for permits, and $W_i = W_i - n$.

Link Access Structure: VBR cells routed towards a link are placed at a finite capacity buffer with high priority, others are placed at an infinite buffer queue.

Network Performance

Packet Classes:

- VBR: Real-time video, voice, MMPP model, highest priority
- ABR: Connection oriented data, Poisson process
- UBR: Internet sources: connectionless data, Self Similar Process

We assume that VBR cells are subject to some access control, the traffic arriving in our model is already the accepted one. Thus, only IP frames and ABR data traffic are subject to our packet selective window. For a given *window size* \mathbf{W} , the *Product of Powers* of class c is given by:

$$P(\mathbf{W}) = \prod_{(o,d)} \frac{\lambda_{o,d}^c}{T_{o,d}^c}$$

$\lambda_{o,d}^c$: Effective class c cell throughput from origin o to destination d

$T_{o,d}^c$: End-to-end delay of class c cells with origin o and destination d .

This performance measure considers the optimization compromise between reducing delays and increasing throughputs. Maximizing performance then gives a Pareto equilibrium solution. We consider the optimization problem of **Maximizing the Product of Powers of UBR cells.**

Implementation Issues:

- **Realistic Traffic Models:** Internet traffic shows *self similar* behaviour: the arrival process has statistical properties that are very different from those of Poisson processes. We compare here the performance under Poisson and Self Similar UBR sources. We use *sgen*, a generator developed by Pedro Iván Sánchez.
- **Scalability:** End-to-end delay and throughputs are global quantities that must be estimated along several nodes in the network. Growth in network size may therefore produce a geometric increase in computational effort required for accurate estimation. We study a hierarchical structure that addresses this problem.

Self-Similar Traffic

Call X_n the number of packet arrivals during the n -th unit of time. If the *arrival process* is a Poisson process then $\{X_n\}$ are i.i.d. variables with Poisson distribution. In this case, the *aggregated process*:

$$X_n^{(m)} = \frac{X_{nm+1} + \dots + X_{(n+1)m}}{m}$$

has also i.i.d. components and satisfies $v^{(m)} = \text{Var}(X_1^{(m)}) = m^{-1}\text{Var}(X_1)$.

Arrival processes where the correlation between X_n and X_{n+l} decreases fast as l increases are called *short range dependent* and also satisfy $v^{(m)} \approx m^{-1}$ for large m .

Internet traffic, however, does not seem to adjust to this behaviour. Due to *long range correlations*, the variance decreases slower.

Definition: A stationary process $\{X_n\}$ is a *Second Order Self Similar* process with self-similarity (or Hurst) parameter H if the process $\{Y^{(m)}\}$ defined as:

$$Y_n^{(m,H)} = \frac{X_{nm+1} + \dots + X_{(n+1)m}}{m^H}$$

has the same finite dimensional distribution as $\{X_n\}$. In particular, it satisfies $v^{(m)} = m^{-2H-2}\text{Var}(X_1)$.

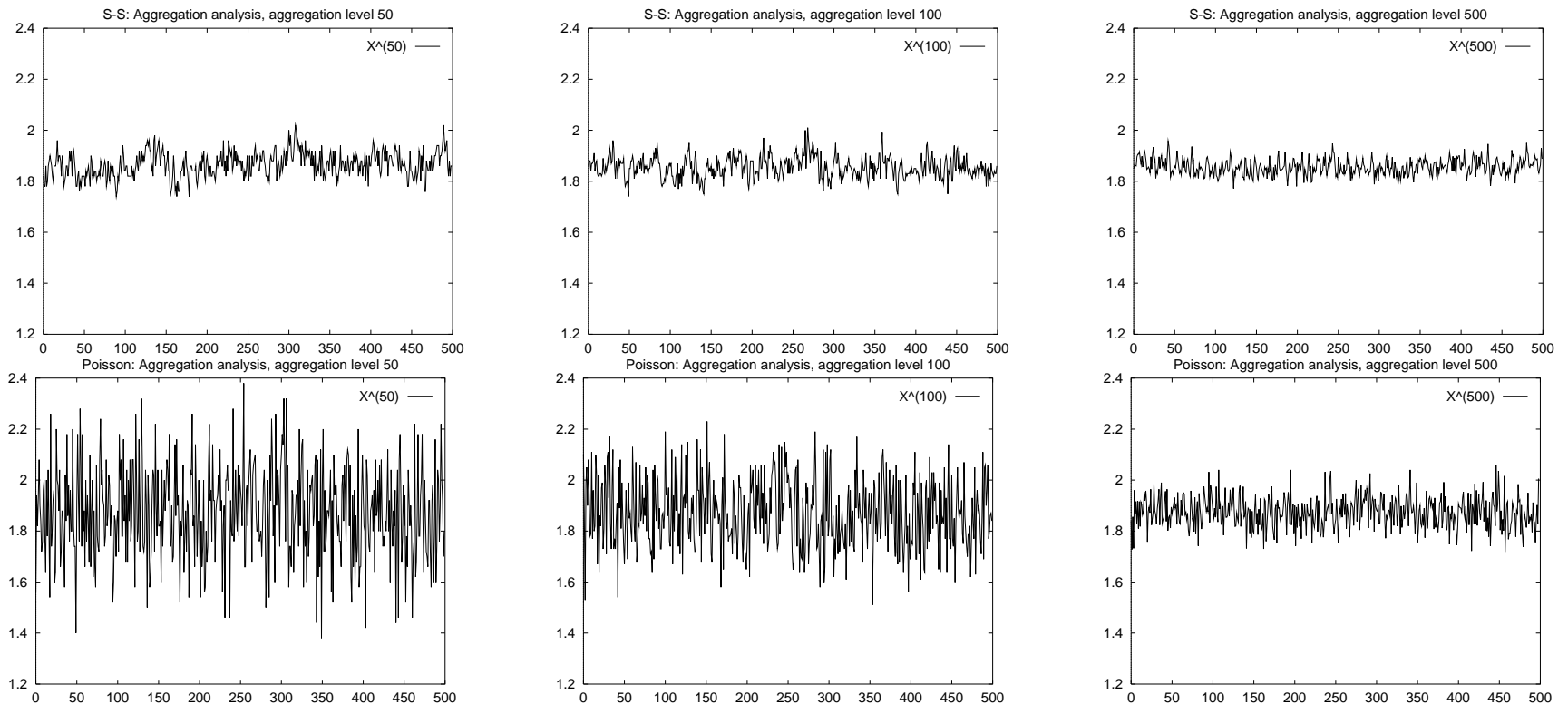
Internet traffic is not actually a stationary self similar process, but statistical tests from real data support the hypothesis that it is *asymptotically second order self-similar* with a Hurst parameter of $H \approx 0.8$, namely that:

$$\forall l > 0, Y_n^{(m,H)} \stackrel{\mathcal{L}}{=} Y_n^{(m+l,H)}, \text{ and } v^{(m)} = m^{-2H-2}\sigma, \text{ for large } m.$$

Remark: Poisson arrivals are asymptotically self-similar processes with $H = 0.5$. Larger values of H yield more “variability” or longer dependence, since $v^{(m)}$ decreases more slowly. Since for $H \approx 1$, the aggregate process $\{X_n^{(m)}\} \stackrel{\mathcal{L}}{=} \{X_n\}$, in this case averaging does not decrease the variance and we cannot use the standard Central Limit Theorem to estimate means.

Aggregate Process Visual Test: The plots show the processes $\{X_n^{(m)}\}$ with increasing aggregation level m , on top for our Self-Similar data, and bottom for a Poisson process with the same mean value.

- For Poisson arrivals the variability decreases noticeably as the aggregation level m grows,
- The Self Similar traffic source presents a slower decreasing variance, with almost identical behaviour of averaged processes, regardless of the level m of aggregation.
- The visual test therefore supports the conjecture that $H > 0.5$ for our UBR generated data.

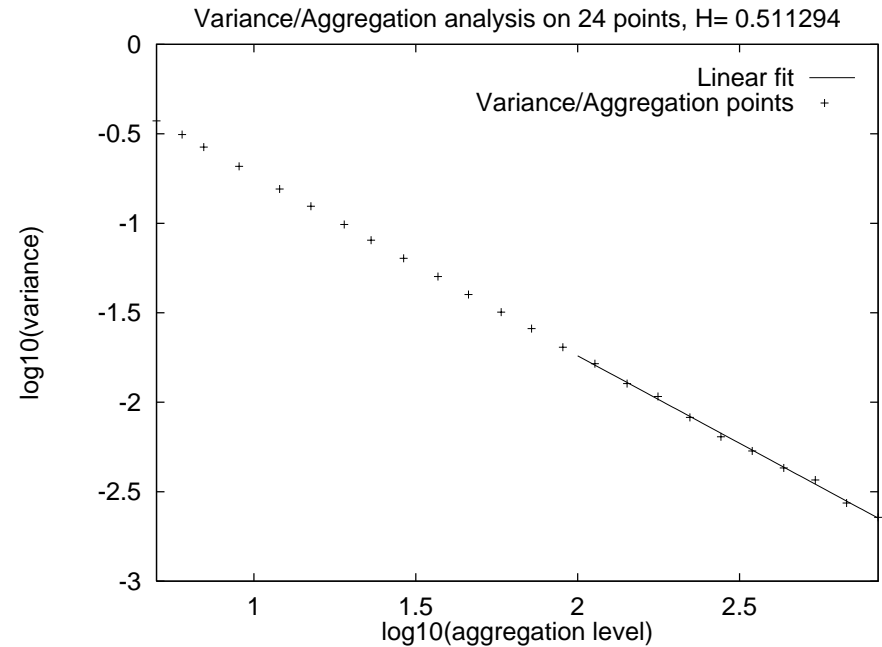
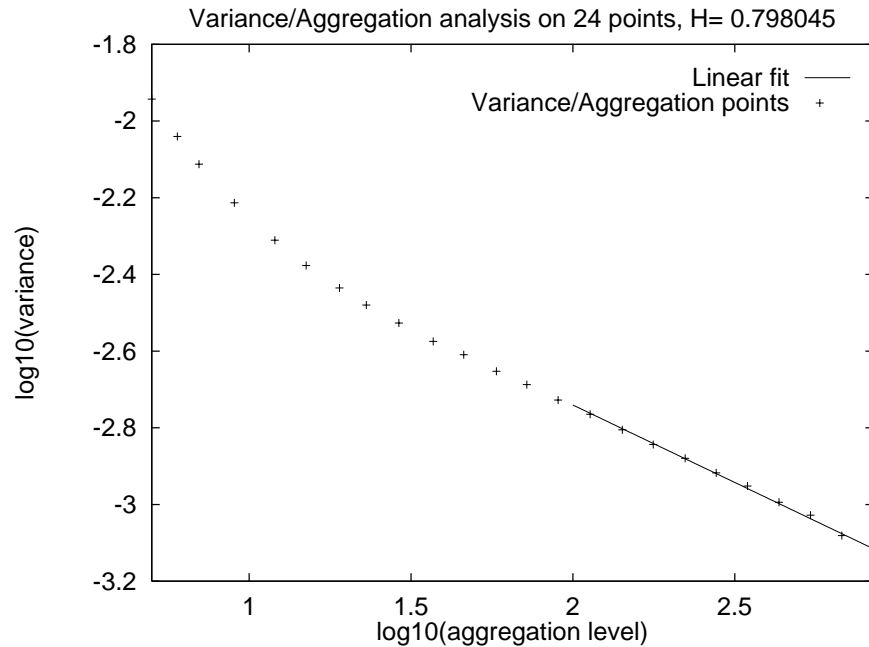


Estimation of H : The variance aggregation test uses the data $\{X_n\}$ to estimate $v^{(m)}$ for increasing m . For a sample of $N = 250000$ values, for example, $v^{(50)}$ is estimated using 5000 batches of size 50 each.

For self-similar processes

$$\log[v^{(m)}] = C + (2H - 2) \log(m), \quad \text{for large } m.$$

The package *sanal* uses the least squares method to fit a straight line through the corresponding points from our data file. Below we show the results for our self-similar generated data and to the right, for the Poisson process.

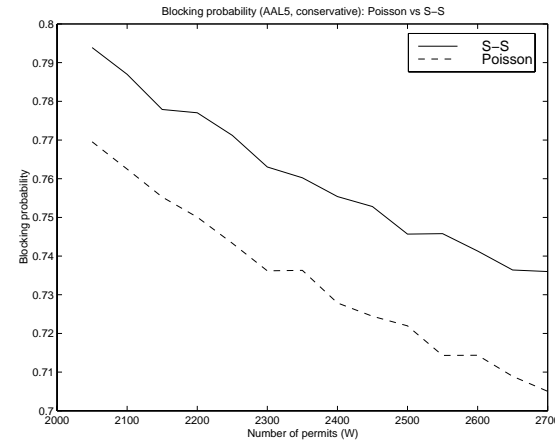
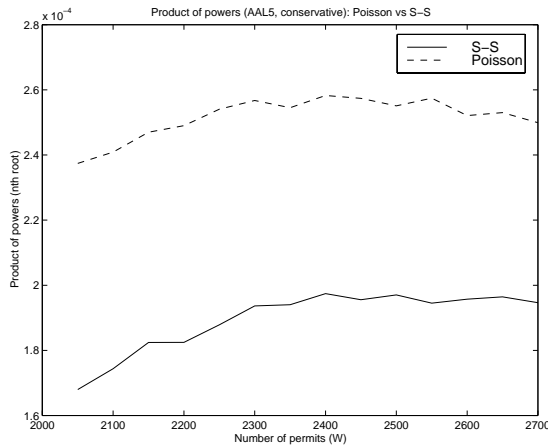


Estimated Hurst Parameter:

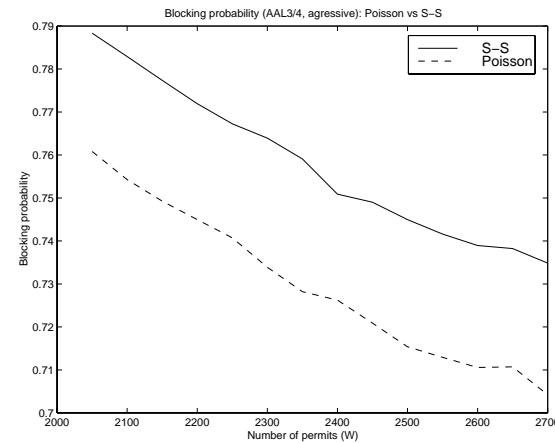
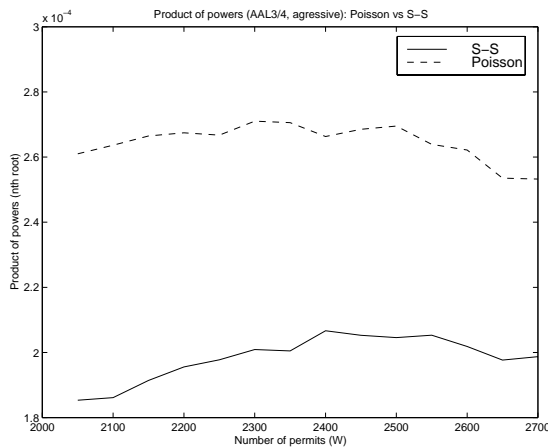
- $\hat{H} = 0.798$ for Self-Similar Traffic

- $\hat{H} = 0.511$ for Poisson Traffic

Comparison, Model 1: Poisson vs Self-Similar



AAL5 under conservative PSW. Product of Powers (left), Blocking Probability (right)

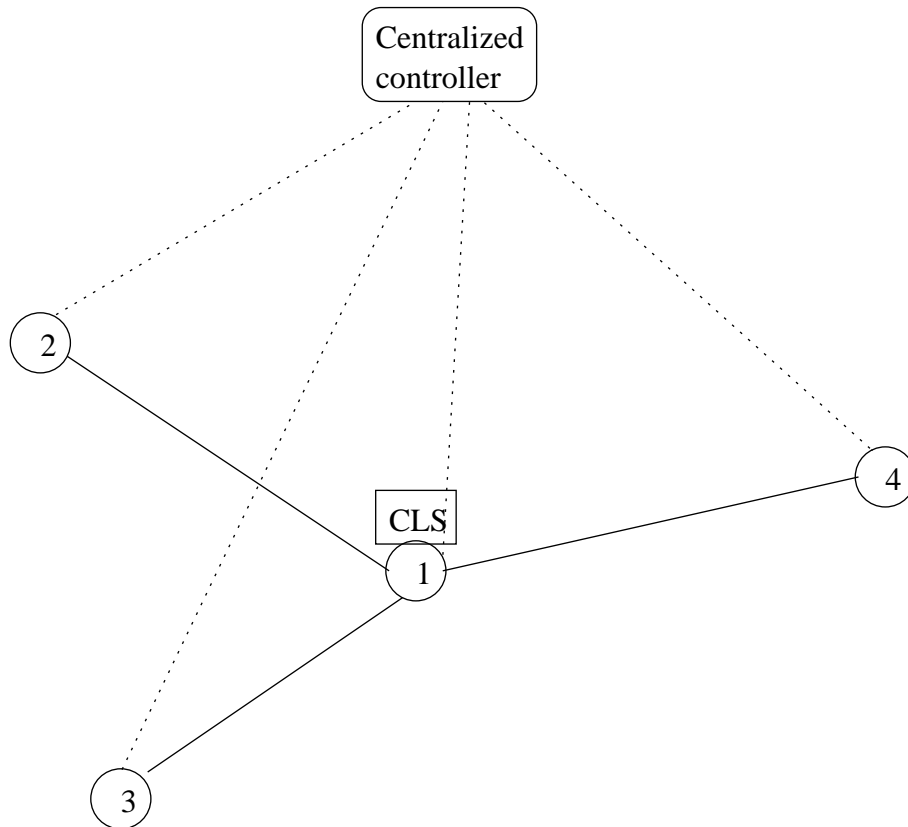


AAL3/4 under aggressive PSW. Product of Powers (left), Blocking Probability (right)

Remarks: Behaviour is similar, but performance is worst for Self-Similar traffic.

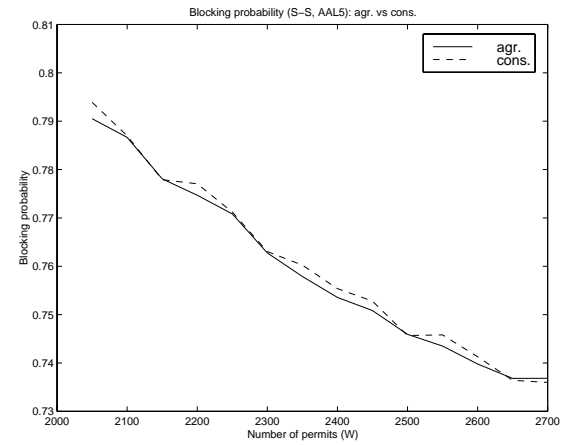
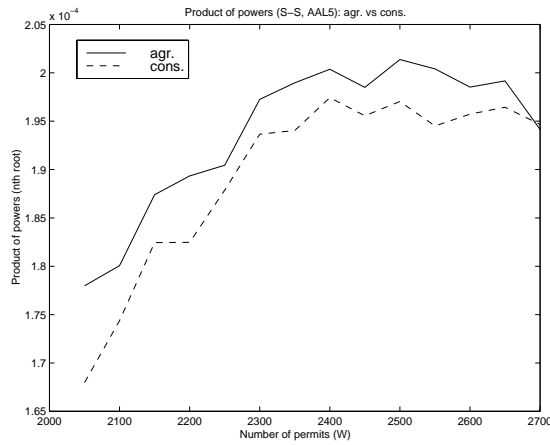
Simulated Network, Model 1

The central controller is in charge of distributing the permits for access control (PWS). All trunks have the same length 17.7 Kms as well as the same capacity, of 150 Mbs. Therefore, the cell transmission time is $\delta = 3.37 \times 10^{-7}$ secs, and the propagation delay is 6×10^{-5} seconds. All incoming packets have a geometrically distributed number of cells.

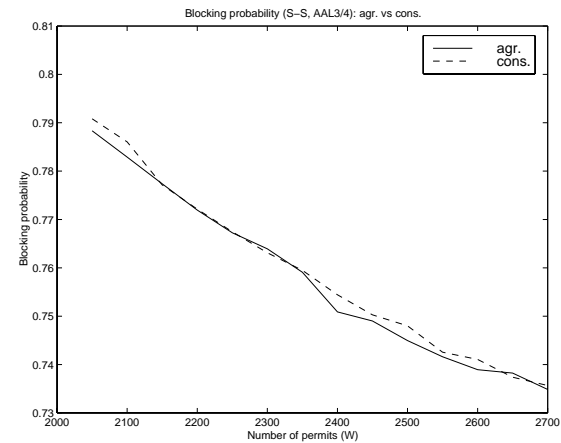
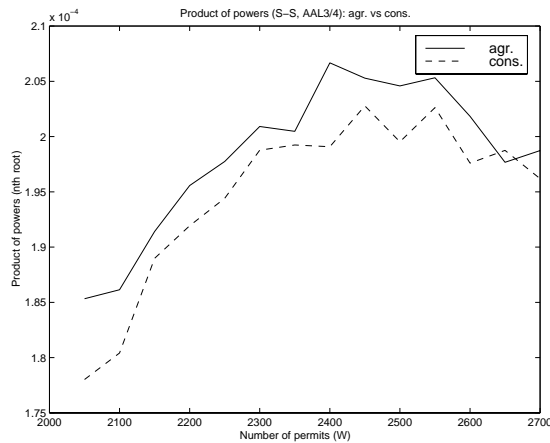


- UBR traffic: Self-Similar (or Poisson) with rate $\lambda_1 = 53,000$ packets per second, average number of cells in packet is 20.
- ABR traffic: Poisson with rate $\lambda_2 = 27,000$ packets per second, average number of cells in packet is 40.
- VBR traffic: MMPP with parameters $r_1 = r_2 = 6.7 \times 10^{-5}$ secs, $\lambda_3(1) = 403,500$, $\lambda_3(2) =$ cells per second.

Comparison, Model 1: Conservative vs Aggressive, Self-Similar Traffic



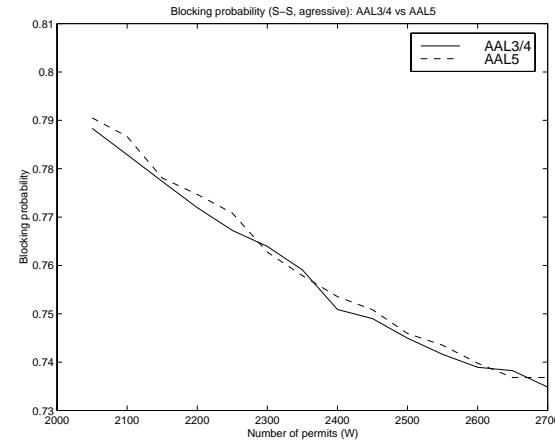
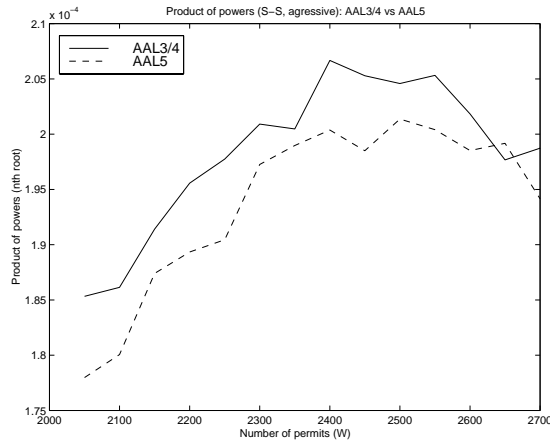
AAL5. Product of Powers (left), Blocking Probability (right)



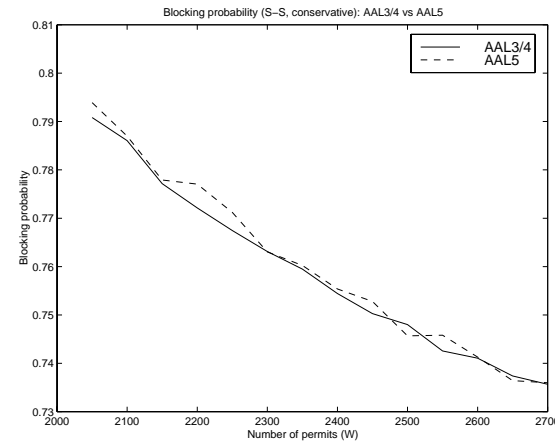
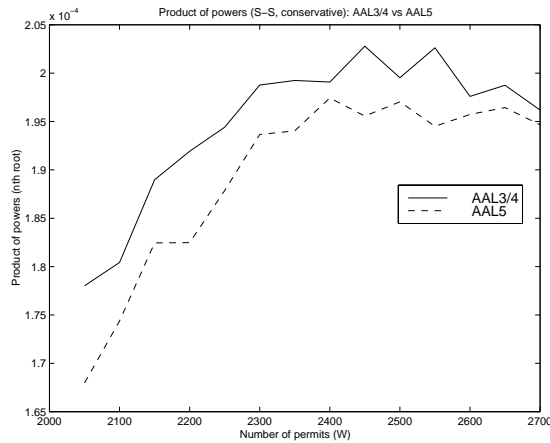
AAL3/4. Product of Powers (left), Blocking Probability (right)

Remarks: The Product of Powers is higher under Aggressive PSW. Poisson traffic has similar behaviour.

Comparison, Model 1: AAL3/4 vs AAL5, Self-Similar Traffic



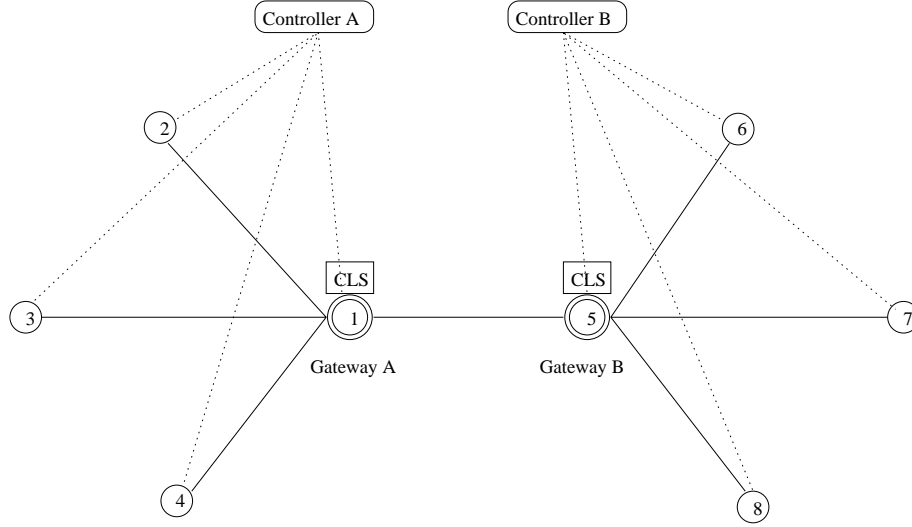
Aggressive PSW. Product of Powers (left), Blocking Probability (right)



Conservative PSW. Product of Powers (left), Blocking Probability (right)

Remarks: Behaviour is similar, but AAL3/4 outperforms AAL5. Similarly for Poisson traffic.

Scalability: Window Management and Estimation by Zones



- Network Topology: divided by **zones**.
- Gateways: entry points to zones, with CLS.
- Window Size: there are W_A permits circulating in zone A and W_B in zone B.
- Decentral Controllers: they gather permits from the local destinations as well as from cells exiting the zone. These latter are sent by the Gateways to the local controller.

Let $P_A = \prod_{o \in A, d} \frac{\lambda_{o,d}}{T_{o,d}}$, so that the global product of powers is $P_A P_B$. Call $G(A)$ and $G(B)$ the Gateways at the respective zones. The following result allows us to use local estimation to produce a scalable control.

Theorem 1 *The best approximation to the product of powers P_A using only one statistics from Zone B and all local information from Zone A is given by:*

$$\hat{P}(A) = \prod_{\{o \in A, d \in A\}} \frac{\lambda_{o,d}}{T_{o,d}} \prod_{o \in A, d \notin A} \frac{\lambda_{o,d}}{T_{o,G(A)} + T_{(G(A),G(B))} + \hat{T}_B}, \quad \text{where } \hat{T}_B = \frac{1}{N_B} \sum_{d \in B} T_{G(B),d}$$

is the arithmetic mean delay in Zone B, N_b being the total number of origin-destination pairs in Zone B.

Hierachical Estimation: Each zone estimates the local delays and all throughputs. Gateways gather this information to pass the *arithmetic mean zone delay* to other Gateways. The global Product of Powers can thus be estimated with minimal information exchange at the controllers.