A Scalable Service Architecture for Providing Strong Service Guarantees

Nicolas Christin and Jörg Liebeherr

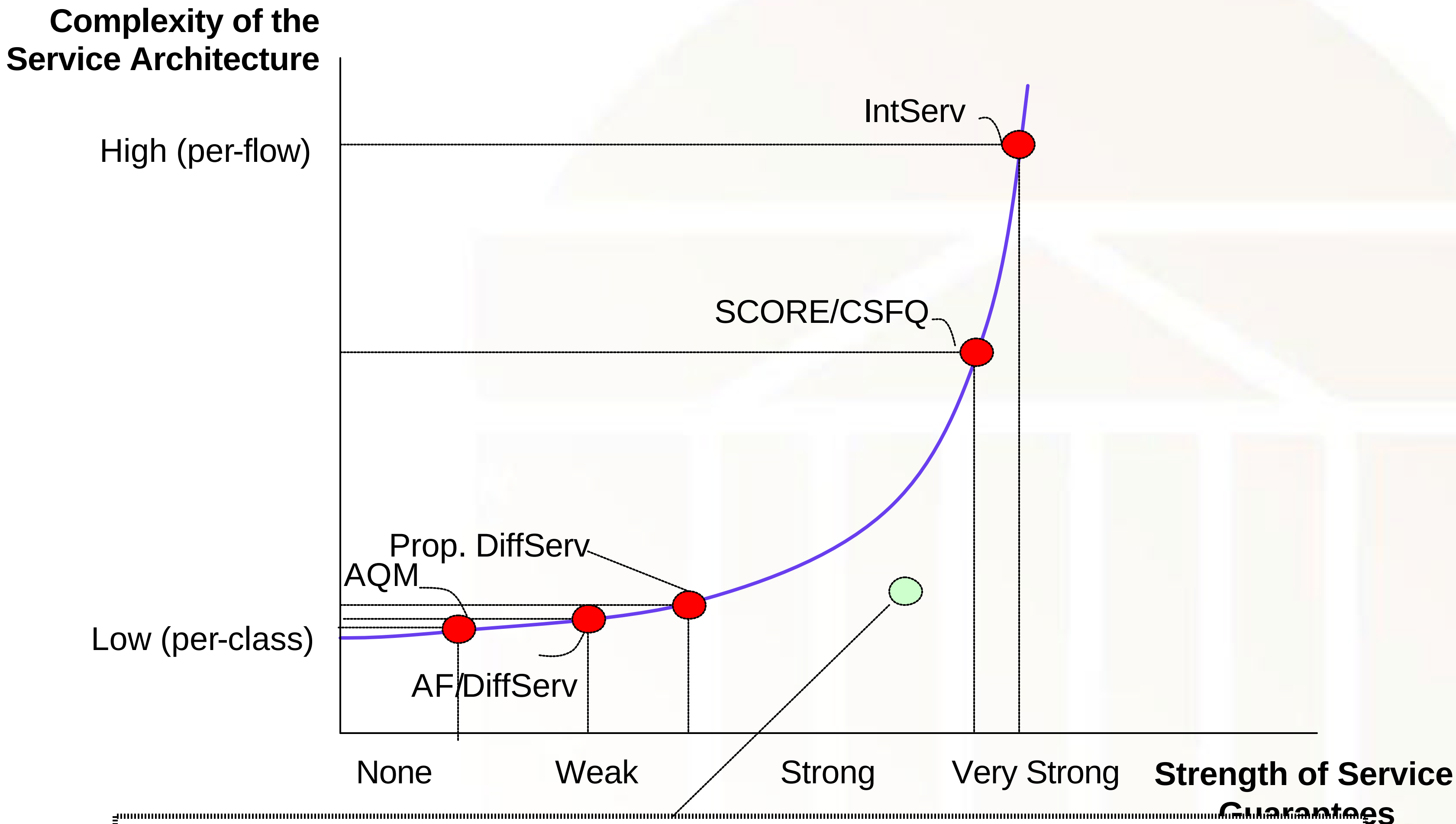
University of Virginia
Department of Computer Science
P.O. Box 400740
Charlottesville, VA 22904-4740

[nicolas|jorg]@cs.virginia.edu

Outline

- Problem: strong QoS with low complexity
- Proposed approach
  - The Quantitative Assured Forwarding service
  - Reference Algorithm: Joint Buffer Management and Scheduling (JoBS)
- Heuristic realization of JoBS
- Current work
- Conclusions

Problem and Context



Challenge: Can we provide strong service guarantees with low computational complexity?

Previous Attempts at Strong QoS with Low Complexity

- Proportional Delay and Loss Differentiation (Dovrolis et al., 1999)
  - No absolute guarantees
- Mean-Delay Proportional Scheduler (Barghavan et al., 2000)
  - No guarantees on losses
- ABE Service (Hurley et al., 2001)
  - Strong guarantees but only two classes
- SCORE/CSFQ/DPS (Stoica & Zhang, 1999)
  - Strong guarantees, but high complexity at access points
- Dynamic Core Provisioning (Campbell and Liao, 2001)
  - No absolute guarantees on delays

Quantitative Assured Forwarding

- Guarantees provided on a per-hop, per-class basis
- No admission control, no signaling, no traffic conditioning
  - No per-flow operations
- **Proportional and absolute** per-class guarantees for both loss and delay and **lower bound on throughput**

$$\frac{\text{Class-2 loss rate}}{\text{Class-1 loss rate}} \approx 2$$

$$\text{Class-2 delay} \leq 5 \text{ ms}$$

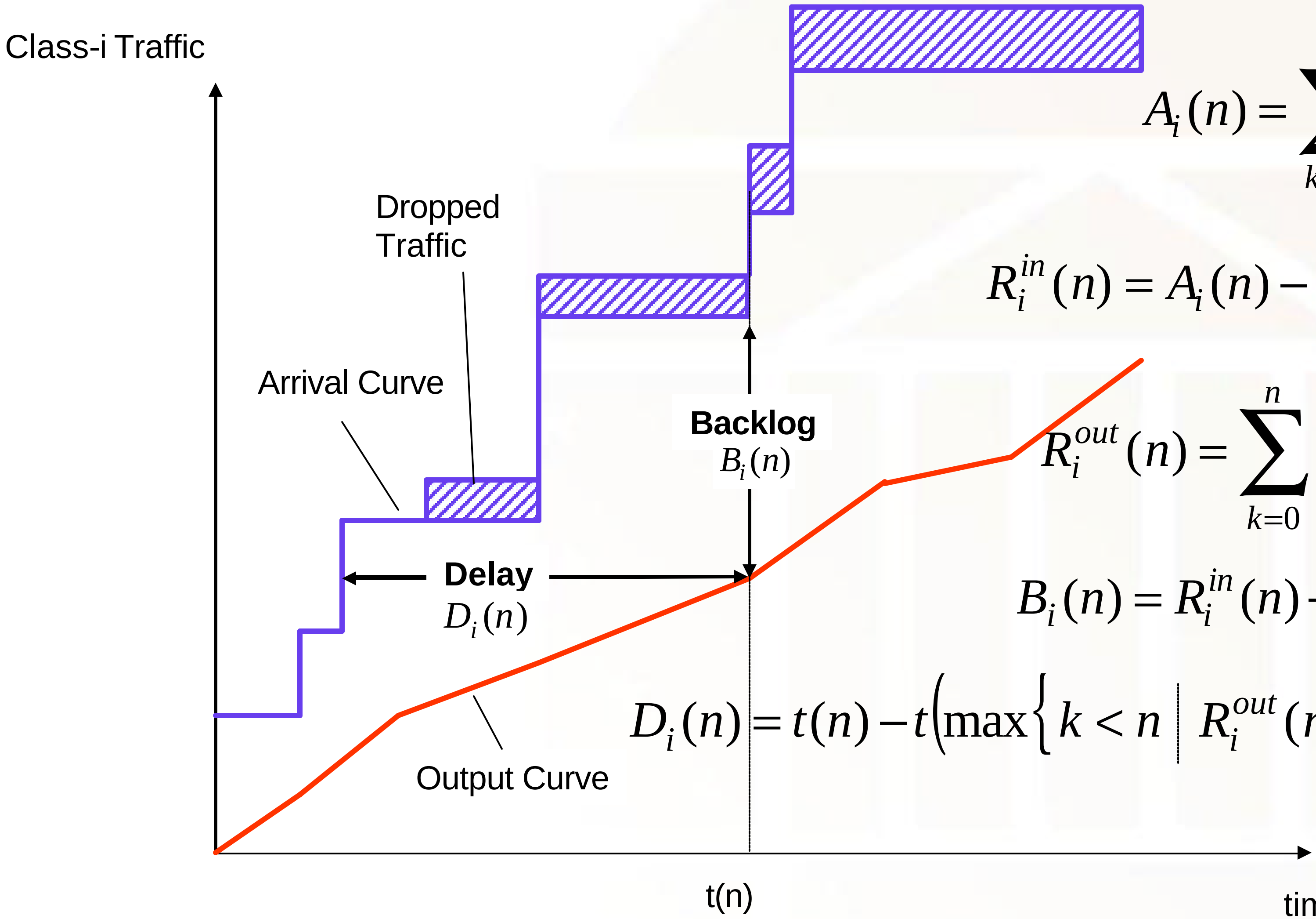
- Concession: service guarantees may need to be temporarily relaxed

None of the existing mechanisms can realize this service

JoBS – Joint Scheduling and Buffer Management

- Key technique:
  - Buffer management and scheduling at the output link of a router are addressed by a single algorithm → JoBS
- JoBS mechanisms:
  - Service rate allocation to traffic classes
  - Service rate allocation is periodically adjusted
  - Rate allocation is based on projections of delays and loss rate
  - If no feasible rate allocation exists, drop traffic
  - If necessary, relax service guarantees
- JoBS can realize the Quantitative Assured Forwarding service

Arrivals, Departures, Losses at a Node



$$A_i(n) = \sum_{k=0}^{n} a_i(k)$$

$$R_i^{in}(n) = A_i(n) - \sum_{k=0}^{n} l_i(k)$$

$$R_i^{out}(n) = \sum_{k=0}^{n} r_i(k) \cdot \Delta t(k)$$

$$B_i(n) = R_i^{in}(n) - R_i^{out}(n)$$

$$D_i(n) = t(n) - t\left(\max\left\{ k < n \;\middle|\; R_i^{out}(n) \geq R_i^{in}(k) \right\}\right)$$

JoBS

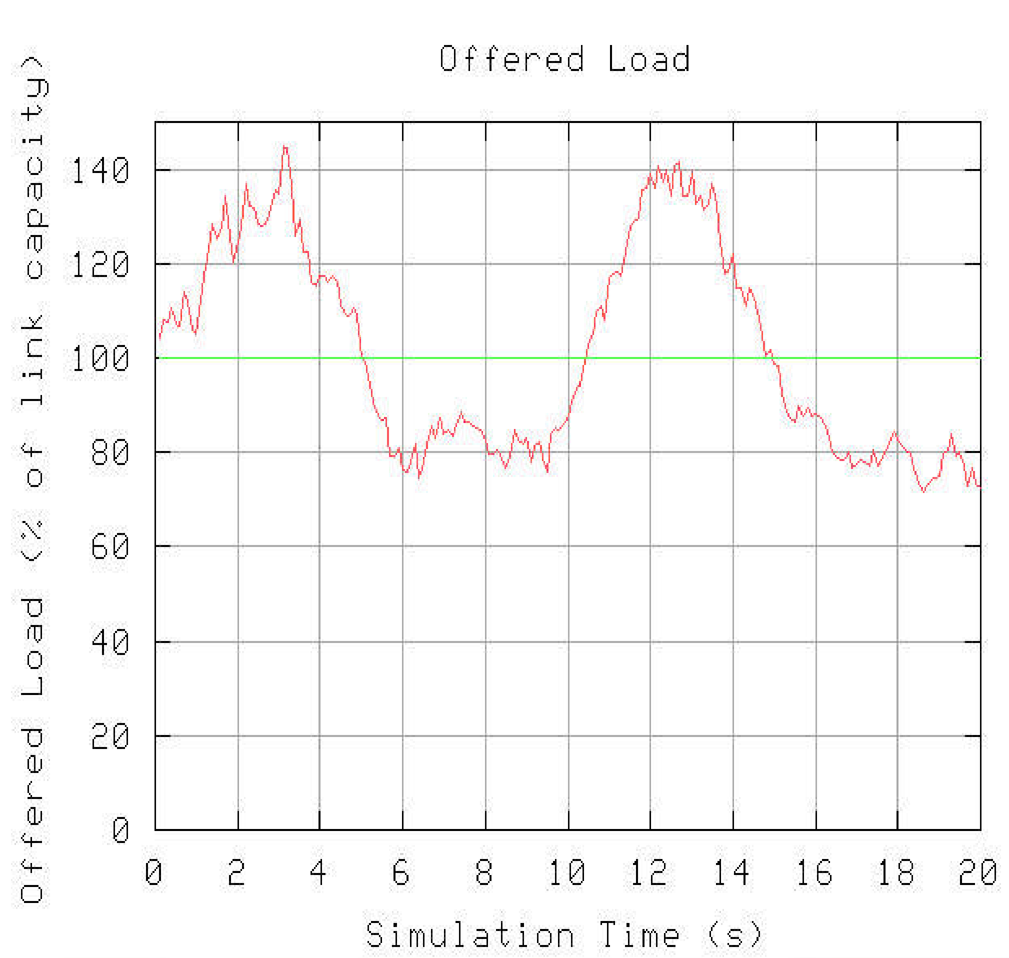
- Future delays are projected
- New rate allocations and drop decisions are obtained from an optimization

<u>Minimize:</u>	losses and changes to the rate allocation,
<u>Subject to:</u>	- absolute bounds on loss, and delay.
	- proportional service differentiation
	- system constraints (e.g., buffer size)

- If constraint system becomes infeasible, relax constraints in a specified order

Evaluation by Simulation

- Single node simulation
- Output link capacity = 1 Gbps,
- Buffer size = 6.25MB,
- Bursty arrival pattern: superposition of 200-550 Pareto sources (α=1.2).
- The offered load curve varies between 70% and 150% of the link capacity,
- 4 traffic classes,
- Each class contributes 25% of the total traffic.



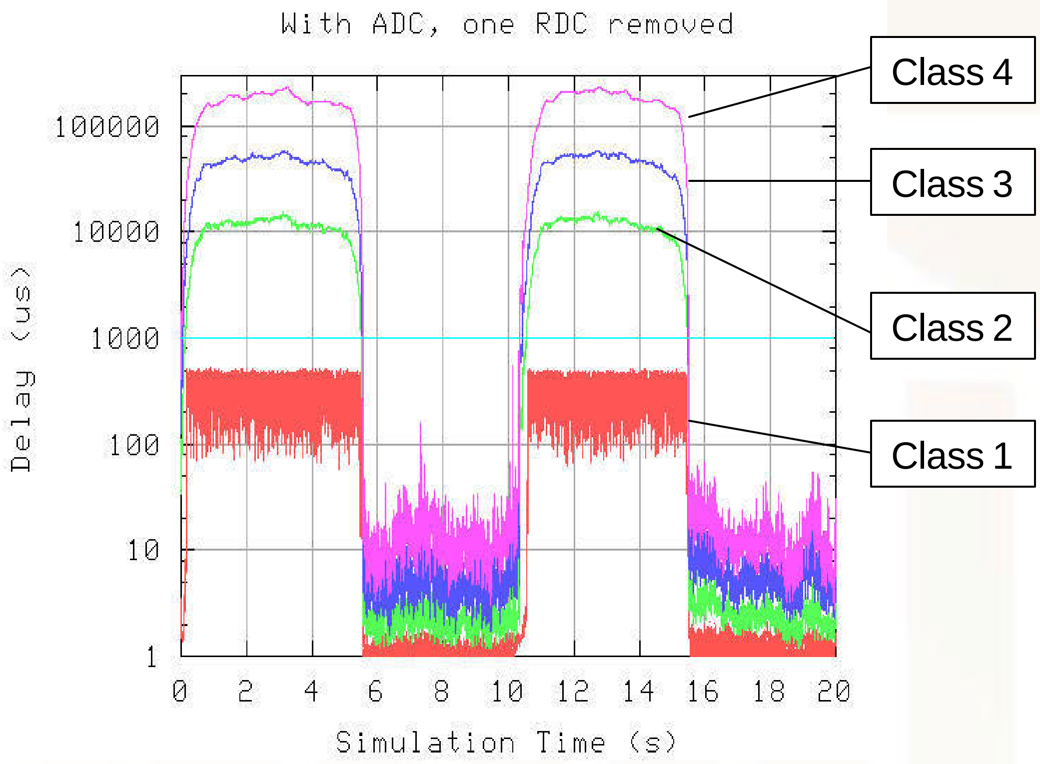
Simulation Results: Delay

$$\frac{\text{Class-4 delay}}{\text{Class-3 delay}} \approx 4$$

$$\frac{\text{Class-3 delay}}{\text{Class-2 delay}} \approx 4$$

Class-1 delay ≤ **1 ms**

$$\frac{\text{Class-(i+1) loss}}{\text{Class-i loss}} \approx 2$$



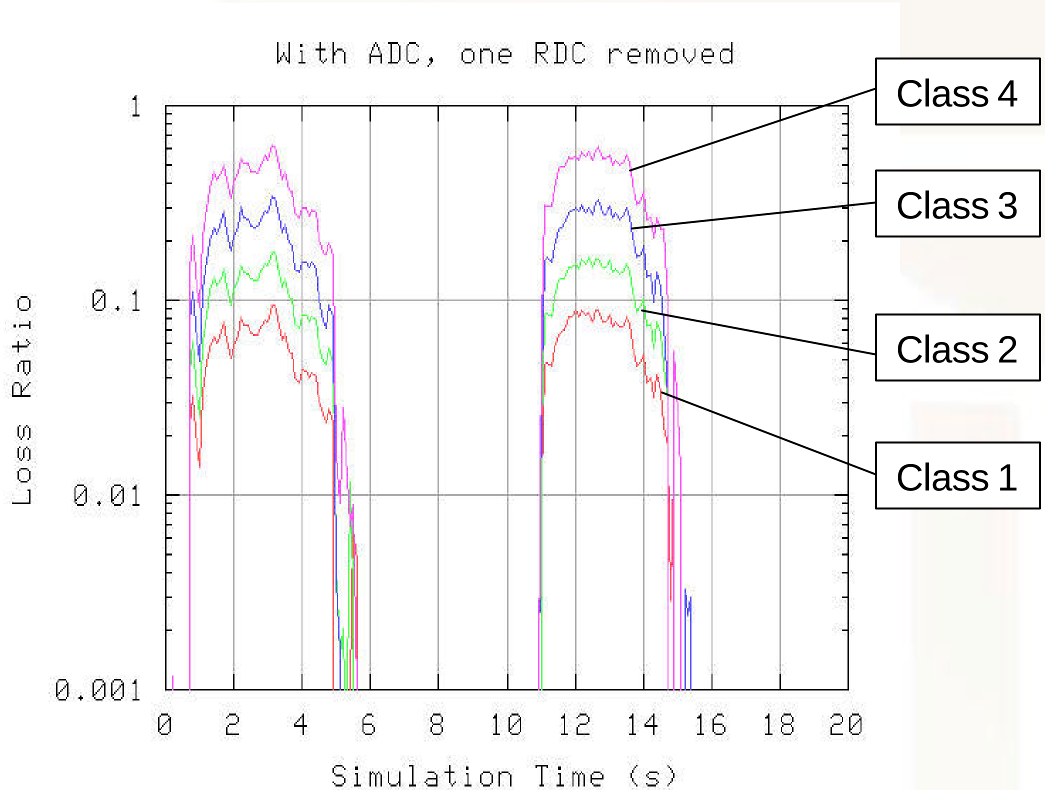
Simulation Results: Loss

$$\frac{\text{Class-4 delay}}{\text{Class-3 delay}} \approx 4$$

$$\frac{\text{Class-3 delay}}{\text{Class-2 delay}} \approx 4$$

Class-1 delay ≤ **1 ms**

$$\frac{\text{Class-(i+1) loss}}{\text{Class-i loss}} \approx 2$$



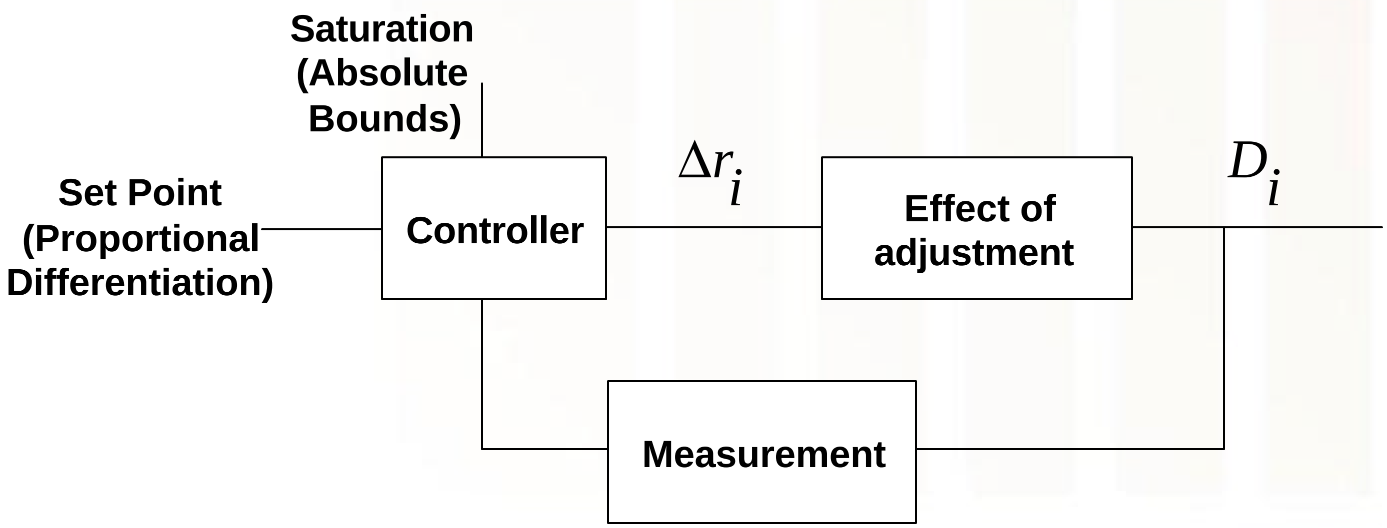
Implementation with Low Complexity: Feedback Loops

- Service rate allocation and loss rates can be viewed in terms of a recursion:

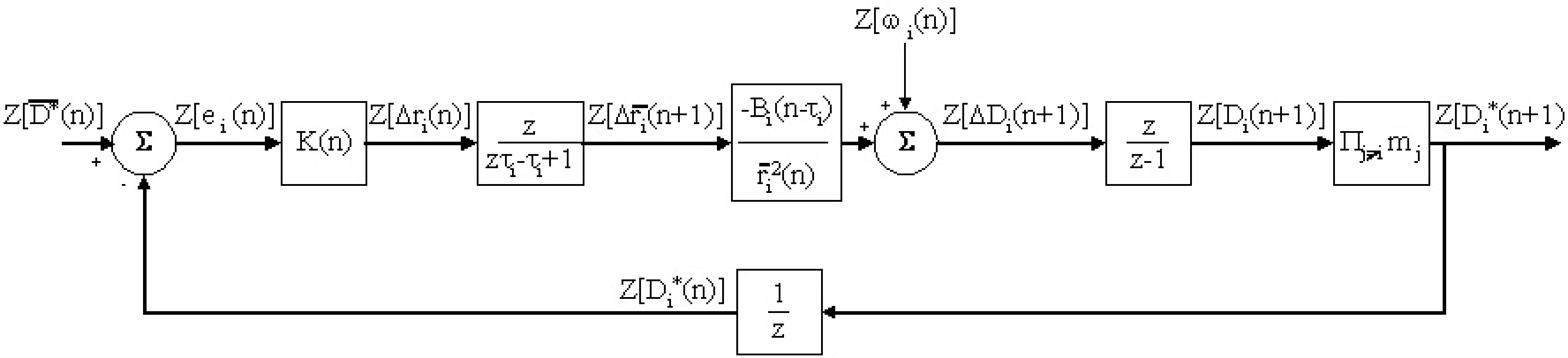
$$r_i(n) = r_i(n-1) + \Delta r_i(n)$$

$$p_i(n) = p_i(n-1)\frac{A_i(n-1)}{A_i(n)} + \frac{l_i(n)}{A_i(n)}$$

- Feedback loops



A Feedback Control Solution



- Linearization of the non-linear system around an operating point.
  - Allows to use linear control theory tools (e.g., derivation of a stability condition)
- Controller is simple: $\Delta r_i(n) = K(n) \cdot e_i(n)$
  - $e_i(n)$ is the deviation of the class-i delay from the desired proportional differentiation
  - $K(n)$ is a proportional coefficient
- Losses are handled by a similar feedback mechanism

Conditions on the Delay Controllers

- Stability condition (proportional differentiation):

$$-2 \cdot \min_i \left\{ \frac{B_i(n)}{\prod_{j \neq i} m_j \cdot D_i^2(n)} \right\} \leq K(n) \leq 0$$

- Saturation effects (absolute delay/throughput guarantees):

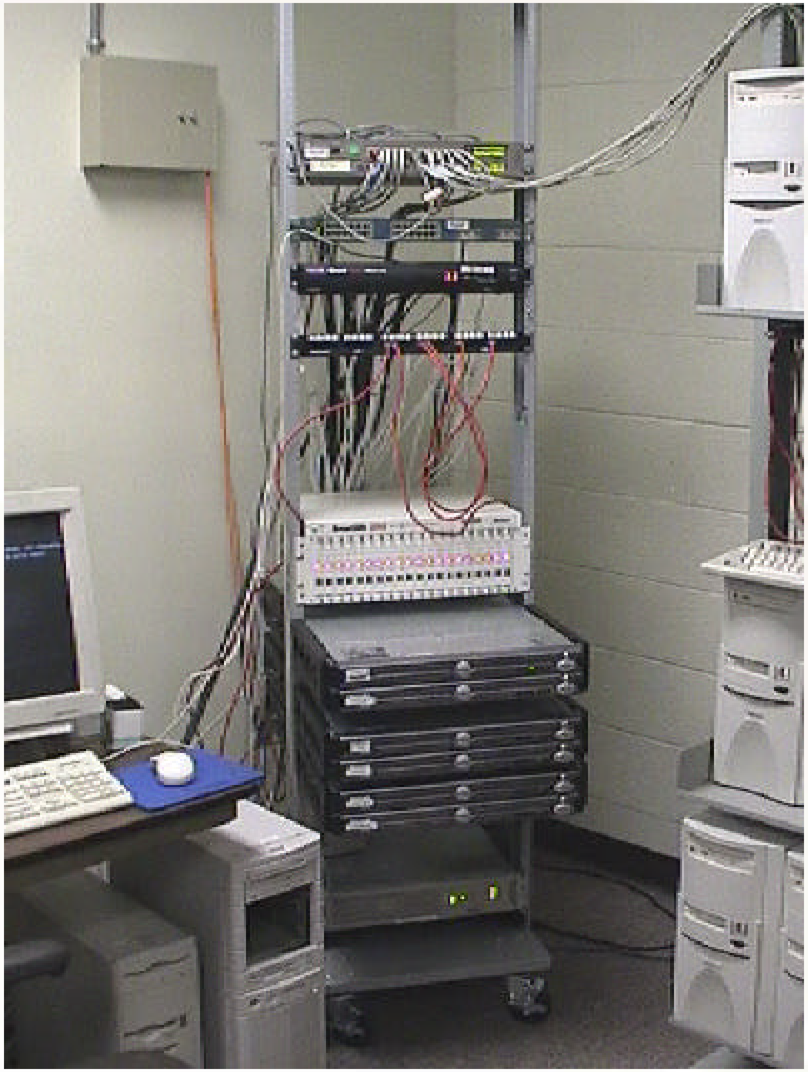
$$K(n) \geq \max_i \left(\frac{r_{i,\min}(n) - r_i(n-1)}{e_i(n)} \right)$$

with

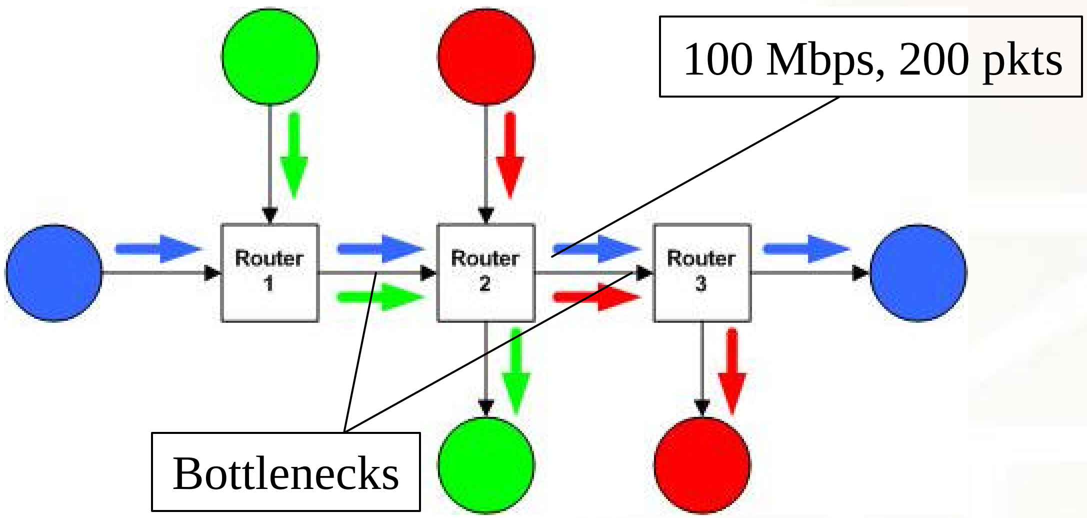
$$r_{i,\min}(n) = \max \left\{ \frac{B_i(n)}{d_i - D_i(n)}, m_i \cdot c_{B_i(n) \geq 0} \right\}$$

Implementation

- Implementation in FreeBSD kernel
  - Testbed of 6 Pentium IIIs 1Ghz with multiple interfaces
  - Allows testing at 100 Mbps (FastEthernet)
  - Developed for ALTQ 3.0 (package allowing modifications to the network stack), now part of ALTQ 3.1



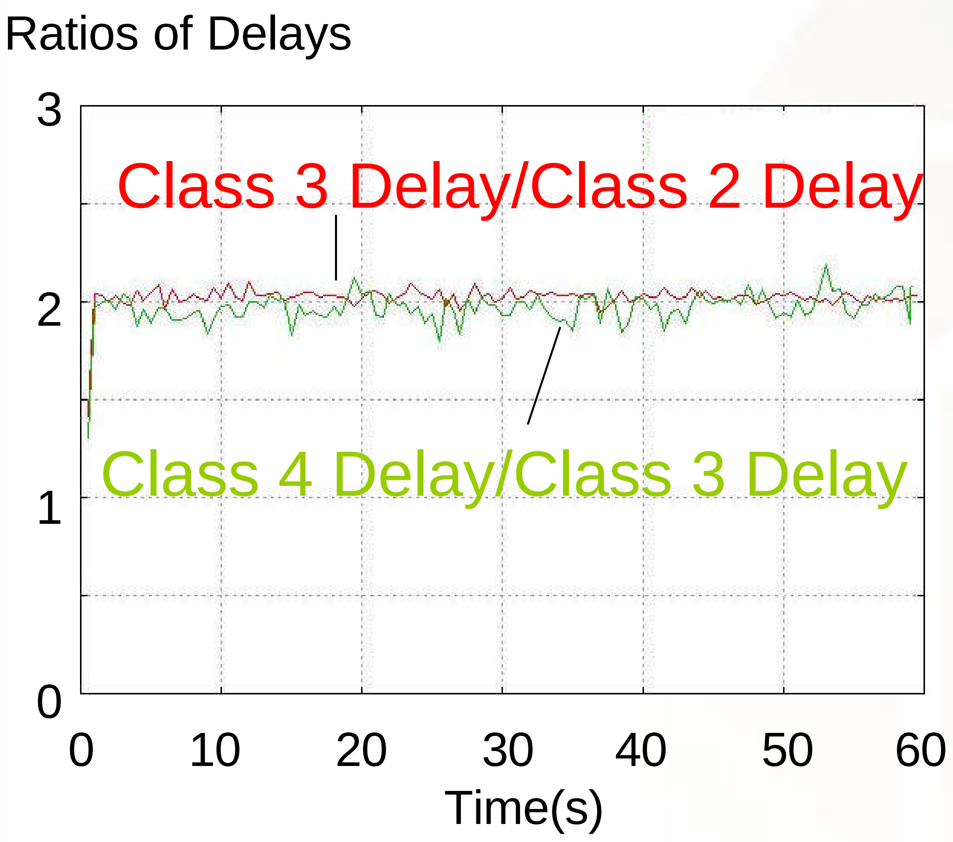
Experimental Setup

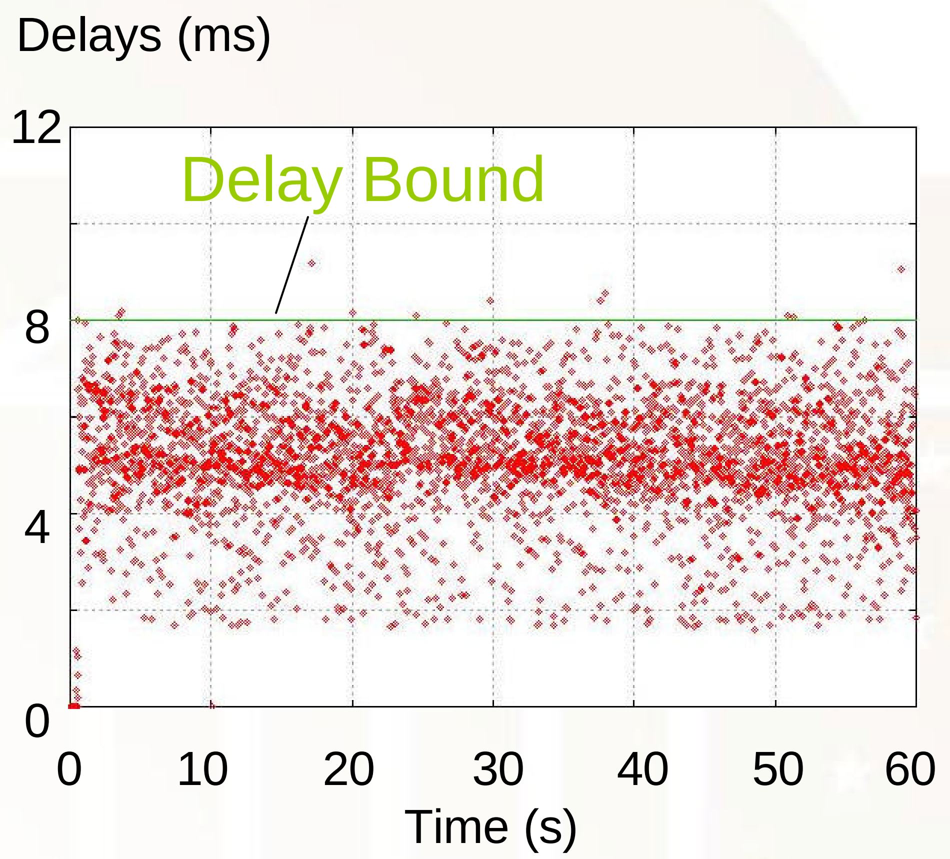


Class	No. of Flows	Proto.	Traffic
1	6	UDP	On-off
2	6	TCP	Greedy
3	6	TCP	Greedy
4	6	TCP	Greedy

Class	d_i	L_i	μ_i	k_i	k'_i
1	8 ms	1 %	-	-	-
2	-	-	35 Mbps	2	2
3	-	-	-	2	2
4	-	-	-	N/A	N/A

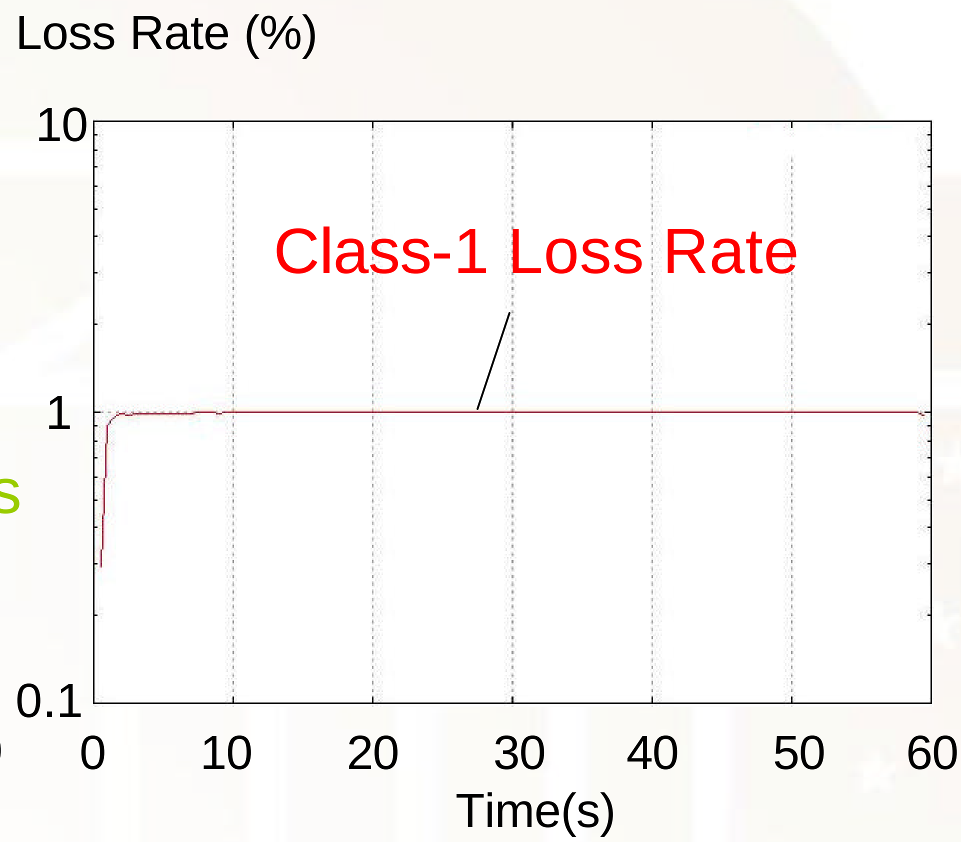
Delay Differentiation (at Router 1)



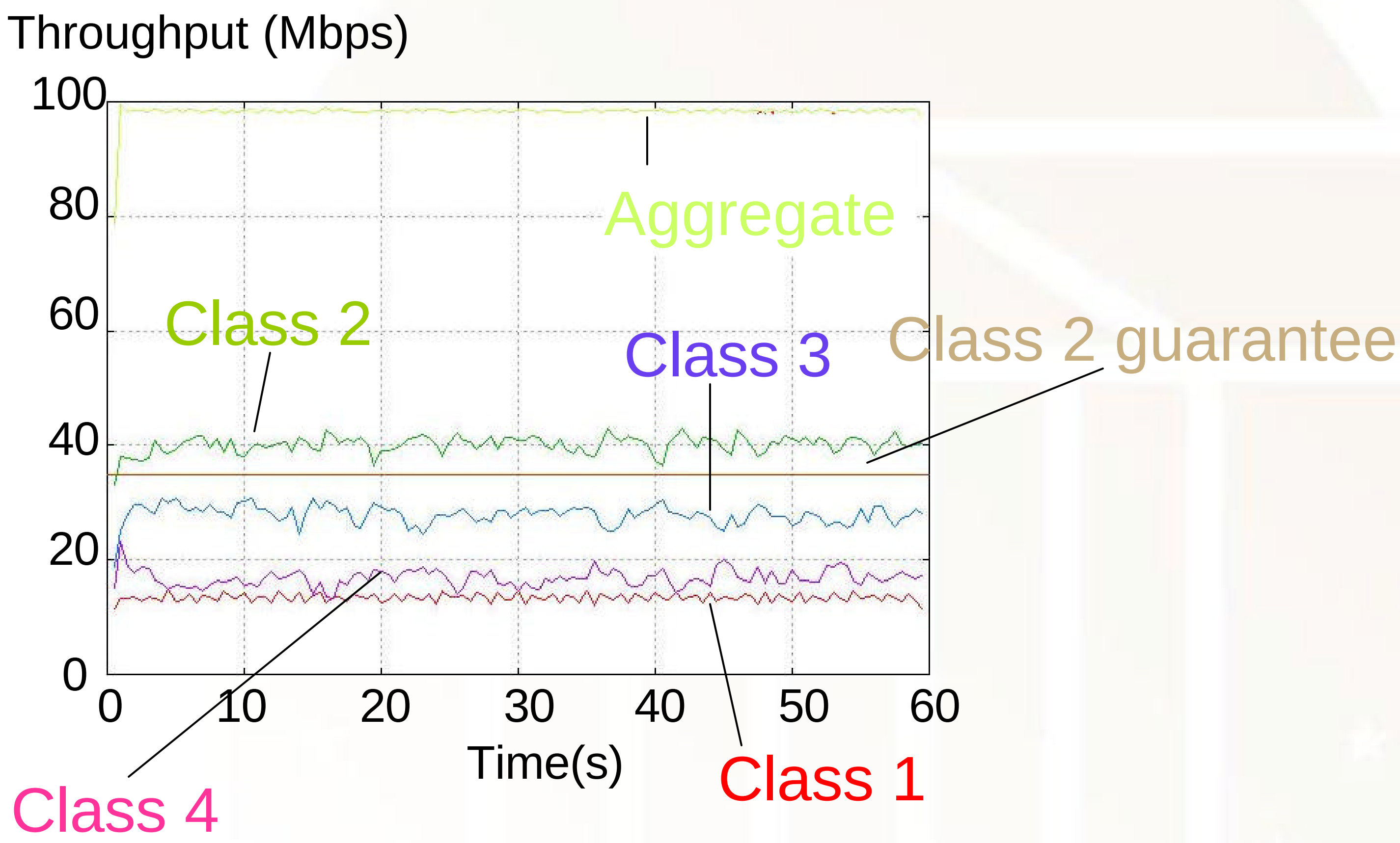


→ Similar results can be observed at Router 2

Loss Differentiation (at Router 1)



Throughput Differentiation (at Router 1)



Current Work: Traffic Regulation

- No admission control and no policing:
  - Service guarantees can be infeasible (cf. delay violations in the example)
- Key observation:
  - Most traffic is TCP
  - Majority of traffic is generated by a limited number of flows (“heavy-hitters”)
- Mechanisms:
  - Identify heavy-hitters via flow filtering
  - Estimate congestion window size and RTT of heavy-hitters
  - Control traffic from heavy-hitters via ECN marking

Does not require any changes to TCP!

Conclusions

- Architecture w/ Low complexity/Strong guarantees
- Can be implemented at high-speeds
- Current work:
  - Avoid infeasible set of service guarantees by regulating traffic using TCP congestion control algorithms
- Software and more information is available at:

http://qosbox.cs.virginia.edu