Scalable Architectures for Integrated Traffic Shaping and Link Scheduling

Jennifer Rexford
Network Mathematics Research
Networking and Distributed Systems
AT&T Labs Research in Florham Park, NJ

`http://www.research.att.com/~jrex`

Outline

- Mechanisms for end-to-end quality of service
- Overview of leaky-bucket traffic shaping
- Scalable shaper architecture for high-speed switches
- Integrating traffic shaping with fair scheduling
- Performance evaluation through simulation
- Conclusions and future work

End-to-End Quality of Service

Application specification:

- Traffic parameters (burstiness, bandwidth)
- Performance requirements (delay, rate, loss, jitter)

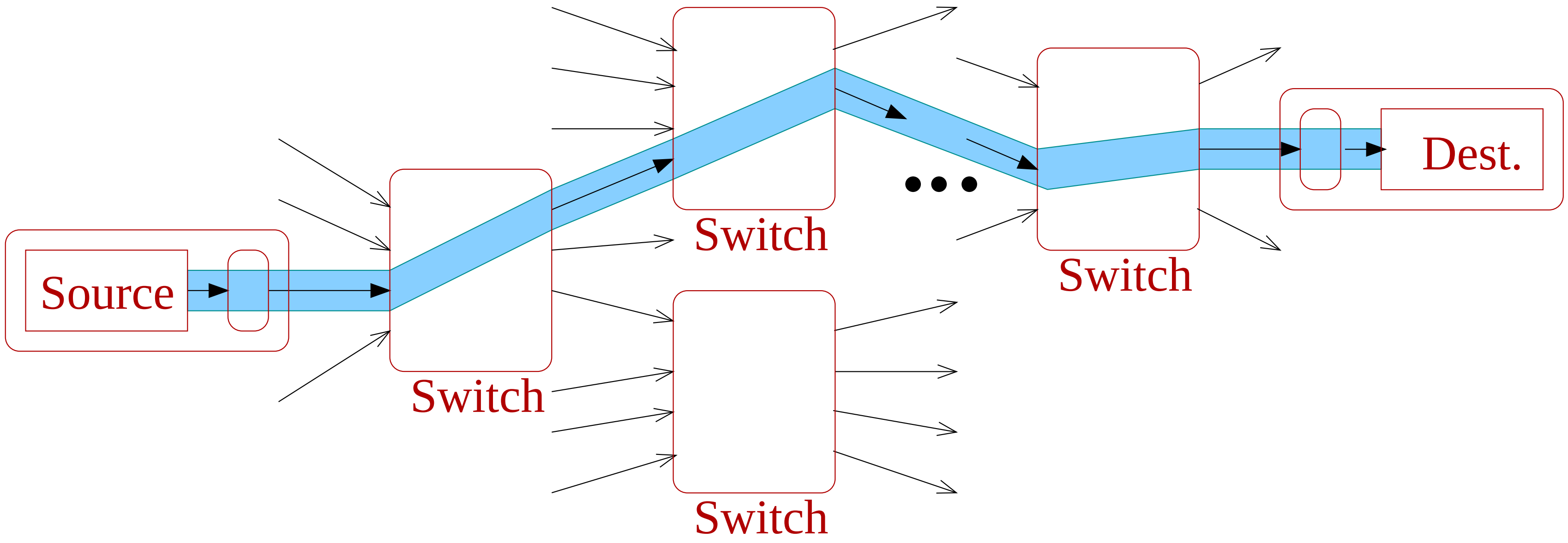
Network connection establishment:

- Select a suitable multi-hop route to the destination
- Reserve buffer and bandwidth resources along path

Network data transfer:

- Enforce connection traffic contract (policing/shaping)
- Arbitrate between connections for buffers and links

Connection Traversing Multiple Switches



- **Traffic shaping**: smooth traffic to obey contract
- **Buffer management**: store transient congestion
- **Link scheduling**: share link between connections

Constraints in Emerging Networks

Network traffic sharing a single link

- Thousands of connections/sessions
- Diverse traffic parameters (video, audio, data)
- Diverse performance requirements (rate, delay, loss)

$\Rightarrow$ powerful algorithms for managing bandwidth & buffers

Network technology trends

- High-speed links (e.g., 155 Mbit/sec; 1 Gbit/sec)
- Small data transfer unit (e.g., 53-byte ATM cell)

$\Rightarrow$ scalable and efficient hardware switch designs

Shaping a Large Number of Connections

Network edge:

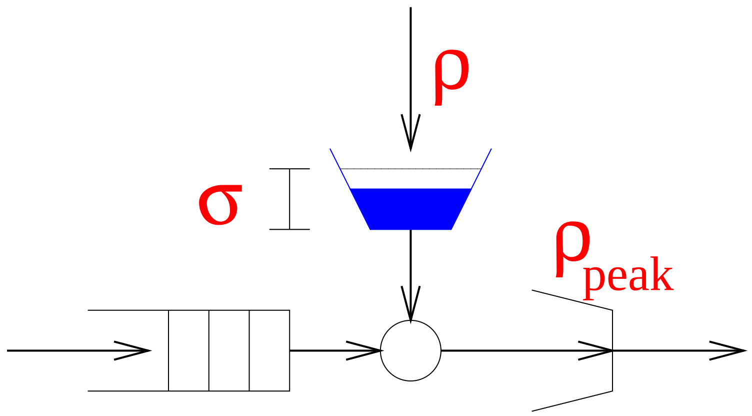
- Large-scale web or video servers with multimedia data
- Shaping as a service for multiple smaller end systems
- Shaping to avoid overloading the receiving LAN/host

Network interior:

- Boundary between different providers in the backbone
- Schedulers that reshape connections to limit jitter
- Available-bit-rate connections with rate enforcement at "virtual sources" inside the network

$\Rightarrow$ high-speed shaping for a large number of connections

Leaky-Bucket Traffic Shaping

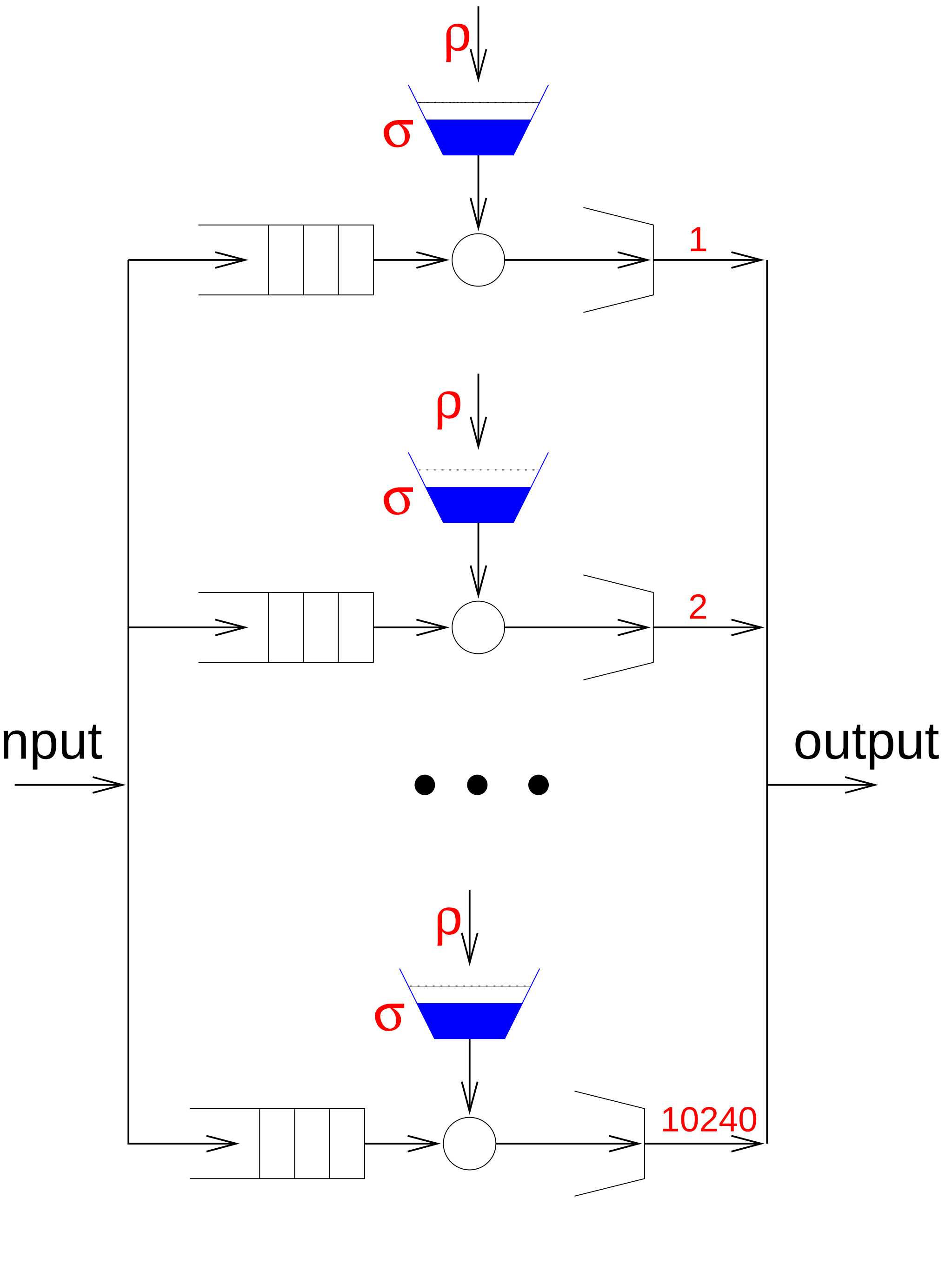


Connection traffic parameters:

- Burst size σ
- Sustainable bandwidth ρ
- Peak rate ρ_{peak}

$\Rightarrow$ leaky-bucket controller for each connection

Shaping Multiple Connections



- Scalable implementation with n connections?
- Handling of contention between connections?

Computing Cell Conformance Times

Avoiding continuous simulation of leaky bucket:

- Only need the bucket state on cell arrival
- Compute new state from state at last arrival
- Compute the time when a token becomes availabe

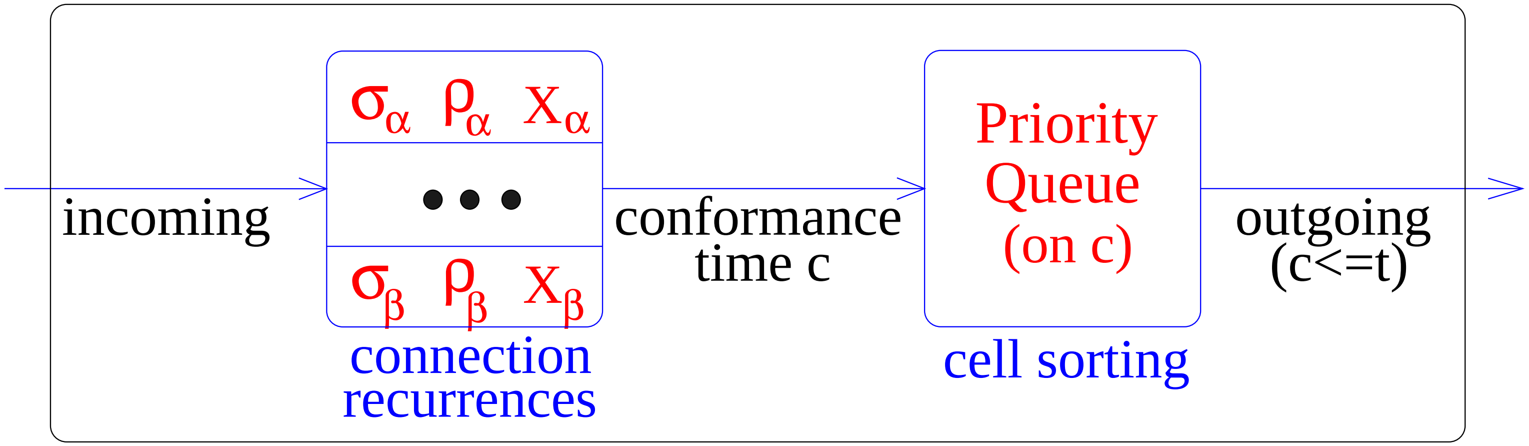
$\Rightarrow$ small, bounded amount of work on cell arrival

Cell arriving at time t:

- Bucket depth σ, bandwidth ρ, and state X
- Estimated arrival time $X = \max\{X + 1/\rho, t\}$
- Cell conformance time $c = \max\{X - \sigma/\rho, t\}$

$\Rightarrow$ schedule incoming cell for departure at time c

Scalable Traffic Shaping



- **Parameters**: burstiness σ, bandwidth ρ, state X
- **Recurrences**: compute conformance time c
- **Cell sorting**: transmit cell with smallest $c \leq t$

$\Rightarrow$ priority queue with thousands of connections

$\Rightarrow$ handling multiple cells with the same value of c

Scalable Sorting Schemes

Implementation scalability challenges:

- Large number of connections
- Multiple cells from the same connection
- Range of burst and bandwidth parameters

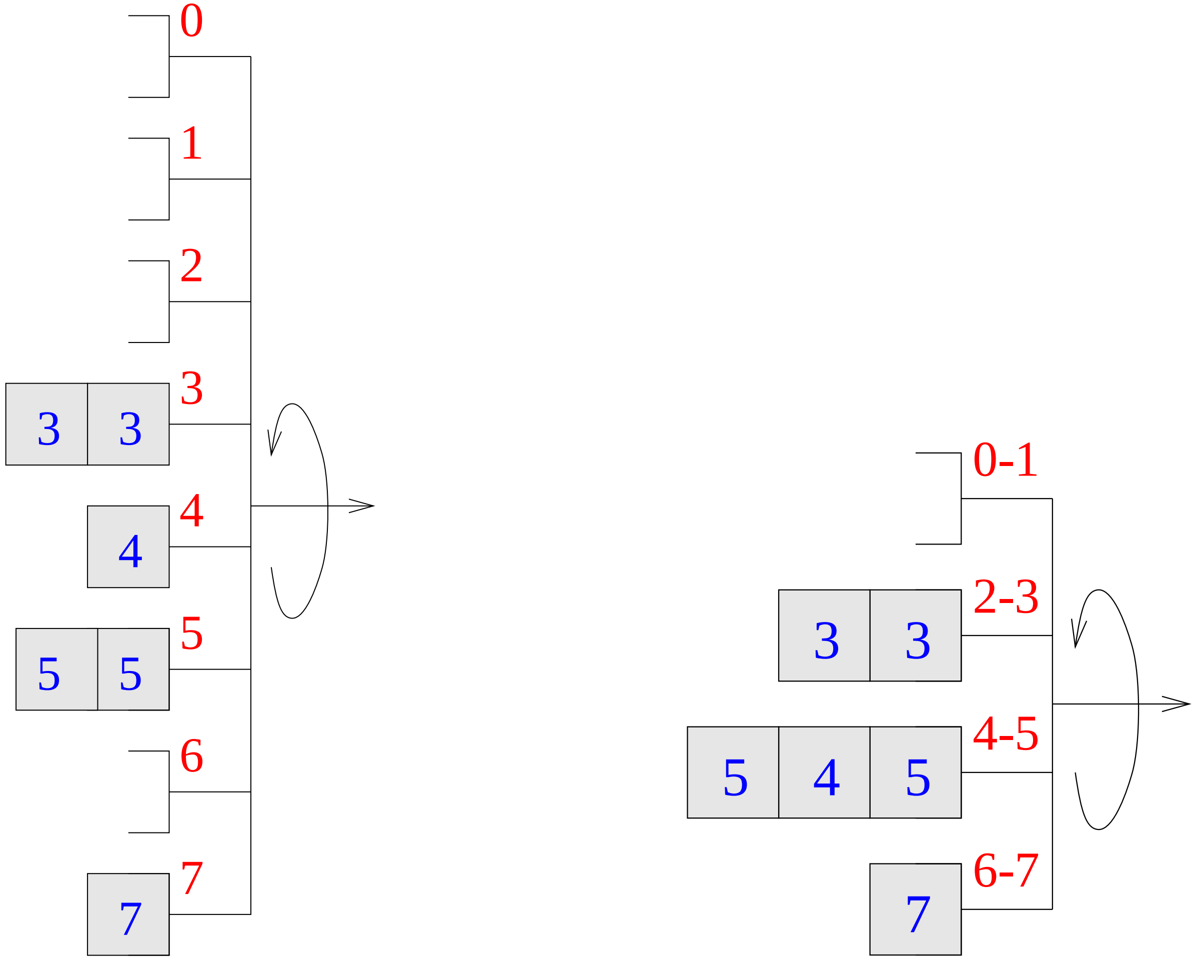
$\Rightarrow$ large number of cells with wide range of c values

Calendar queue sorting:

- Collection of b first-in-first-out bins
- Grain of g time units per sorting bin
- Incoming cell joins the bin based on its c value

$\Rightarrow$ cell joins bin number $\lfloor (c \bmod bg)/g \rfloor$

Traffic Shaping With Calendar Queues



(a) $b = 8$ bins of grain $g = 1$ (b) $b = 4$ bins of grain $g = 2$

Reducing the Number of Bins

Head-of-line cells:

- Only consider the first cell in each connection α
- Keep other cells in per-connection FIFOs
- Head cell must conform soon ($c_\alpha \leq t + 1/\rho_\alpha$)

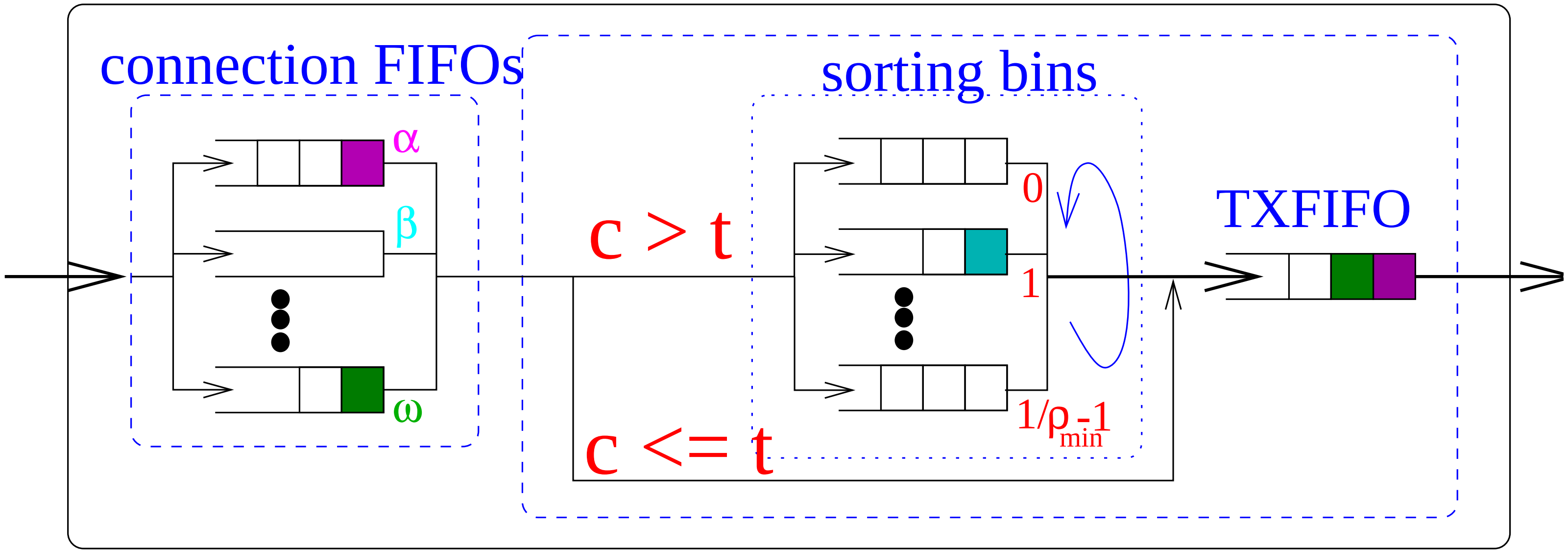
$\Rightarrow$ bounds range of bins to $bg = 1/\rho_{\min}$

Conforming head-of-line cells:

- Transient backlog of conforming traffic ($c_\alpha \leq t$)
- Hard to bound number of cells and range of c_α
- Avoid exact sorting of backlogged conforming cells

$\Rightarrow$ single FIFO for head-of-line cells with $c_\alpha \leq t$

Scalable Architecture



- Connection FIFOs: all but the head-of-line cells
- Sorting bins: non-conforming head-of-line cells
- Transmission FIFO: conforming head-of-line cells

$\Rightarrow$ bounded work on cell arrival and departure

Event-Driven Algorithm

Cell Arrival

```
compute c for the new cell;
if (connection is not empty)
  cell joins connection FIFO;
else if (c == t)
  cell joins TXFIFO;
else
  cell joins bin ⌊(c mod bg)/g⌋;
```

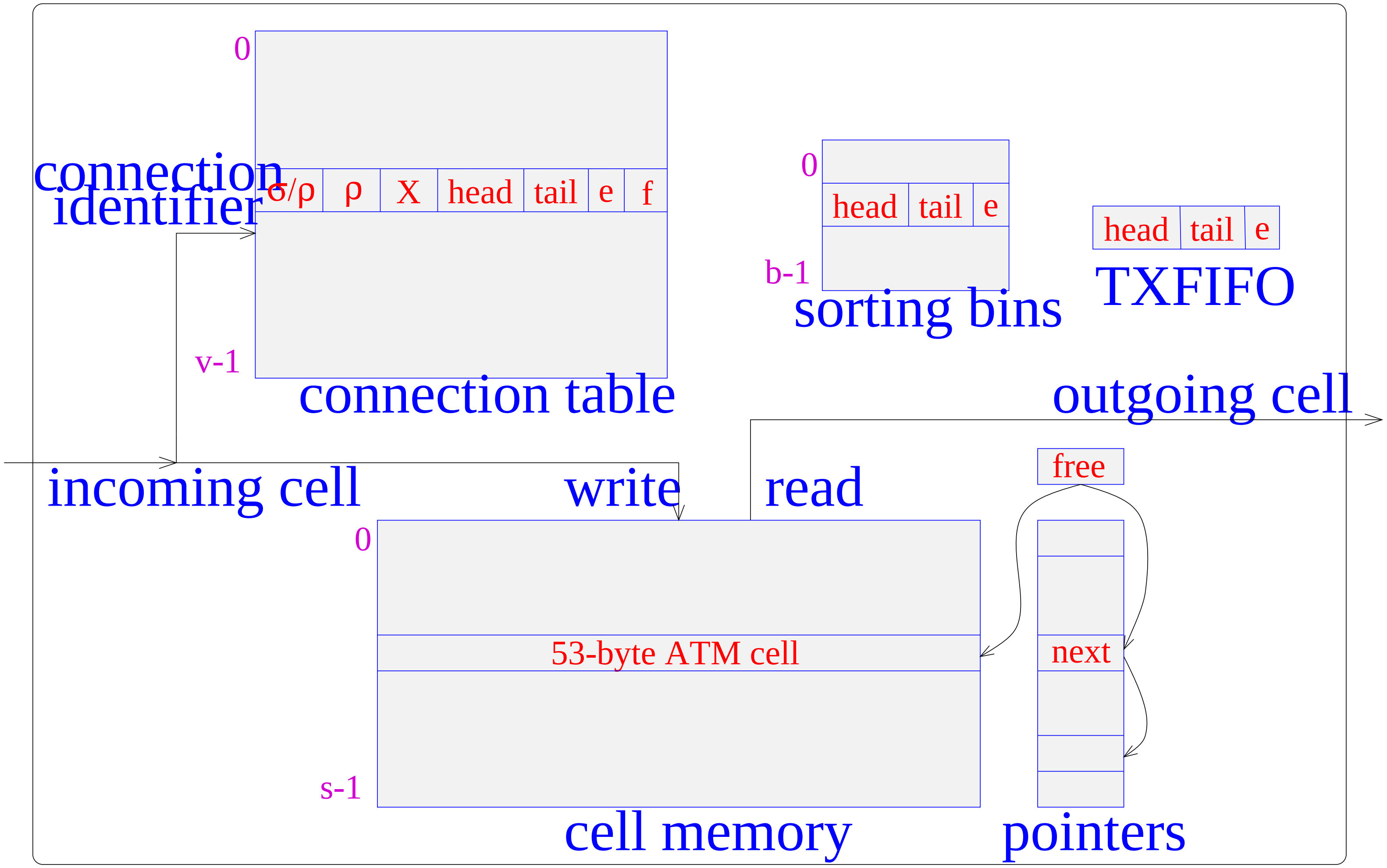
Cell Departure

```
transmit current cell;
if (connection is not empty)
  get cell from connection FIFO;
  if (c ≤ t)
    cell joins TXFIFO;
  else
    cell joins bin ⌊(c mod bg)/g⌋;
```

Conforming Bin

```
if (t mod g == g − 1)
  append bin ⌊(t mod bg)/g⌋ onto TXFIFO;
```

Data Structures in Shaper Architecture



Wide Range of Connection Rates

Challenges of diverse connection rates ρ_α:

- Bandwidths can range over 4–5 orders of magnitude
- High-bandwidth traffic needs accurate sorting
- Low-bandwidth connections have large c values

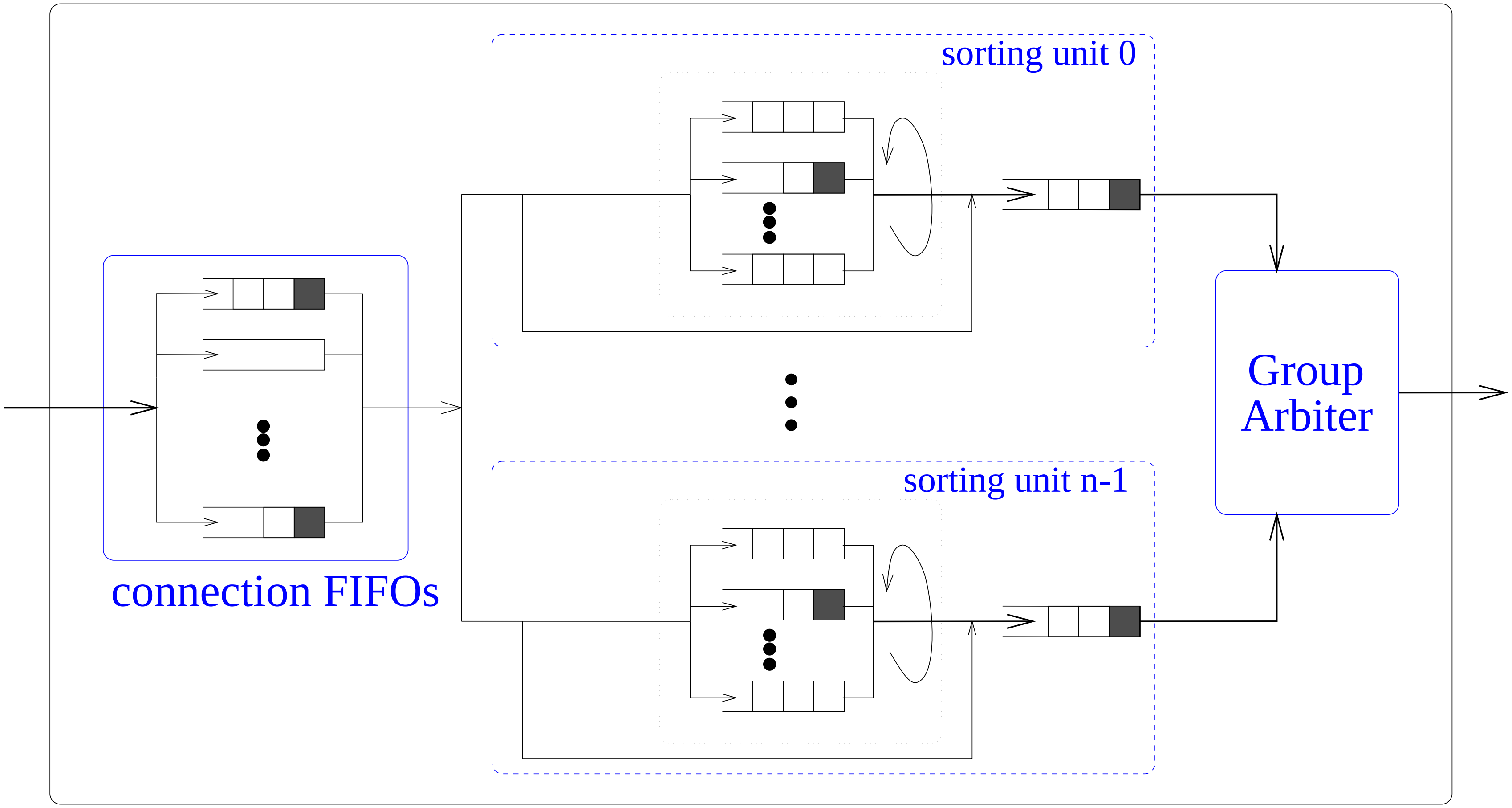
$\Rightarrow$ tension between range (bg) and grain (g) of sorting

Segregate connections based on rates ρ_α:

- Group rates ρ_α within a factor of m (say, 8 or 16)
- Small number of groups $n = \log_m\{\rho_{\max}/\rho_{\min}\}$
- Tailor range bg and grain g within each group

$\Rightarrow$ small number of groups with small number of bins

Hierarchical Architecture



Performance Scalability

Performance challenges:

- Effects of ties (multiple cells with same c value)
- Large number of connections ganging up on others
- Bursty connections denying service to others
- Inaccuracies from approximate sorting (grain g)

Performance implications:

- Cell shaping delay $(t - c > 0)$
- Traffic distortions $(\sigma^{out} > \sigma)$
- Difficulty of provisioning downstream resources
- Cell loss (buffer overflow) at downstream switches

Congestion Behavior of Existing Shapers

Exact sorting:

- Cell departure in order of increasing c values
- FIFO service amongst conforming cells
- Favors connections with large σ values

$\Rightarrow$ distorts *low-burstiness* connections

Initial scheme with per-connection queueing:

- Cell departure inserts next cell into the TXFIFO
- Round-robin service amongst conforming connections
- Favors connections with smaller ρ values

$\Rightarrow$ distorts *high-bandwidth* connections

Weighted Fairness Within a Group

Minor modification of proposed architecture:

- Connections should receive service proportional to ρ
- Allow α to have $\lfloor \rho_\alpha / \rho_{\min} \rfloor$ cells in the sorting unit
- Maintain counter of number of cells in the sorting unit
- Event-driven algorithm and no additional sorting bins
- Complexity independent of the number of connections

$\Rightarrow$ weighted-round-robin scheduling of conforming cells
with small work on cell arrivals and departures

Weighted Fairness Between Groups

- Small number of groups with a wide range of rates
- Divide bandwidth fairly at the group levels
- Group i has rate $\phi_i = \sum \rho_{\alpha_i}$ from busy connections
- Adjust weight as connections become idle or busy
  - α_i becomes busy: $\phi_i = \phi_i + \rho_{\alpha_i}$
  - α_i becomes idle: $\phi_i = \phi_i - \rho_{\alpha_i}$
- Extensions to existing fair scheduling algorithms

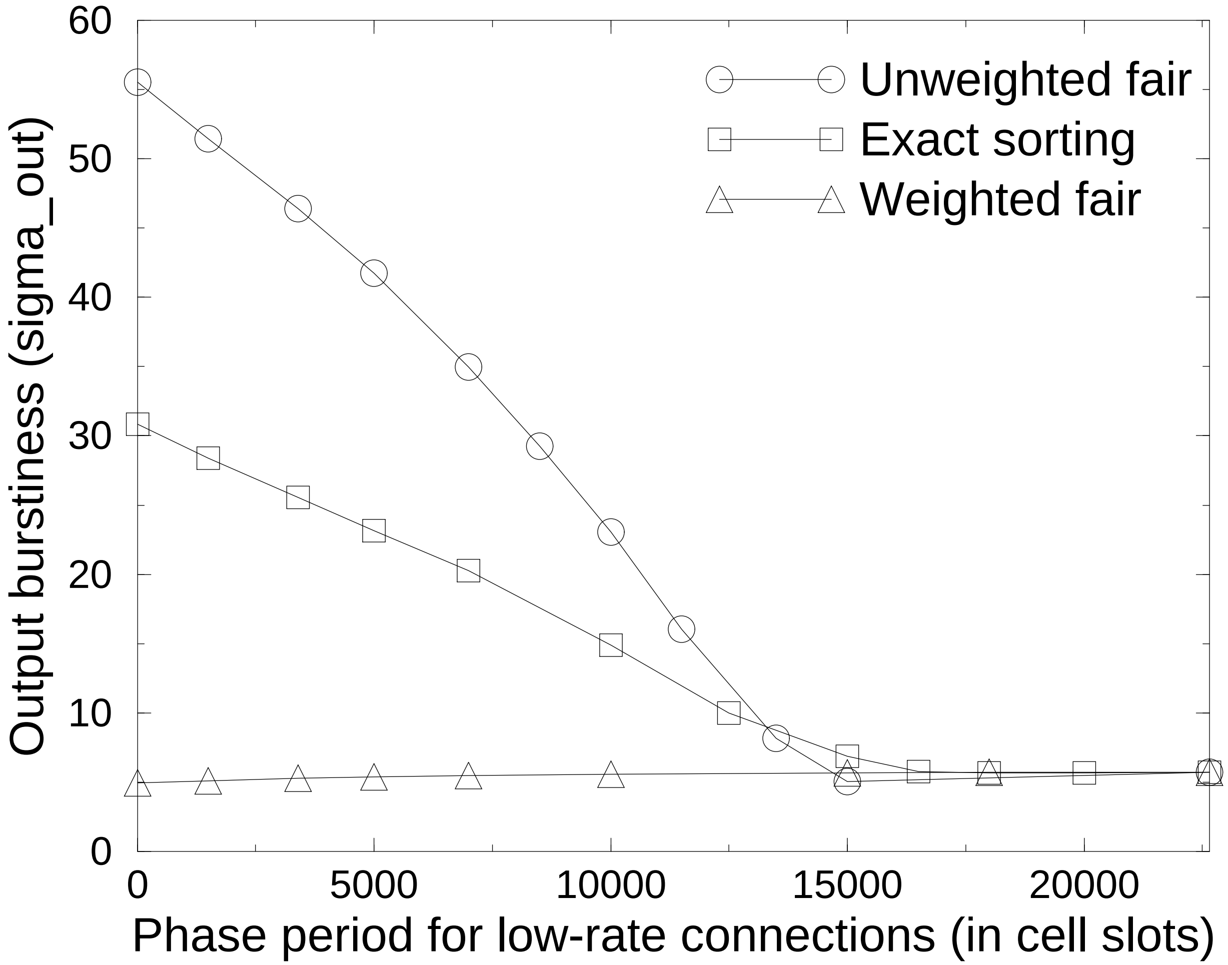
$\Rightarrow$ weighted fair arbitration amongst a *small* number of groups based on the *aggregate* rates

Performance Evaluation

- **Source pattern:** greedy on/off traffic with (σ, ρ)
- **Start times:** uniformly distributed in $[0, x]$
- **Congestion:** varied using start-time parameter x
- **Performance metric:** avg output burstiness $\overline{\sigma}^{out}$

	Arrival		Shaping	
N	σ	ρ	σ	ρ
320	50	0.0022	25	0.0025
10	50	0.0090	5	0.0100

Fair Shaping Within a Group



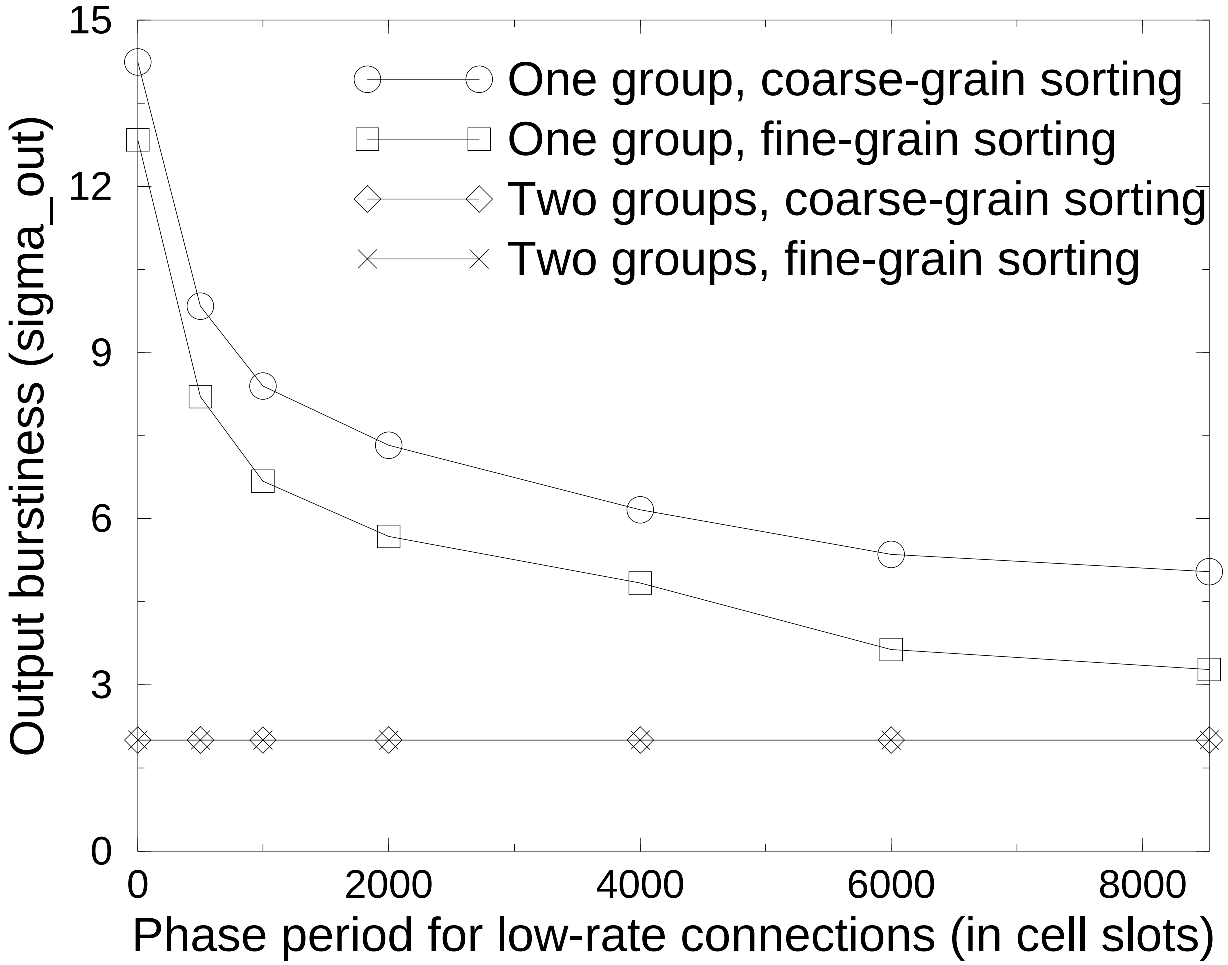
Evaluating Fair Group Arbitration

	Arrival		Shaping	
N	σ	ρ	σ	ρ
5	10	0.0909	0	0.1000
10	50	0.0094	25	0.0100
70	40	0.0048	20	0.0050

Shaper configurations:

- **One group:** 201 bins of grain 1; 10 bins of grain 21
- **Two groups:** 11 bins of grain 1, 201 bins of grain 1; 5 bins of grain 3, 5 bins of grain 41

Performance of High-Rate Connections



Conclusions:

- Large number of connections with diverse traffic
- Importance of efficient shaping and scheduling
- Approximate sorting and hierarchical arbitration

$\Rightarrow$ scalable integration of shaping and scheduling

Future work:

- Performance evaluation under realistic traffic patterns
- Extensions to the hierarchical fair arbitration policies
- Generalizations of integrated shaping/scheduling

$\Rightarrow$ balance between powerful algorithms and efficient architectures in high-speed switches