End-to-end modelling of DiffServ mechanisms

Eeva Nyberg, Samuli Aalto,
Jorma Virtamo, Riikka Susitaival,
Eemeli Kuumola

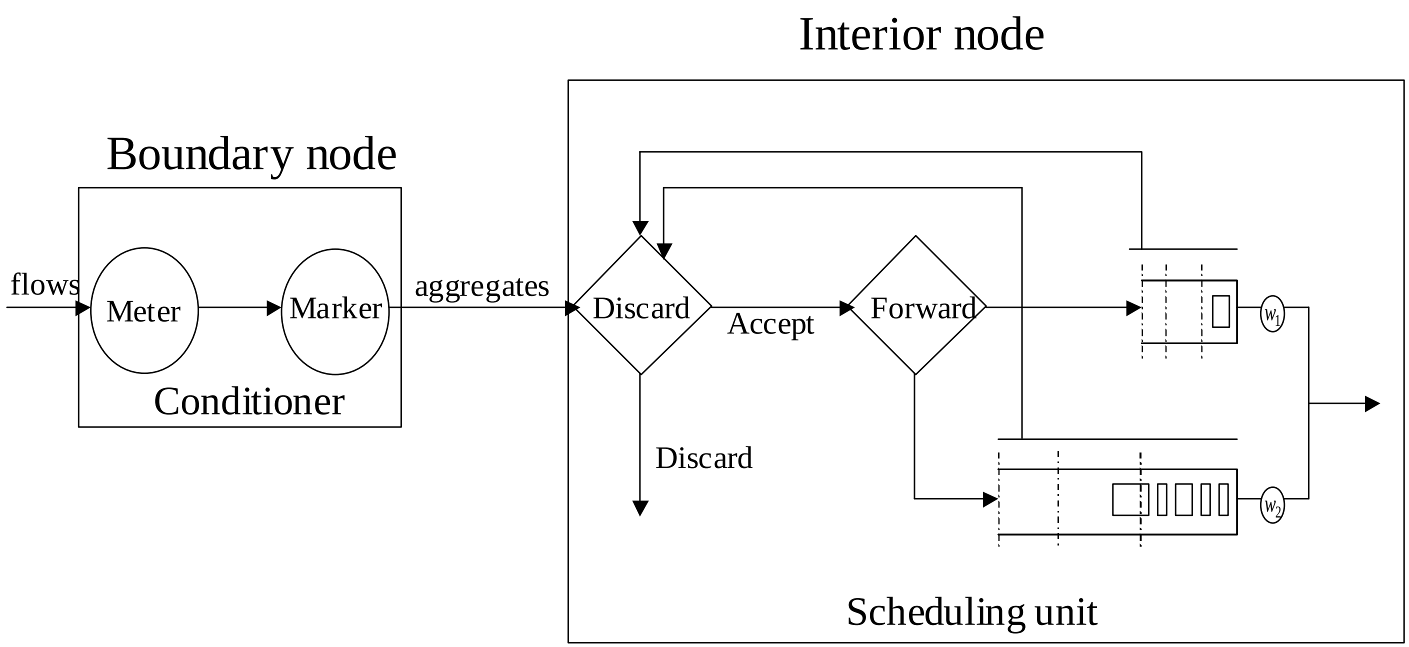
Setting

- Evaluating service differentiation
  - achieved by packet level mechanisms
- Study of how bandwidth is divided between,
  - TCP (elastic) and non-TCP (real-time) flows
  - as a function of number of flows not load

Network model

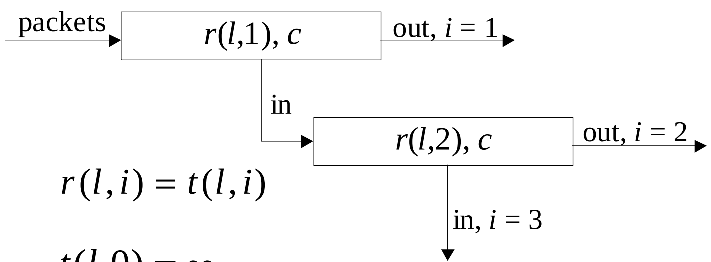
- 2 delay classes
  - Real-time (rt) and elastic non-real time (nrt)
- I priority levels,
  - I highest, 1 lowest
- L flow groups
  - Grouped according to weight $\varphi(l)$ and delay class
  - n_l flows in group l
- Network with one link
  - Capacity $C = 1$

DiffServ model



Metering

- **Token bucket:**
  - Packets are marked in-profile if the bucket holds enough tokens upon arrival, out-of- profile otherwise



$$r(l,i) = t(l,i)$$

$$t(l,0) = \infty$$

$$t(l,i) = \varphi(l)a(i), \quad i = 1,\ldots, I-1$$

$$t(l,I) = 0$$

$$a(i) = 2^{I/2-i-0.5}$$

- **Exponential weighted moving average:**
  - Measured bit rate of previous time instants are exponentially dampened by a time parameter α and the time interval between the measurements

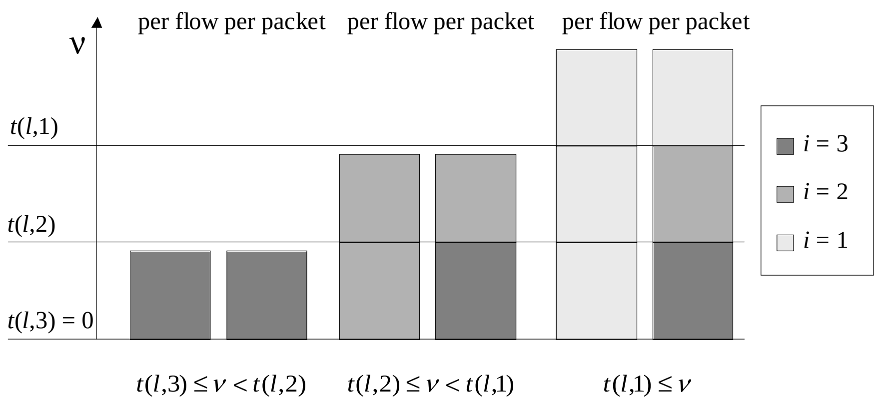
$$mbr(k,j) = \frac{\ln(1-\alpha)}{\ln(1-\alpha/\rho(k,j))}$$

$$\rho(k,j) = \alpha + \rho(k,j)(1-\alpha)^{N_{kj}}$$

$$t(l,i) \le mbr(k,j) < t(l,i-1)$$

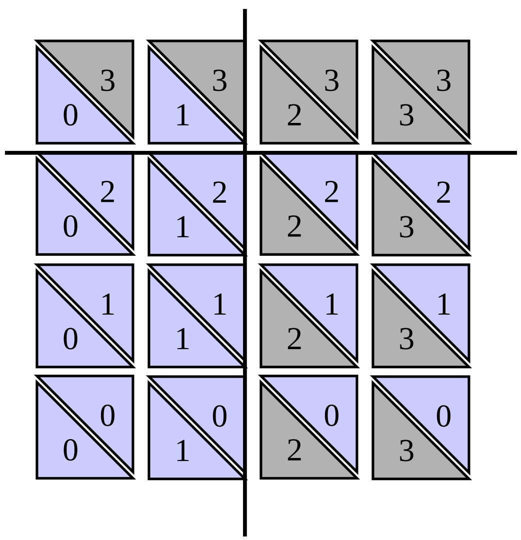
Marking

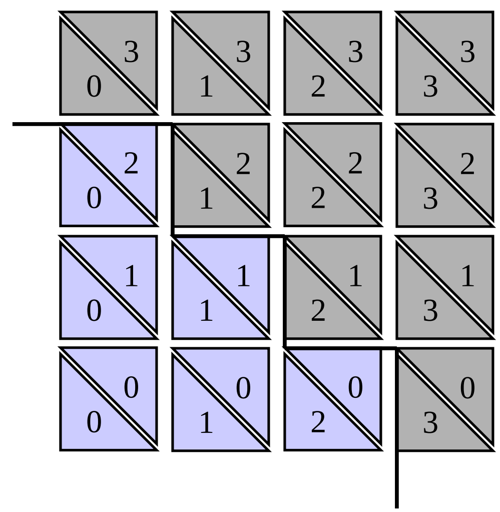
- **Per packet marking:**
  - Only the packets of a flow that exceed the marking threshold are marked to lower precedence level
- **Per flow marking:**
  - Once the measured load of a flow exceeds a marking threshold, all packets of the flow are marked to the same precedence level



Discarding

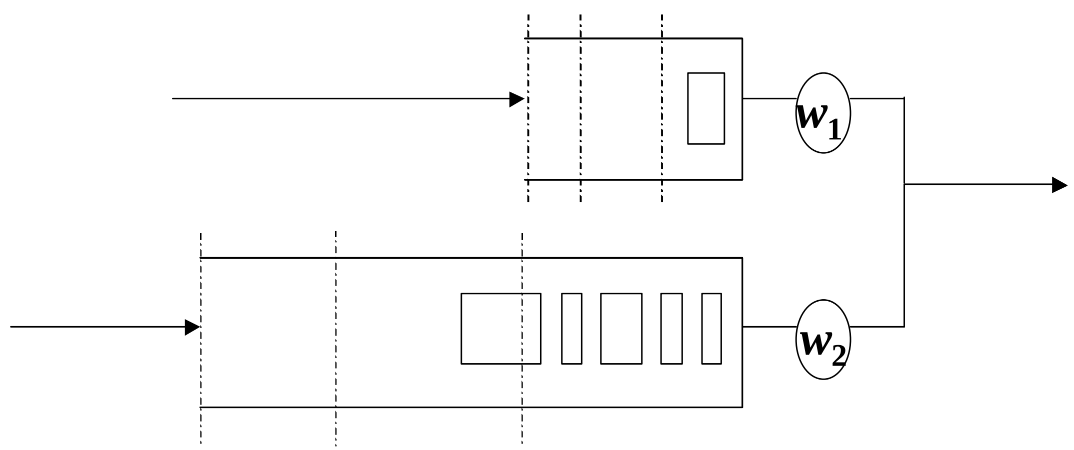
- **Independent**
  - Separate thresholds for each delay class buffer
- **Dependent**
  - Thresholds as a function of buffer contents of both delay classes



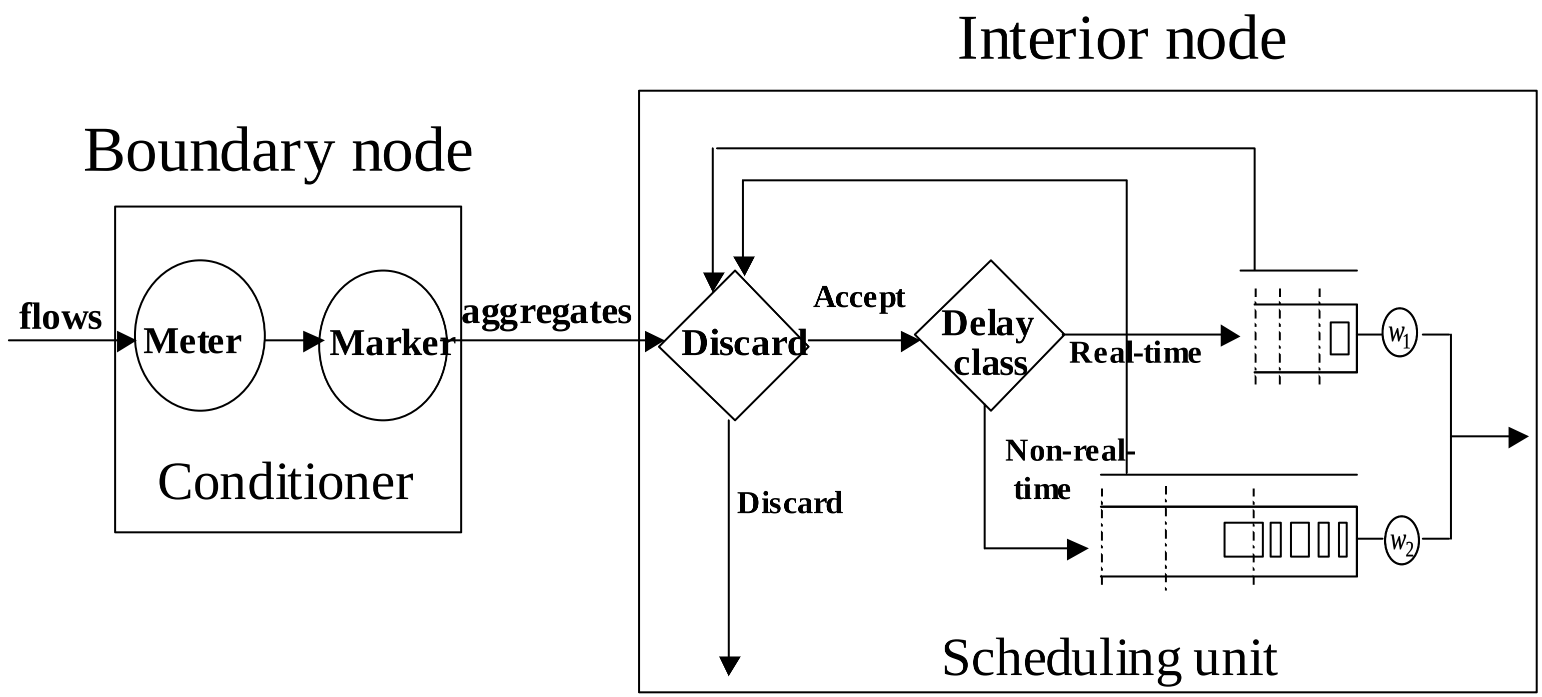


Forwarding

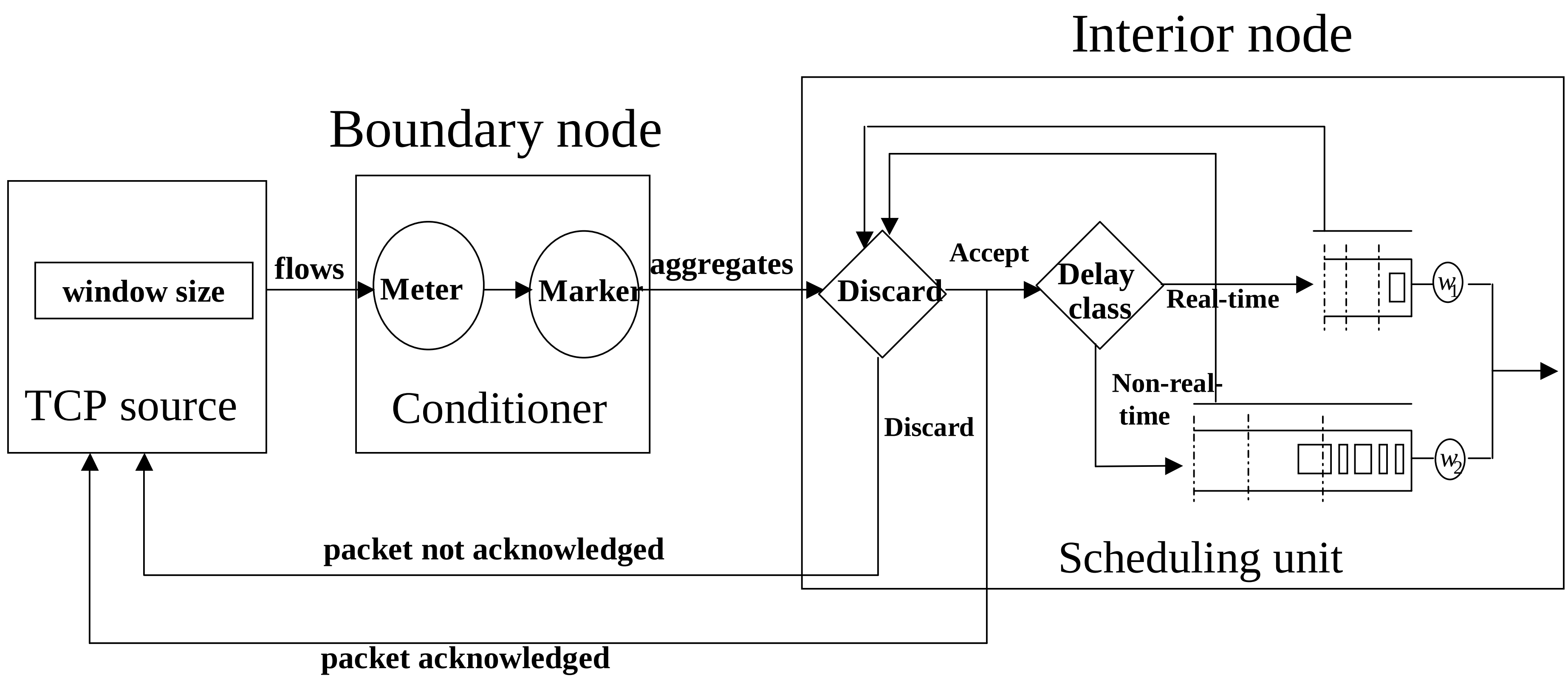
- **Priority queuing**
  - **FIFO, 2 buffers**
  - **Weights:**
  - **$w_1 = 1, w_2 = 0$**
- **Weighted fair queuing**
  - **FIFO, 2 buffers**
  - **Weights:**
  - **$w_1 < 1, w_2 > 0, w_1 + w_2 = 1$**



DiffServ mechanisms



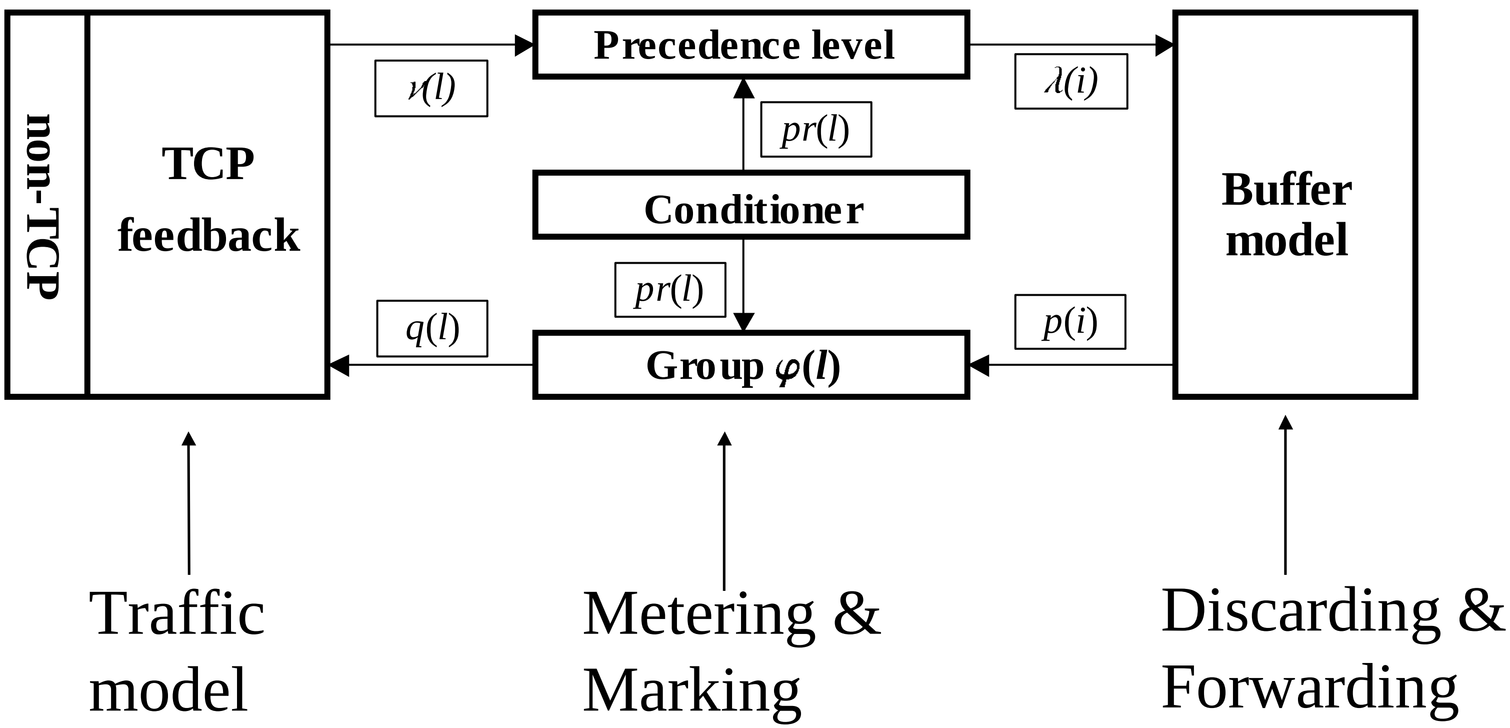
End-to-end view



Simulation results

- Marking and metering flows to *I* priorities
  - with *I*-1 cascaded token buckets can be modeled as *per packet* marking
    - only those packets exceeding a predefined threshold are marked to lower priority.
  - EWMA principle in measuring the bit rate is able to capture the flow rate and the resulting marking is *per flow*
    - all packets of the flow are marked to the same precedence level when the measured bit rate of a flow exceeds the predefined threshold.
- Time parameters α and c have to be on time scale of RTT,
  - for differentiation to occur.

Analytical Model



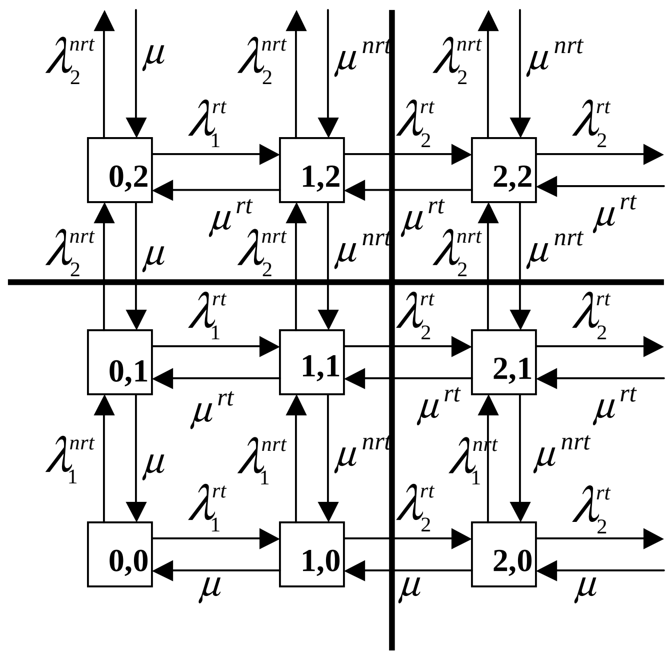
Analytical model

fixed point approach

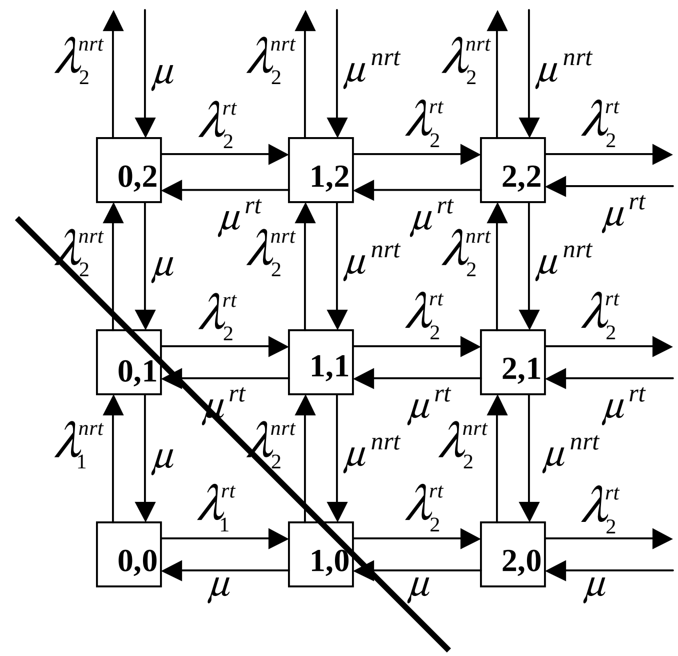
Buffer model

- Two buffers
  - one for each delay class: rt and nrt
  - Poisson arrivals
  - discarding: state dependent arrivals
  - minimum weights in dividing capacity between buffers
  - If holding times exponentially distributed
    - steady state probabilities $\pi_{j,k}$ solved numericallly
    - $p^{nrt}(i)$ and $p^{rt}(i)$ numerically

Two buffer model



independent discarding



dependent discarding

Analytical model

fixed point approach

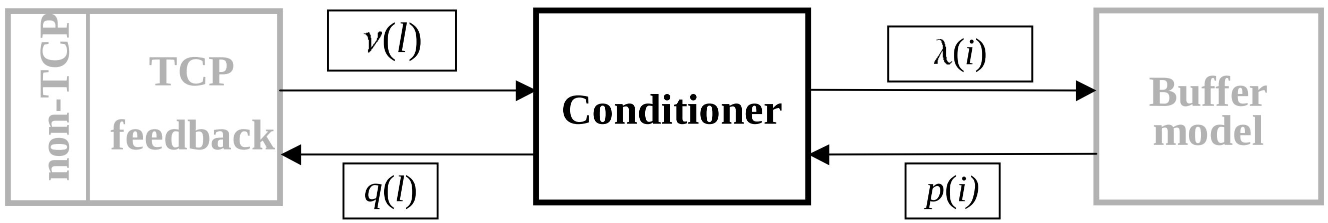
TCP feedback model

- large delay bandwdith product
- Congestion avoidance
  - Equilibrium rate for flow in group $l \in \mathcal{L}^{nrt}$

$$\nu(l) = \frac{1}{RTT}\sqrt{2\frac{1-q(l)}{q(l)}}, l \in \mathcal{L}^{nrt}$$

Analytical model

fixed point approach



Conditioner model

- Priority of user
  - Each flow in group *l* has packet arrival intensity ν(*l*)
  - priority

$$pr(l) = \max\left[\min\left[\left\lfloor I/2 + 0.5 - \frac{\ln\left(\nu(l)/\varphi(l)\right)}{\ln(2)}\right\rfloor, 1\right], I\right]$$

  - Thresholds for marking to priority level *i* are

$$t(l,0) = \infty$$

$$t(l,i) = \varphi(l) \cdot 2^{I/2-i-0.5}, i = 1,\ldots, I-1$$

$$t(l,I) = 0$$

Conditioner model

- Aggregate arrival intensities for priority class i
  - per flow marking

$$\lambda^m(i) = \sum_{l \in \mathcal{L}^m : pr(l)=i} n_l \nu(l)$$

$m = rt$ or nrt

  - per packet marking

$$\lambda^m(i) = \sum_{l \in \mathcal{L}^m : pr(l) \le i} n_l \left[\min(\nu(l), t(l, i-1)) - \min(\nu(l), t(l, i)) \right]$$

Conditioner model

- Flows according to group *l* instead of priority class
- Loss probability expereinced by flows in group *l* $q(l)$
  - per flow marking

$$q(l) = p^m(pr(l)), l \in \mathcal{L}^m$$

$m = rt$ or nrt

  - per packet marking

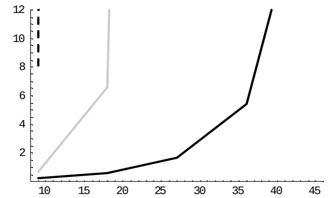
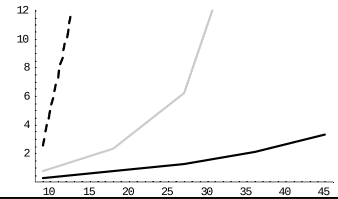
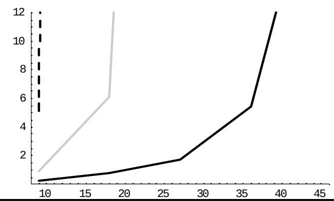
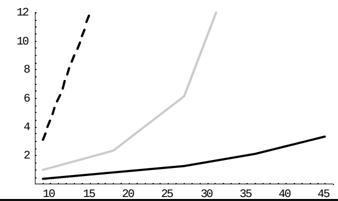
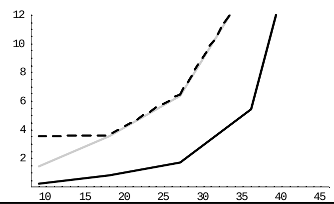
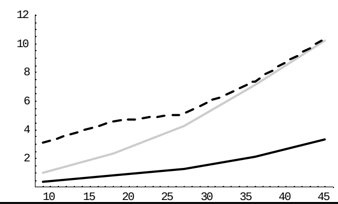
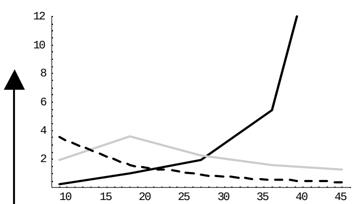
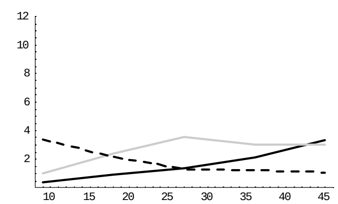
$$q(l) = \sum_{j=1}^{I} p^m(j) \left[\frac{\min(\nu(l), t(l, j-1)) - \min(\nu(l), t(l, j))}{\nu(l)} \right], l \in \mathcal{L}^m$$

Numerical results

- Results for the two buffer case ($\mu = 1$)
- Two user groups $L = 2$
  - with different NBRs (0.04, 0.08)
  - group $l = 1$, send elastic TCP flows
  - group $l = 2$, send streaming non-TCP flows
- Three priority levels $I = 3$

Throughputs

- RTT = 1000/μ, K_{nrt} = 39, K_{rt} = 13
- Ratio $[\nu(2)(1-q(2))]/[\nu(1)(1-q(1))]$ between throughputs

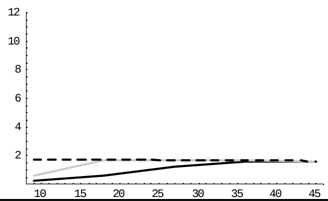
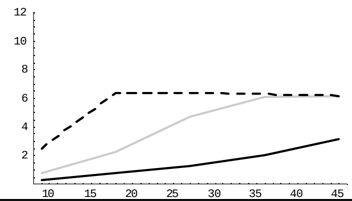
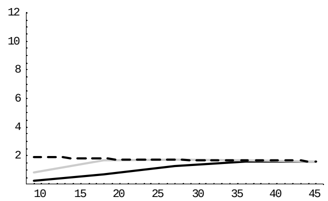
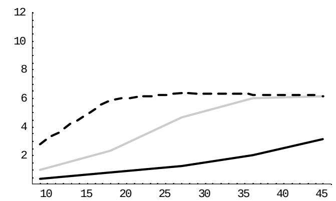
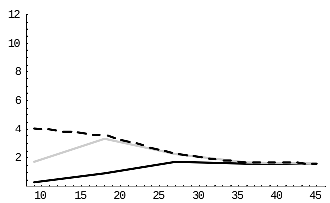
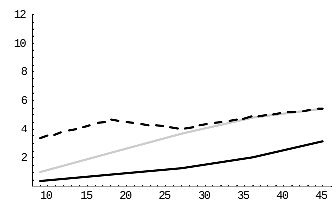
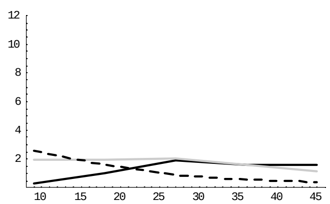
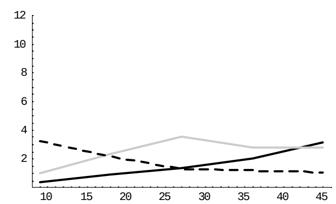
	33% TCP flows	66% TCP flows
no differentiation		
per packet marking/ per flow marking independent discarding		
per packet marking dependent discarding		
per flow marking dependent discarding		

throughput

# of all users

$w_1 = 1$
$w_2 = 0$

dashed: $i = 1$
$\nu(2) = 0.16$
gray: $i = 2$
$\nu(2) = 0.079$
black: $i = 3$
$\nu(2) = 0.039$

	33% TCP flows	66% TCP flows
no differentiation		
per packet marking/ per flow marking independent discarding		
per packet marking dependent discarding		
per flow marking dependent discarding		

throughput

# of all users

$w_1 = 0.75$
$w_2 = 0.25$

dashed: $i = 1$
$\nu(2) = 0.16$
gray: $i = 2$
$\nu(2) = 0.079$
black: $i = 3$
$\nu(2) = 0.039$

Conclusions

- Independent discarding
  – Regardless of marking same as no differentiation
- Dependent discarding + per flow marking = SIMA
  – Gives incentive for ALL flows to adjust sending rate according to the state of the network
  – Promotes TCP friendliness
- Weights
  – Give some upper bound to ratio of bandwdith
  – But not according to ratio of weights