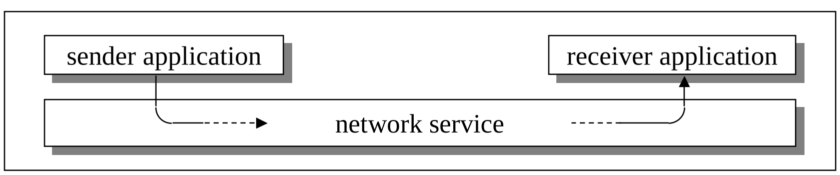
Real-Time Communication

- Integrated Services: Integration of variety of services with different requirements (real-time and non-real-time)
- Traffic (workload) characterization
- Scheduling mechanisms
- Admission control / Access control (policing)
- Deterministic *vs.* stochastic analysis
  - Traffic characterization
  - Performance guarantees
- Integration with other protocols
  - ATM
  - TCP

Providing Real-Time Guarantees



traffic specification

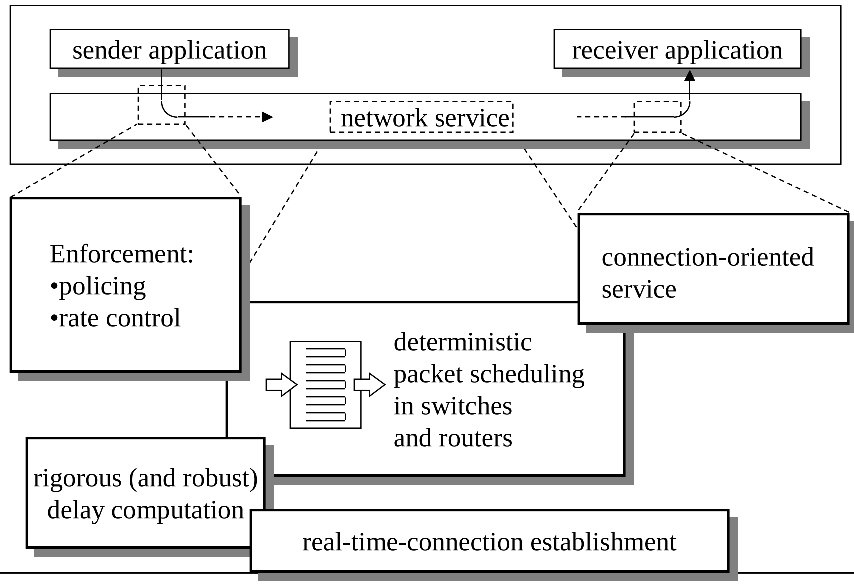
- **packet sizes**
- **packet inter-arrival times**
- **general traffic descriptors**

performance requirements

- **delay**
- **bandwidth**
- jitter
- packet loss

As long as the traffic generated by the sender
does not exceed the specified bounds,
the network service will guarantee the required performance.

Real-Time Guarantees: Mechanisms



Traffic Models

Deterministic:

1. Periodic model: *(e, p)*
2. Defered Server, Sporadic Server model: (e_S, p_S)
3. (σ, ρ) model [Cruz]
4. Leaky bucket model [Turner, ...]: (β, ρ)
5. $(x_{min}, x_{ave}, I, s_{max})$ model [Ferrari & Verma]
6. D-BIND model (Deterministic Bounding Interval Length Dependent) [Knightly & Zhang]
7. Γ-functions [Zhao]

Probabilistic:

1. S-BIND model (Stochastic Bounding Interval) [Knightly]
2. Markov-Modulated Poisson Processes

Traffic Bounding Function *b(.)*

- Let *b(.)* be a monotonicaly increasing function.
- *b(.)* is a deterministic traffic constraint function of a connection if during any interval of length *I*, the number of bits arriving during the interval is no greater than *b(I)*.

- Let *A[t1,t2]* be the number of packets arriving during interval *[t1,t2]*. Then, *b(.)* is a traffic constraint function if

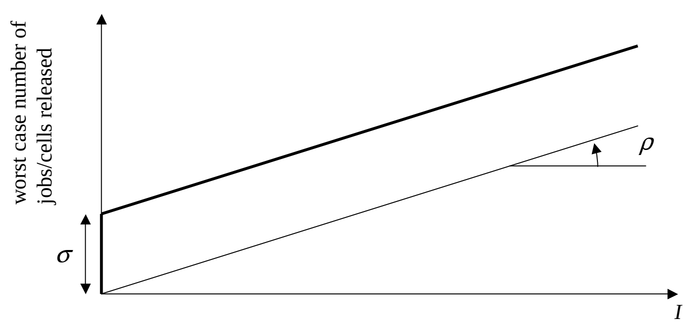
$$A[s, s+I] \leq b(I), \forall s, I > 0$$

- Each model defines inherently a traffic constraint function.
- The accuracy of models can be compared by comparing their constraint functions.

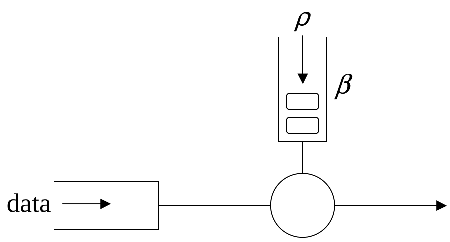
Cruz' (σ, ρ) Model

- If the traffic is fed to a server that works at rate ρ while there is work to be done, the size of the backlog will never be larger than σ.
- IOW: The number of jobs/cells released during any interval *I* does not exceed $\rho I + \sigma$.
- Graphical representation:

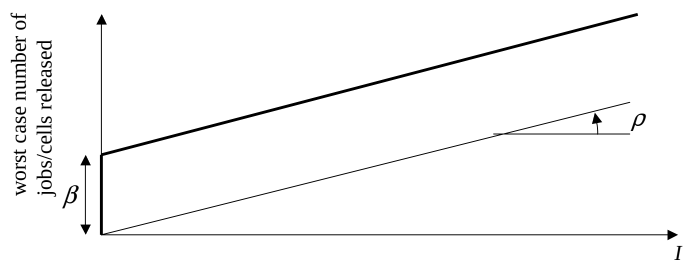
$$b^{(\sigma,\rho)}(I) = \sigma + \rho I$$



The Leaky Bucket Model

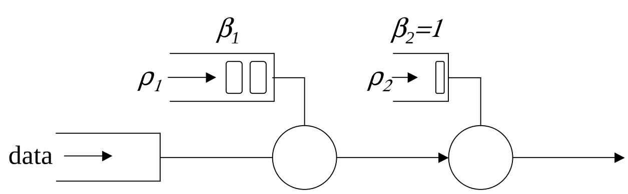


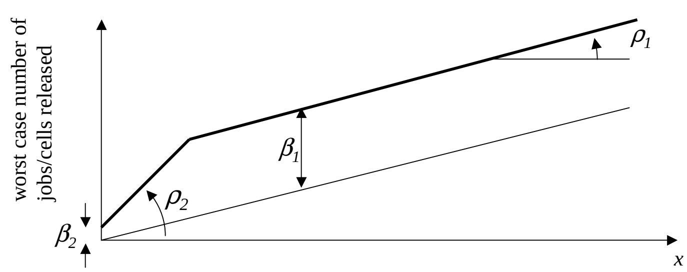
- Implementation:
  - Maintain counter for each traffic stream.
  - Increment counter at rate ρ, to maximum of β.
  - Each time a packet is offered, the counter is checked to be > 0.
  - If so, decrement counter and forward packet; otherwise drop packet.



Concatenating Leaky Buckets

- What about limiting the maximum cell rate?

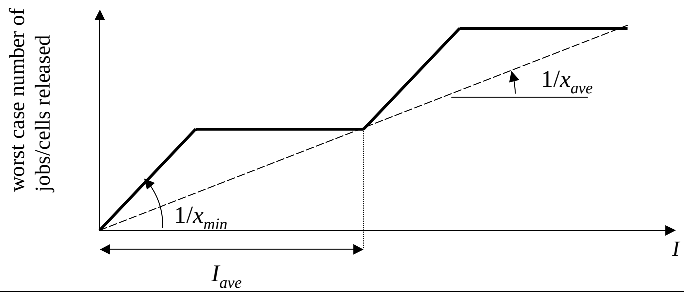




(x_{min}, x_{ave}, I_{ave}, s_{max}) model [Ferrari & Verma]

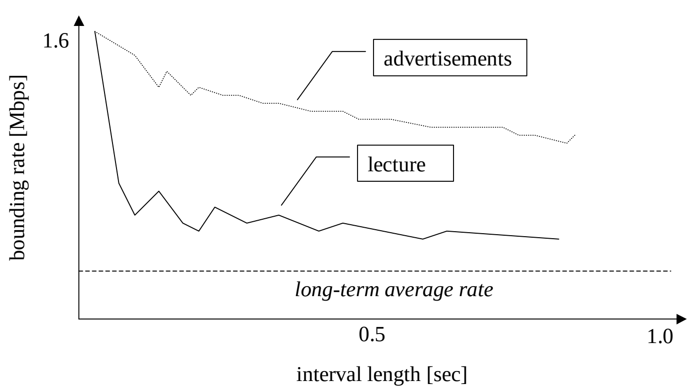
- x_{min} : minimum packet interarrival time
- x_{ave} : average packet interarrival time
- I_{ave} : averaging interval length
- s_{max} : maximum packet length

$$b(x_{\min}, x_{ave}, I_{ave}, s_{\max})(I) = \left(\min\left(\left\lceil \frac{t \bmod I_{ave}}{x_{\min}} \right\rceil, \left\lceil \frac{I_{ave}}{x_{ave}} \right\rceil \right) + \frac{I}{I_{ave}} \right) s_{\max}$$



D-BIND [Knightly & Zhang]

- Other models do not accurately describe burstiness.
- Rate-interval representation:



- Model traffic by multiple rate-interval pairs: (R_k, I_k), where rate R_k is the worst-case rate over every interval of length I_k.

D-BIND (2)

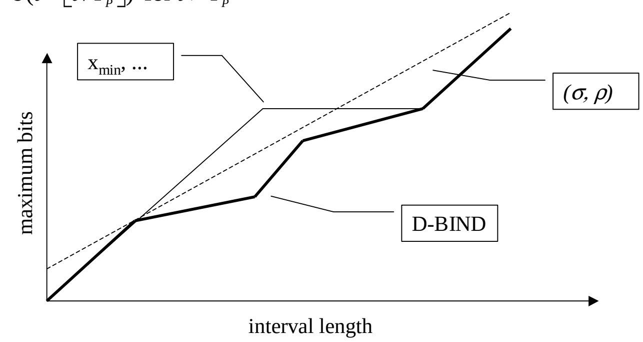
- Constraint function for D-BIND model with *P* rate-interval pairs:

$$b(t) = \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}}(t - I_k) + R_k I_k, \qquad I_{k-1} \le t \le I_k$$

$$b(0) = 0$$

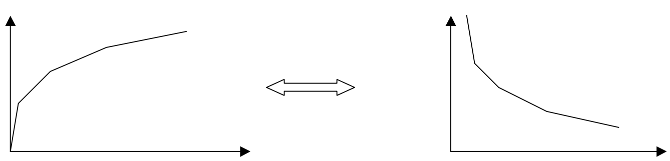
$$b(t) = b(t - \lfloor t / I_P \rfloor) \text{ for } t > I_P$$

- Comparison:

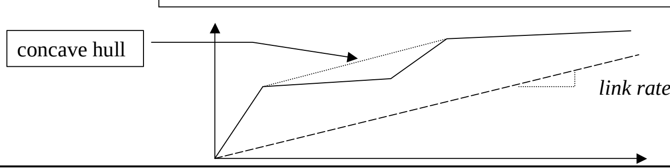


Policing for the D-BIND Model

- Lemma: If *b(t)* is piece-wise linear <u>concave</u>, then R_k is strictly decreasing with increasing I_k.



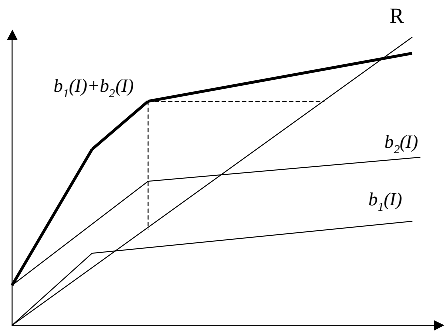
- Lemma: If a piece-wise linear constraint function *b(t)* with *P* linear segments is <u>concave</u>, then the source may be fully policed with a cascade of *P* leaky buckets.



Delay Computation: Overview

- Delay computation for FIFO server with deterministically constraint input traffic:

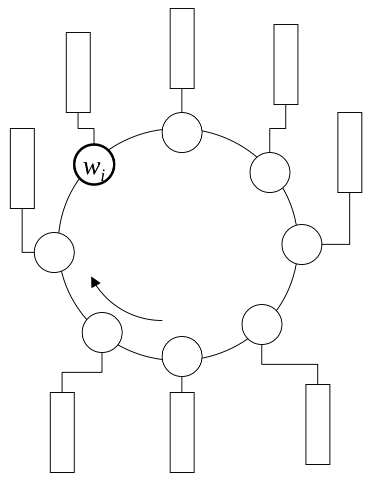
$$d = \max_{I>0}\left\{\sum_i b_i(I) - RI\right\} / R$$



Switch Scheduling

- Work-conserving (greedy) *vs.* non-work-conserving (non-greedy) mechanisms.
- Rate-allocating disciplines: Allow packets to be served at higher rates than the guaranteed rate.
- Rate-controlled disciplines: Ensures each connection the guaraneed rate, but does not allow packets to be served above guaranteed rate.
- Priority-based scheduling:
  - fair queueing
  - virtual clock
  - earliest due date (EDD)
  - rate-controlled static priority (RCSP)
- Wheighted Round-Robin scheduling:
  - WRR

Bit-by-Bit Weighted Round-Robin



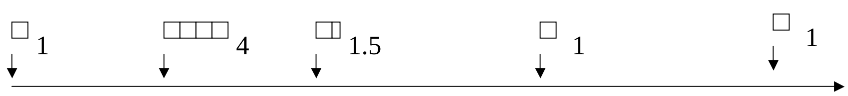
- bit-by-bit round robin
- each connection is given a weight
- each queue served in FIFO order

Fair Queueing [Demers, Keshav, Shenker]

- Emulate Bit-by-Bit Round Robin by prioritizing packets.
- Prioritize packets on basis of their finish time f_j:
  - a_j: arrival time of ${}_j$-th packet
  - e_j: length of packet
  - f_j: finish time
  - BW: allocated fraction of link bandwidth

$$f_j = \max(f_{j-1}, a_j) + e_j / BW$$

- Example:



- Complications:
  - What if connections dynamically change?

Virtual Clock Algorithm [L.Zhang]

- Emulate time-division multiplex (TDM) mechanism
- However:
  - TDM: when some connections idle, the slots assigned are idle
  - VC: idle slots are deleted from TDM frames

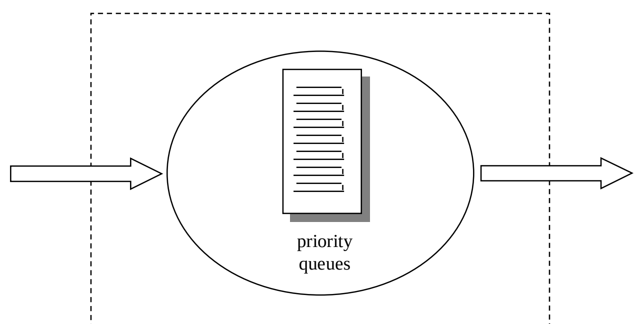
- auxiliary virtual clock ($auxVC_j$): finish time of *j*-th packet.
- virtual tick ($Vtick_j$) :time to complete transmission of ready *j*-th packet.

  $Vtick_j = e_j/BW$

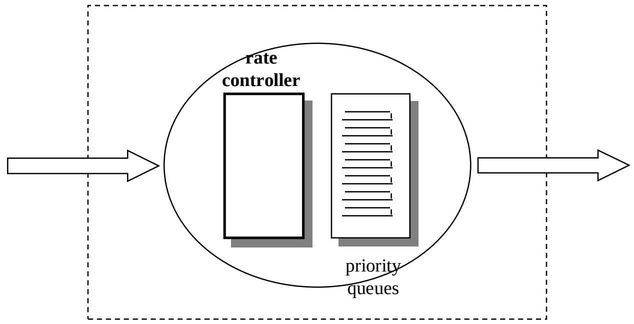
- Replace f_j by $Vtick_j$: VC becomes identical to WFQ algorithm!

- Will analyze delay analysis later.

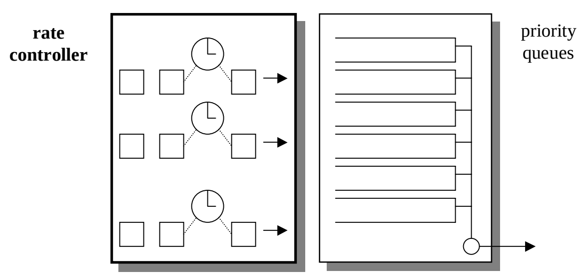
Rate-Controlled Static Priority (RCSP) [Zhang&Ferrari]



RCSP (2)



Traffic Regulation in RCSP



- Hold packets in regulator to guarantee minimum inter-packet arrival time.

  $r_{i,j} = max(a_{i,j}, r_{i,j-1}+p_i)$
- Implementation: buffer and timers in traffic regulator.
- Buffer requirements:

$$B_{ik} = \left(\left\lceil \frac{d_{ik}}{p_i} \right\rceil + \left\lceil \frac{d_{ik-1}}{p_i} \right\rceil \right) e_i$$

Is it Necessary to Regulate?

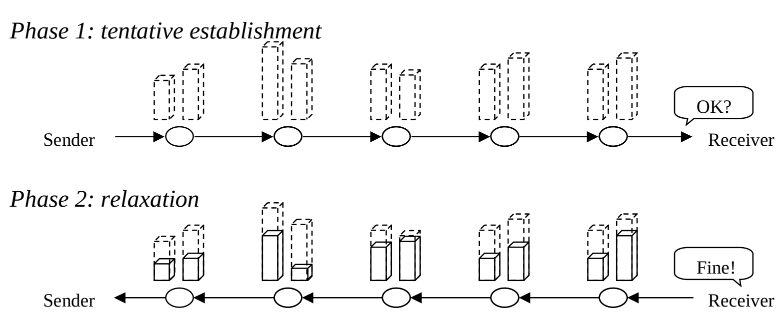
- [Liebeherr, Wrege, Ferrari, Transactions on Networking, 1995]
- Generalization of schedulability for arbitrary traffic constraint functions *A*(I)*:

Theorem: A set *N* of connections that is given by $\{A^*_j, d_j\}$ is schedulable according to a static-priority algorithm if and only if for all priorities *p*, and for all *I* >= 0 there is a *t* with $t <= d_p - s_p^{min}$ such that:

$$\forall I, \exists t \le d_p - s_p^{\min} : I + t \ge \sum_{j \in C_p} A_j * (I) - s_p^{\min} + \sum_{q=1}^{p-1} \sum_{j \in C_q} A_j * ((I+t)^-) + \max_{r>p} \{ s_r^{\max} \}$$

Earliest Due Date (EDD) [Ferrari]

- based on EDF
- delay-EDD *vs.* jitter-EDD
- works for periodic message models (single packet in period): *(pi, 1, Di)*
- partition end-to-end deadline D_i into local deadlines $D_{i,k}$ during connection establishment procedure.
- 2-Phase establishment procedure:



Delay EDD

- Upon arrival of Packet j of Connection i:
  - Determine effective arrival time: $a^e_{i,j} = max(a^e_{i,j-1} + p_i, a_{i,j})$
  - Stamp packet with local deadline: $d_{i,j} = a^e_{i,j} + D_{i,k}$
  - Process packets in EDF order.
- Delay EDD is greedy.
- Can be mapped into special case of Sporadic Server.
- Acceptance test (Δ = total density): $\Delta + 1/p_i < 1 - 1/p_{min}$
- Offered local deadline: $LD_i = \min(p_i, 1/(1-\Delta-1/p_{min}))$
- Problem with EDD: **jitter**
  - max end-to-end delay over k switches: $\sum_k D_{i,k}$
  - min end-to-end delay over k switches: k

Jitter EDD

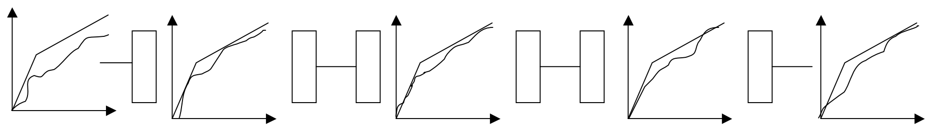
- Problem with Delay-EDD: does not control jitter. This has effect on buffer requirements.
- Jitter-EDD maintains <u>Ahead Time</u> $ah_{i,j}$, which is the difference between local relative deadline $D_{i,k-1}$ and actual delay at Switch $k-1$.
- Ahead time is stored in packet header (alternatively, we use global time synchronization)
- Upon receiving the j-th packet of Connection i with $ah_{i,j}$ at time $a_{i,j}$:
  - Calculate ready time as Switch k:

    $a^e_{i,j}=max(a^e_{i,j-1} + p_i , a_{i,j})$

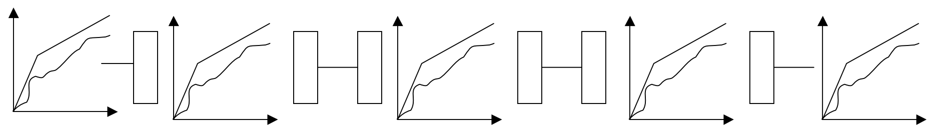
    $r_{i,j} = max(a^e_{i,j} , a_{i,j} + ah_{i,j})$
  - Stamp packet with deadline $d_{i,j}=r_{i,j}+D_{i,k}$ and process according to EDF starting from ready time $r_{i,j}$.
- Result: Regenerate traffic at each switch.

Rate Control *vs.* Jitter Control

- Rate Control



- Jitter Control



Simple EDF with Arbitrary Arrival Functions

[Liebeherr, Wrege, Ferrari: Transactions on Networking, 1995]

Theorem: A set Π of connections that is given by $\{A_i^*; d_i\}_{i \in \Pi}$ and $d_i <= d_j$ whenever $i<j$ is EDF schedulable if and only if for all $I >= d_1$:

$$I \geq \sum_{j \in \Pi} A_j^*(I - d_j) + \max_{k, d_k > I} \{ s_k^{\max} \}$$

where

$$\max_{k, d_k > I} \{ s_k^{\max} \} \equiv 0 \text{, for } I > \max_{k \in \Pi} \{ d_k \}$$

Informal "proof": A deadline violation occurs at time I if the maximum traffic arrivals with deadline before or at time I, i.e.

$$I < \sum_{j \in \Pi} A_j^*(I - d_j)$$

exceeds I.

EDF Test for Special Cases: Example (σ,ρ)

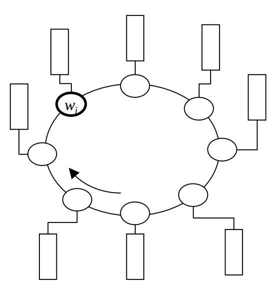
- For some traffic models, closed form expressions for the schedulability test exist.
- For (σ, ρ) traffic:

$$\begin{cases} I \geq \sum_{i=1}^{j} \sigma_i + \rho_i (I - d_i) + \max_{k>j} \{ s_k^{\max} \} \text{ for } d_j \leq I < d_{j+1} : 1 \leq j < |\Pi| \\ I \geq \sum_{i=1}^{|\Pi|} \sigma_i + \rho_i (I - d_i) \text{ for } I \geq d_{|\Pi|} \end{cases}$$

- A closed form for the delay can be given as follows:

$$d_j = \frac{\sigma_j + \sum_{i=1}^{j-1} (\sigma_i - \rho_i d_i) + \max_{k>j} \{ s_k^{\max} \}}{1 - \sum_{i=1}^{j-1} \rho_i}$$

Weighted Round Robin (WRR)



- Each connection i is assigned a weight w_i, i.e., it is allocated w_i slots during each round.
- Slot: time to transmit maximum-sized packet.

- Traffic model:
  - periodic (p_i, e_i, D_i)
  - variable bit rate models possible
- Realizations:
  - greedy WRR
  - Stop-and-Go (SG)
  - Hierarchical Round Robin (HRR)

Throughput and Delay Guarantees

- Each connection i is guaranteed w_i slots in each rounds.
- Round length RL : upper bound on sum of weights (design parameter)

$$\sum w_i \le RL$$

- Constraints:

1. $RL \le p_{\min}$
2. $w_i \ge \left\lceil \frac{e_i}{\lfloor p_i / RL \rfloor} \right\rceil$

- Delays:
  - at first switch: $\left\lceil \frac{e_i}{w_i} \right\rceil RL$
  - downstream: once packet passes first switch, it is immediately eligible on switches downstream -> has to wait at most RL

  => end-to-end delay through N switches:

$$W_i \le (\lceil e_i / w_i \rceil + N - 1) RL \le p_i + (N-1) RL$$

Problems with Greedy WRR

- Greedy WRR does not control jitter:

- min end-to-end delay: $e_i+(N-1)$
- max end-to-end delay: $p_i+(N-1)RL$
- jitter: $p_i-e_i+(N-1)(RL-1)$

- Buffer needed at k-th switch for connection i:

$$(1+\lceil (k-1)(RL-1)/p_i \rceil) e_i$$

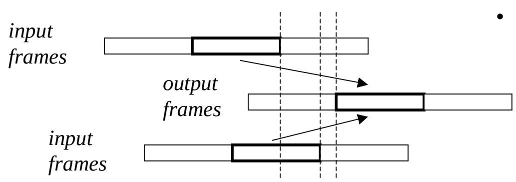
- Need traffic shaping at each switch.

Non-Greedy WRR

- Actual length of rounds in greedy WRR varies with amount of traffic at switch.
- Non-greedy WRR schemes fix round length into fixed-length frames.
- Stop-and-Go [Golestani]
- Hierarchical Round Robin [Kalmanek, K., K.]

Stop & Go [Golestani, 1990]

- Frame-based: divide time in *frames* of length *RL*.
- Packet arriving during frame at input link is eligible for transmission during *next* frame on output link.



- Implementation: 2 queues per outgoing link: *input queue* and *output queue*
  - swap queues at end of each frame

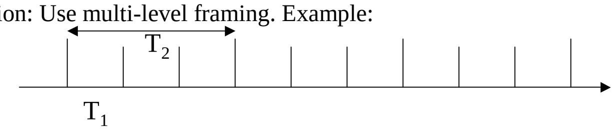
- Stop-and-Go is *not* work-conserving.
- Traffic model **[*(r, RL)* smooth traffic]**: during each frame of length *RL*, the total number of bits transmitted by source does not exceed *rRL* bits.

- **Proposition:** If the connection satisfies (r,RL) smoothness at the input of the first server, and each server ensures that packets will always go out on the next departing frame, the connection will satisfy (r,RL) smoothness at each server throughout the network.

Multi-Frame Stop-and-Go

[For example, Zhang&Knightly: "Comparison of RCSP and SG", UC-Berkeley EECS tech report TR-94-048]

- Problem with Stop-and-Go (or any other frame-based approach): **delay-bandwidth coupling**
  - Delay of packet is bounded by a multiple of frame time. This is a problem, for example for low-bandwidth, low-delay connections. (Why?)
- Solution: Use multi-level framing. Example:

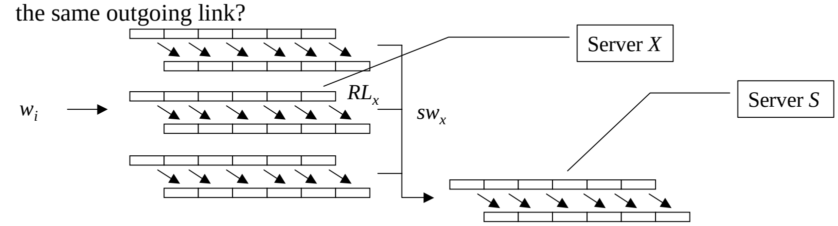


- Hierarchical framing with n levels with frame sizes $T_1, ..., T_n$, where $T_m+1=K_mT_m$ for $m = 1, ..., n-1$.
- Stop-and-Go rule for packets of level-$_p$ connection: Packets that arrived during a T_p frame will not become eligible until the start of the next T_p frame.
- Packets with smaller frame size have higher priority (non-preemptively) over packets with larger frame size.

Hierarchical Round Robin

[Kalmanek, Kanadia, Keshav, 1990]

- End-to-end delay and jitter of S&G depends on *RL* only.
- How about having multiple S&G servers, with different *RL*'s, and multiplex them on the same outgoing link?



- Server *X* is seen as periodic stream of requests by Server *S*, with
  - $e_x = sw_x$, $p_x = RL_x$, $D_x = RL_x$
  - schedule using rate-monotonic scheduler
  - Configuration time test: check whether task set $\{(sw_x,RL_x,RL_x)\}$ is schedulable.
- Admission Control Test:
  - Bandwidth test: check sum of required w_i's <= sw_x
  - Delay test: End-to-end delay: $p_i + N\ RL_x$
  - Jitter test: 2 RL_x, with buffer requirement 2 w_i