Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors

T. S. Eugene Ng, Ion Stoica, Hui Zhang

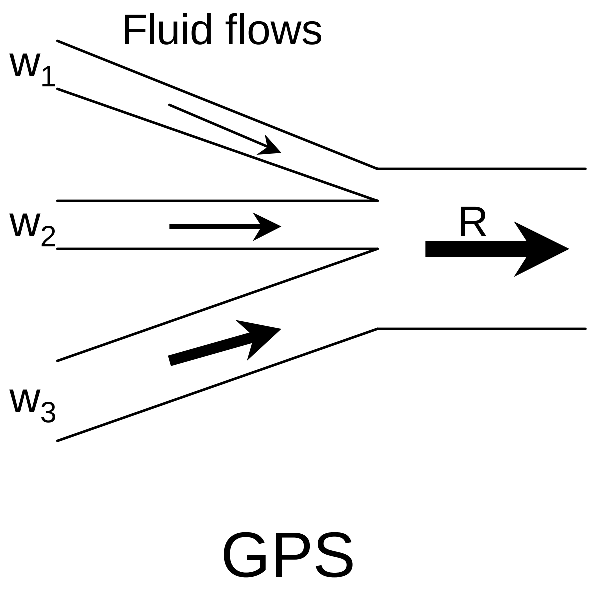
School of Computer Science

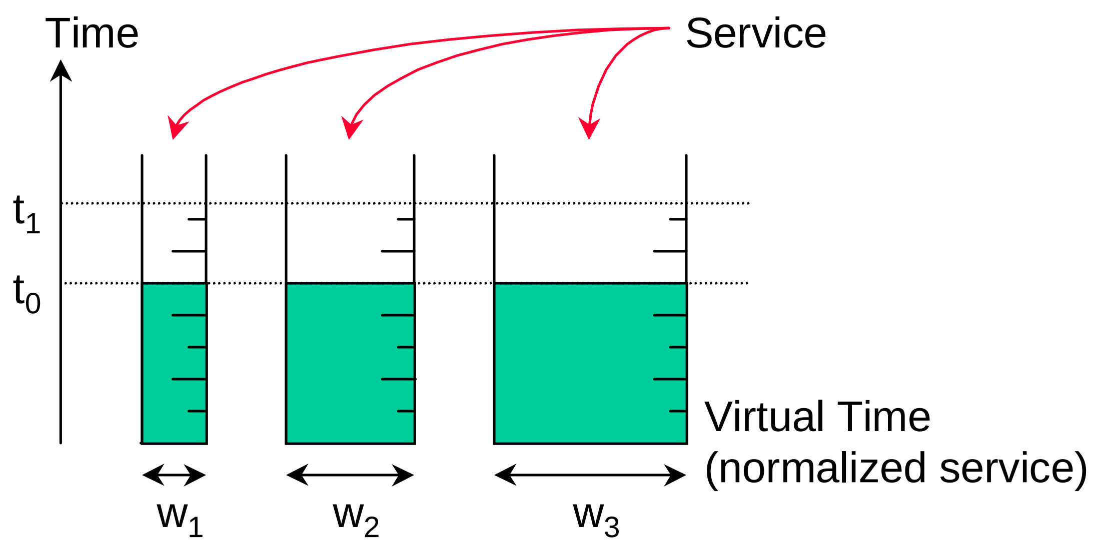
Carnegie Mellon University

Outline

- Generalized Processor Sharing (GPS) and Packet Fair Queueing (PFQ)
- Simplified wireless network model
- *Why GPS does not work well for wireless networks*
- Related work
- *Channel-condition Independent Fair (CIF) properties*
- *Achieving CIF properties -- the CIF-Q algorithm*
- Theoretical results
- Simulation results
- Conclusions

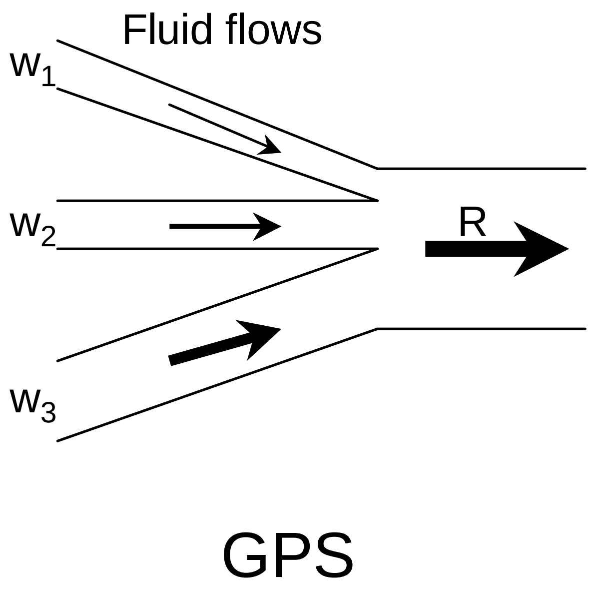
Generalized Processor Sharing (GPS)

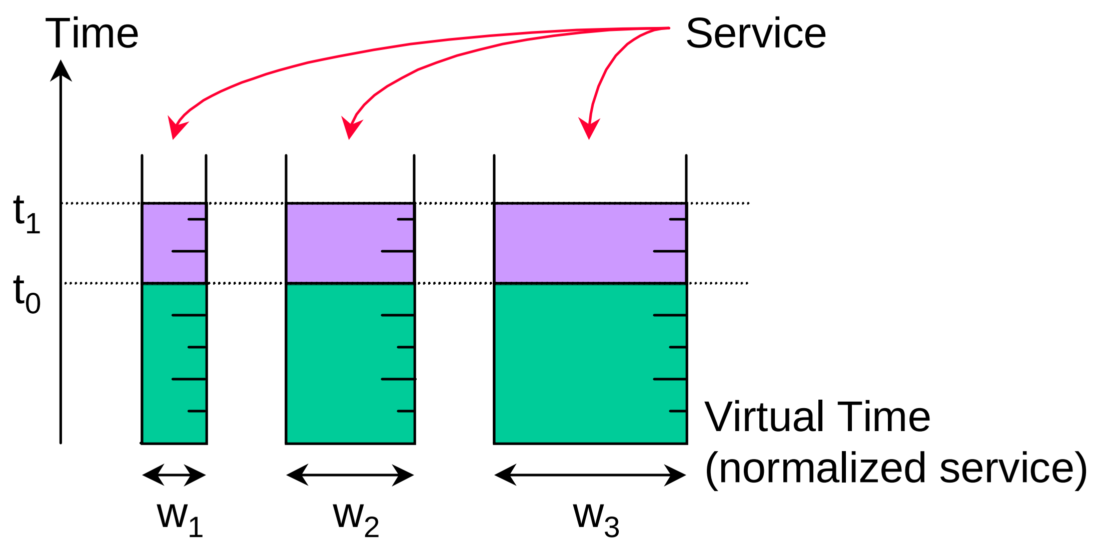




Packet Fair Queueing (PFQ) is the packet by packet approximation of GPS

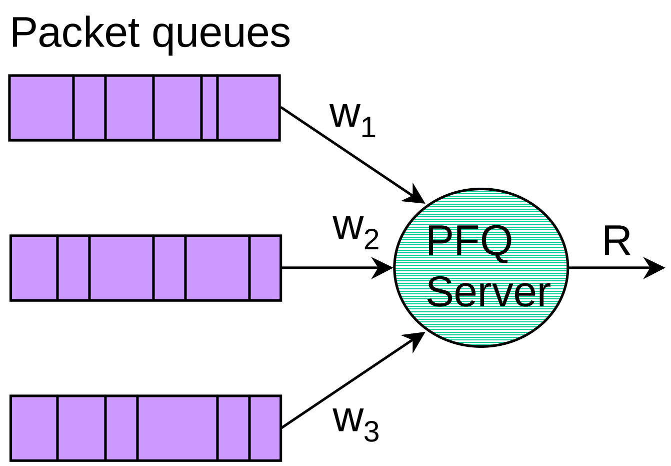
Generalized Processor Sharing (GPS)



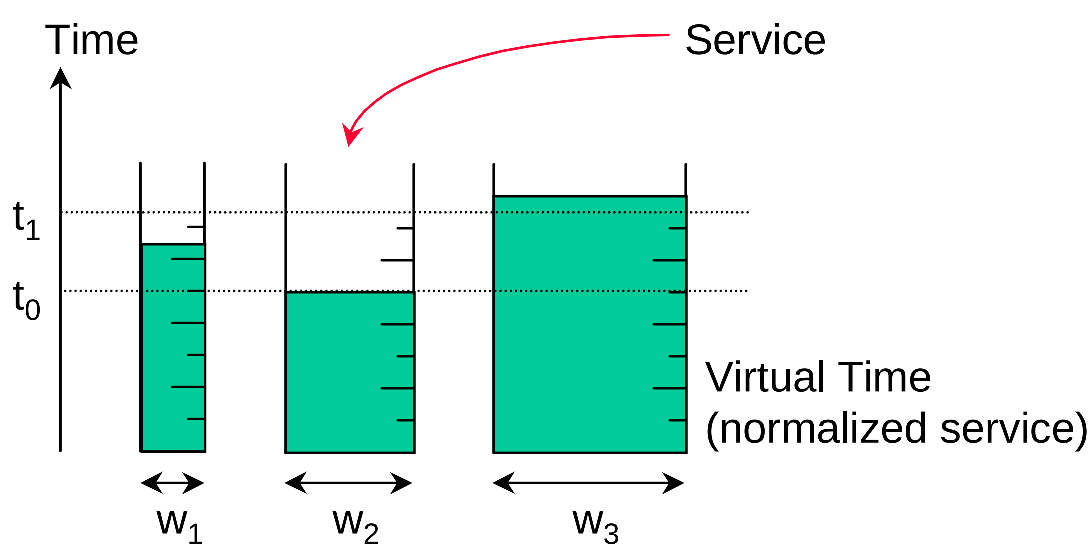


Packet Fair Queueing (PFQ) is the packet by packet approximation of GPS

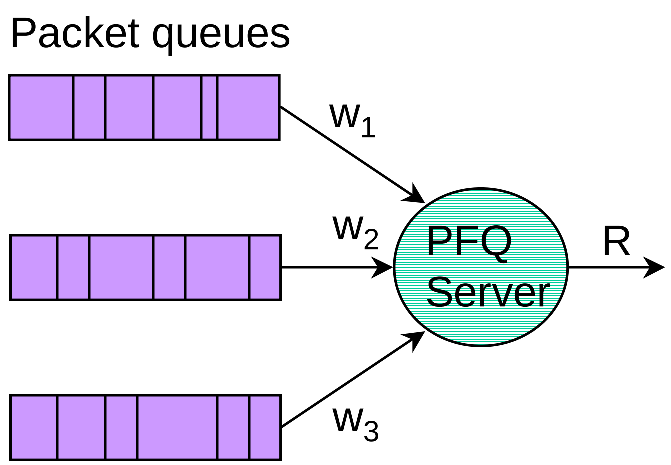
Packet Fair Queueing (PFQ)



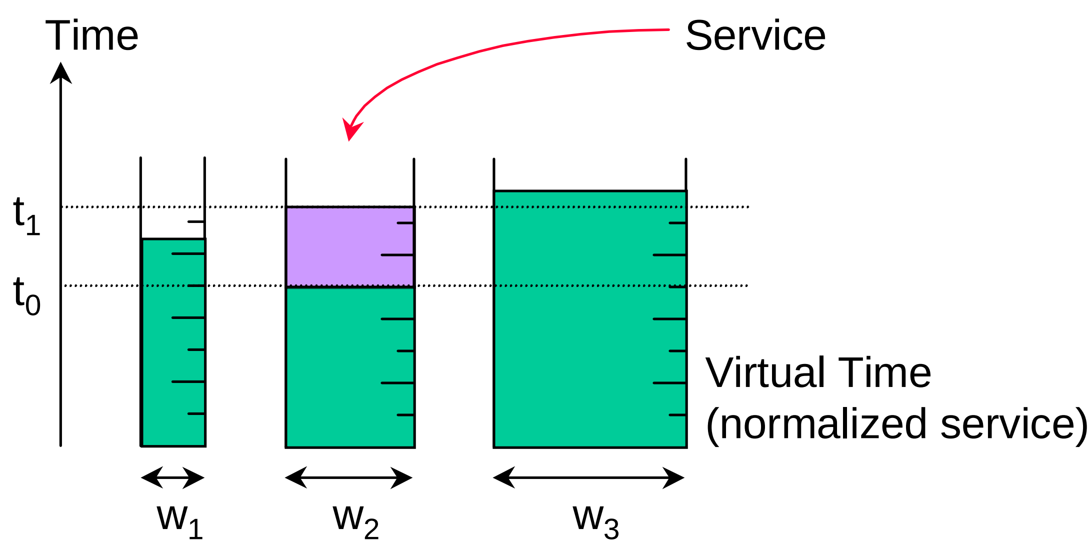
PFQ



Packet Fair Queueing (PFQ)



PFQ

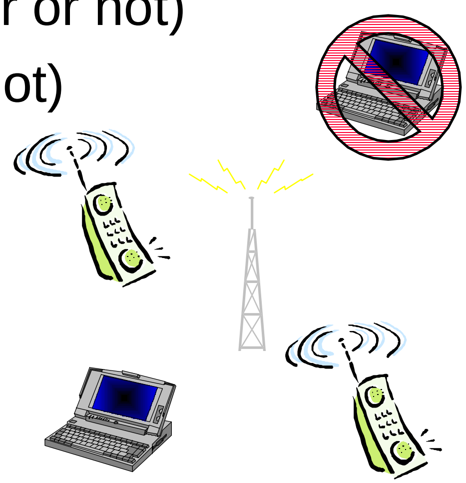


Why are GPS and PFQ desirable?

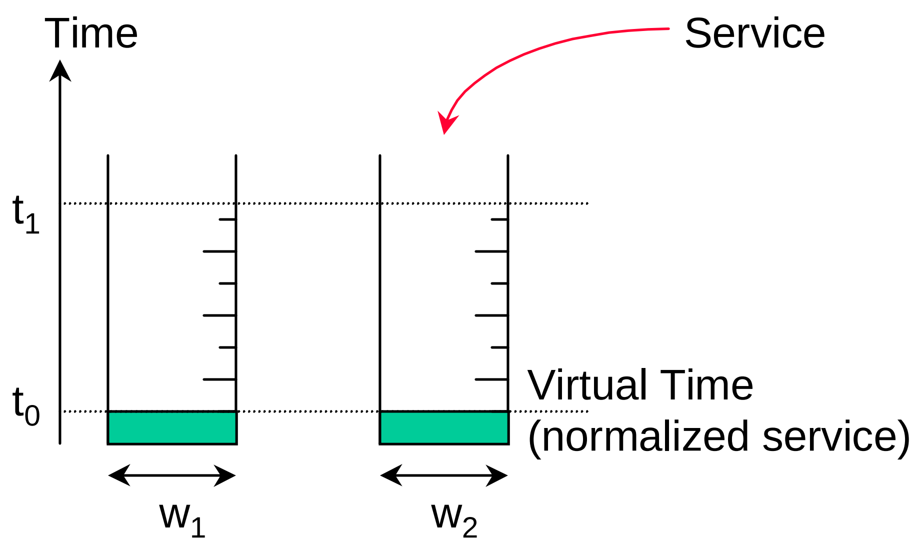
- End-to-end delay bound for guaranteed service [Parekh and Gallager '93]
- Fair allocation of bandwidth for best effort service [Demers et al. '89, Parekh and Gallager '92]
- Work-conserving for high link utilization
- Flexible for diverse QoS needs

Simplified wireless network model

- Shared-channel wireless cellular network
  - e.g. Lucent's WaveLAN
- Centralized packet scheduling at the base station
  - Coupled with media access control
- Instantaneous knowledge
  - Channel condition of each session (in error or not)
  - Status of uplink sessions (backlogged or not)
- Location-dependent channel errors
  - Good or bad

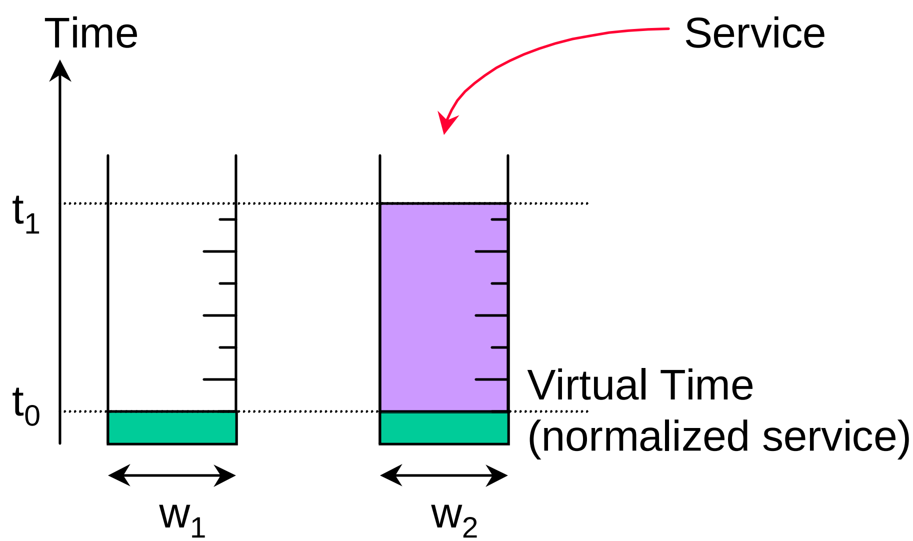


GPS with location-dependent errors



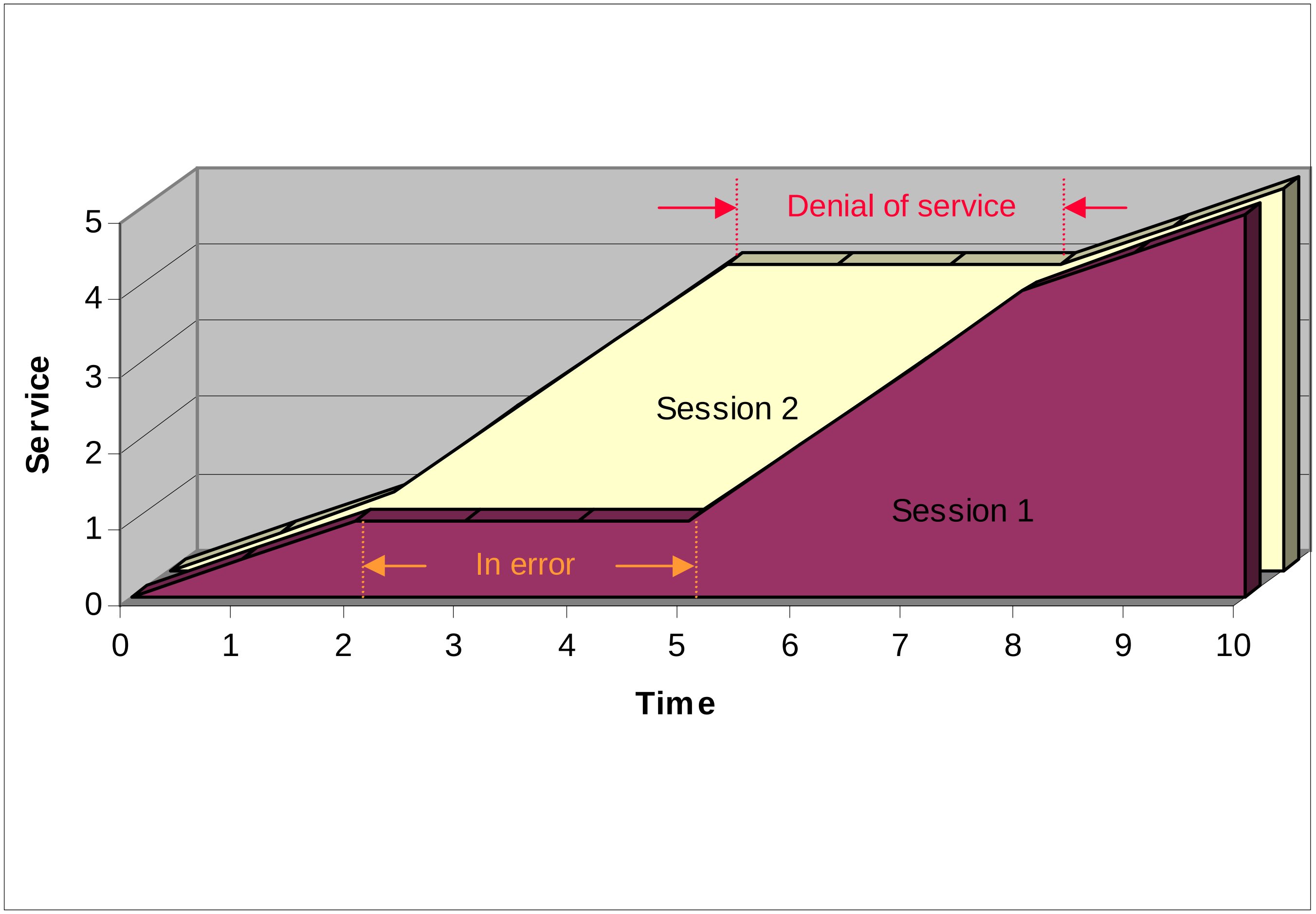
- Session 1 enters error at time t_0
- Session 1 exits error at time t_1
- What should GPS do after time t_1?
  - To compensate or not to compensate?

GPS with location-dependent errors



- Session 1 enters error at time t_0
- Session 1 exits error at time t_1
- What should GPS do after time t_1?
  - To compensate or not to compensate?

GPS with location-dependent errors



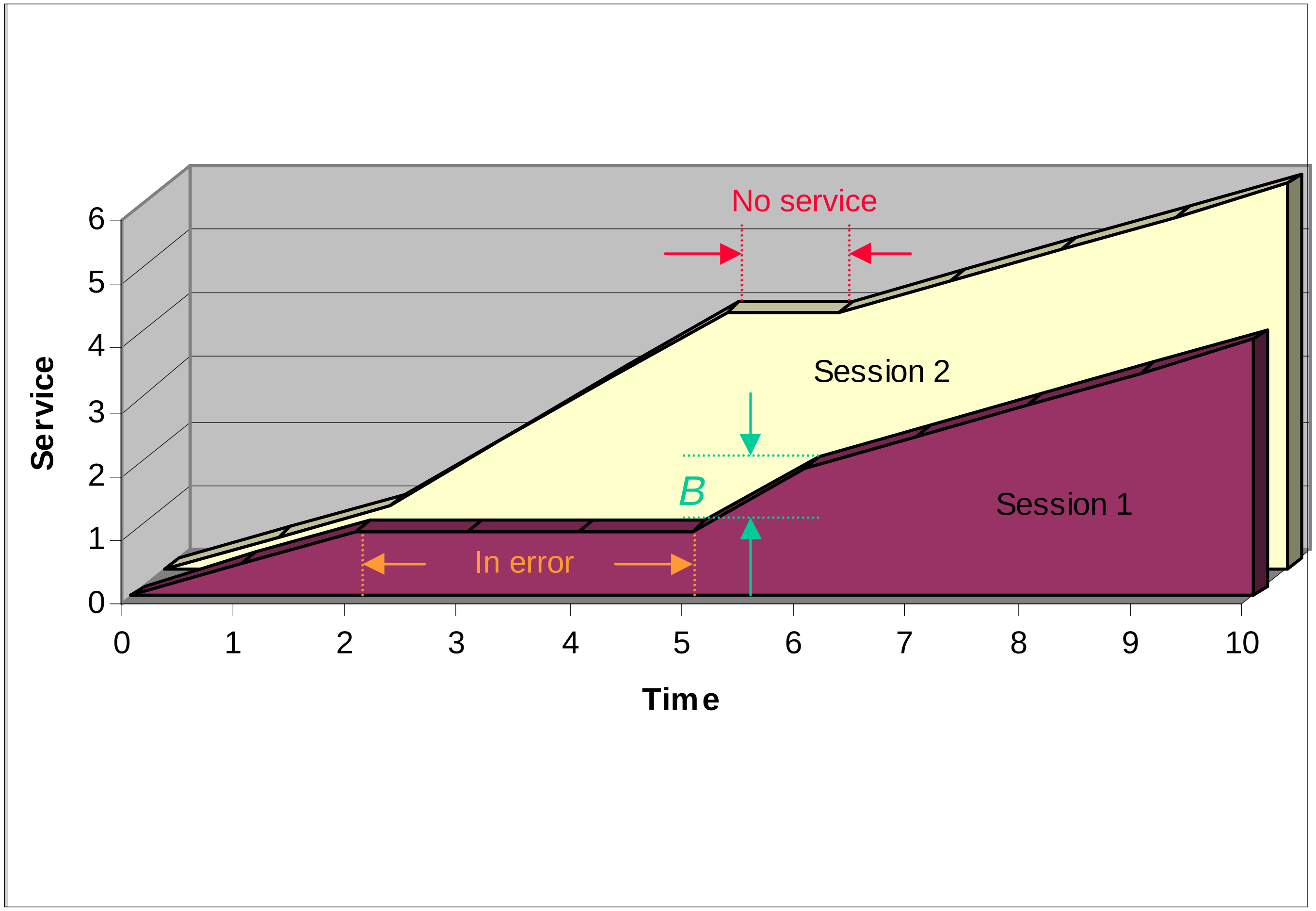
Two plausible simple solutions

- Equalize the virtual times by bumping up session 1's virtual time artificially
  - Delay bounds for error-free sessions hold
  - Error-free sessions get extra service, but...
  - No fairness
- Equalize the virtual times by serving session 1 exclusively
  - Perfect fairness, but...
  - Other sessions receive no service
  - Sessions see abrupt changes in service
  - No delay bound

Related work

- Idealized Wireless Fair Queueing (IWFQ) [Lu et al. SIGCOMM '97]
  - Control total amount of compensation, *B*
  - Provide delay bounds for error-free sessions
  - Deal with practical implementation issues
    - Channel condition detection and prediction
    - Session status detection
    - Media access control

IWFQ



No service
Session 2
B
Session 1
In error
Service
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
8
9
10
Time

Channel-condition Independent Fair (CIF)

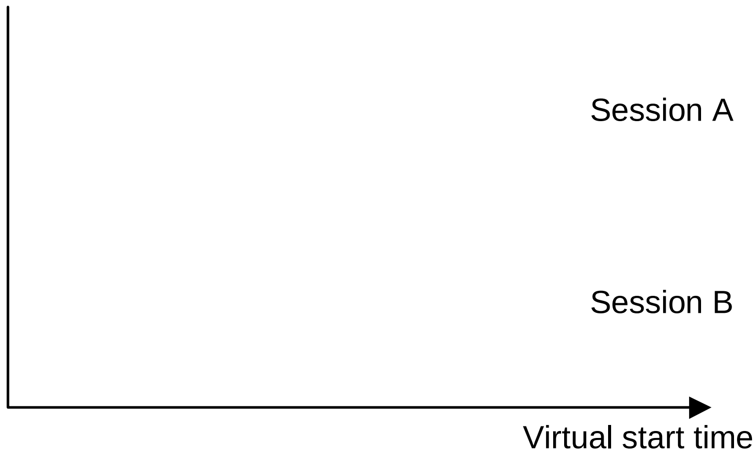
- Delay and throughput guarantees for error-free sessions
  - Independent of other sessions' error
- Long-term fairness for error sessions
  - No artificial bound
- Short-term fairness for error-free sessions
  - Similar to GPS's fairness property
- Graceful degradation for sessions that have received excess service

Achieving CIF properties -- the *Channel-condition Independent packet Fair Queueing (CIF-Q)* algorithm

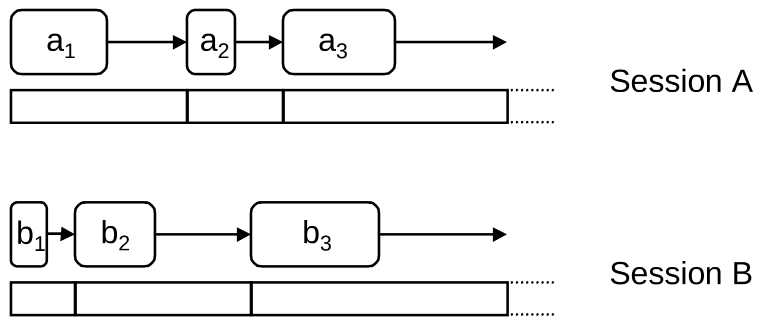
Key techniques used in CIF-Q

- Use an error-free reference system for scheduling
  - Based on SFQ [Goyal et al. SIGCOMM '96]
- Use a parameter lag_i to keep track of the difference between the real system and the reference system
- Leading sessions give back only a fraction of their service
- Leading unbacklogged sessions are not allowed to leave the active set to ensure fairness
- Use forced compensation to ensure delay bounds
- Use extra virtual times to distribute services fairly to leading and non-leading sessions

Use an error-free reference system for scheduling

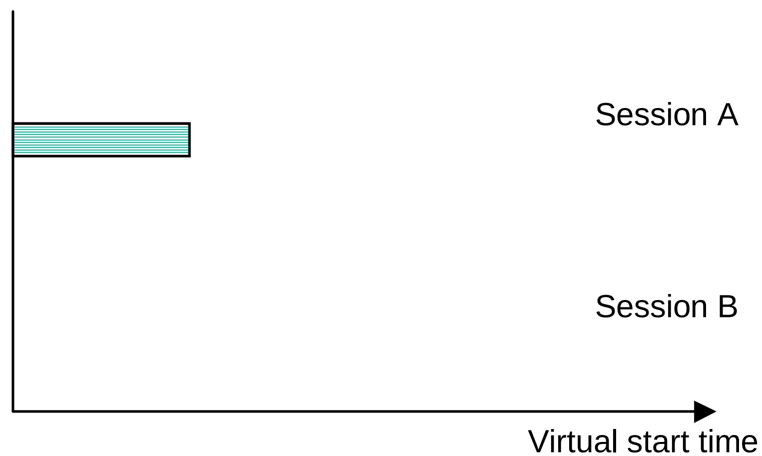


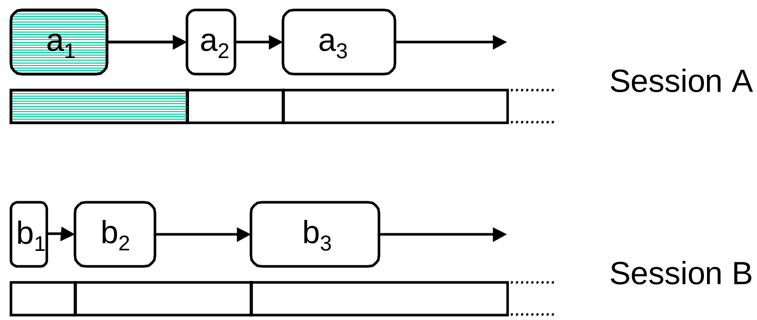
Error-free reference system S^r_{SFQ}



Real system S

Use an error-free reference system for scheduling

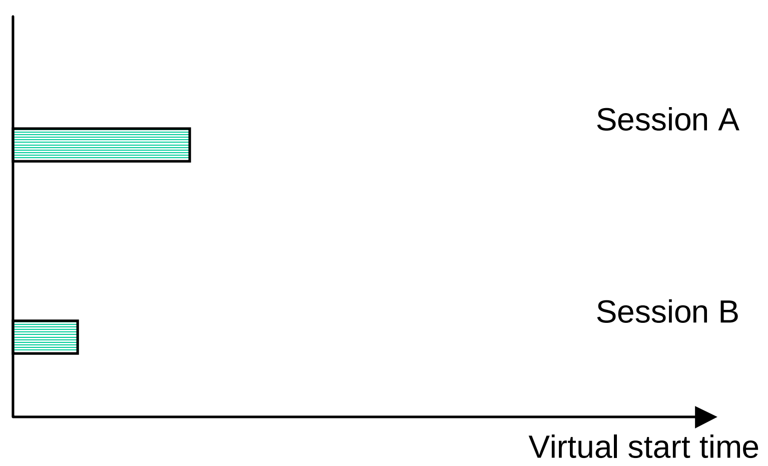




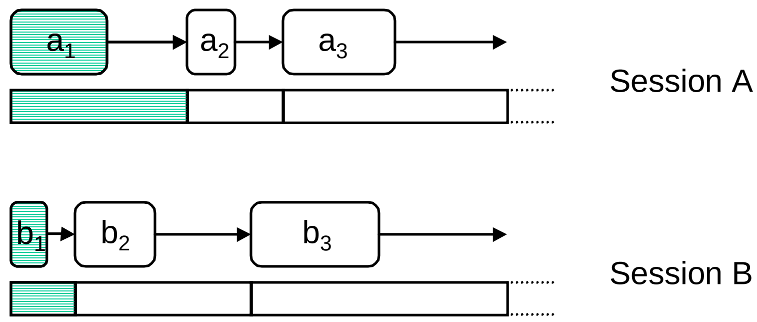
Error-free reference system S^r_{SFQ}

Real system S

Use an error-free reference system for scheduling

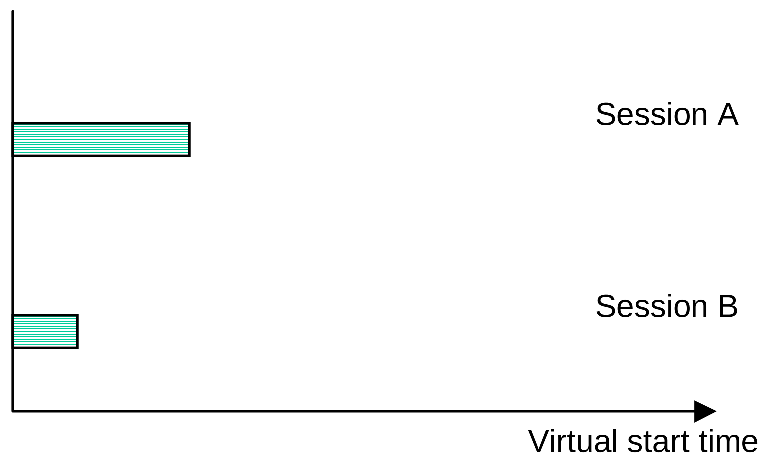


Error-free reference system S^r_{SFQ}

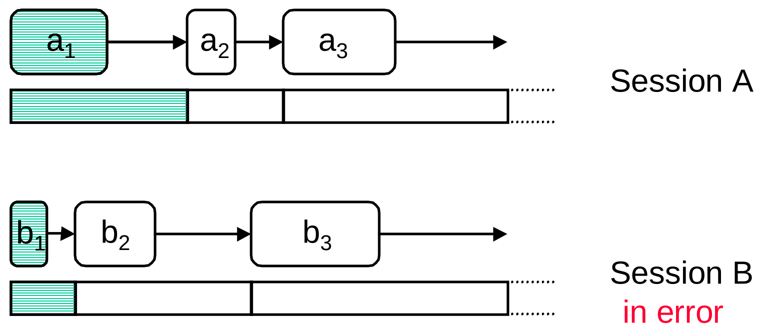


Real system S

Use an error-free reference system for scheduling

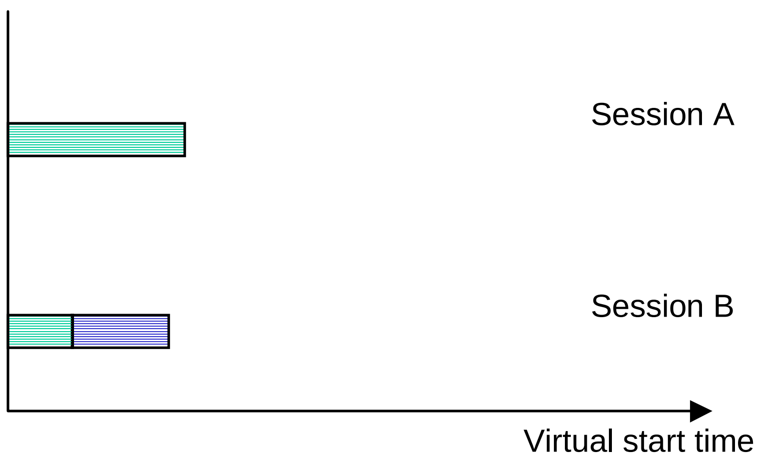


Error-free reference system S^r_{SFQ}

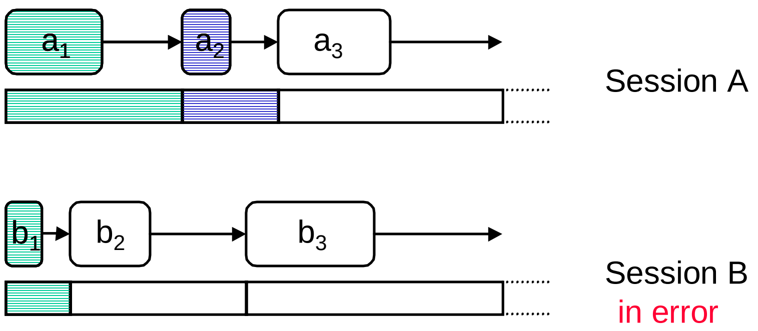


Real system S

Use an error-free reference system for scheduling

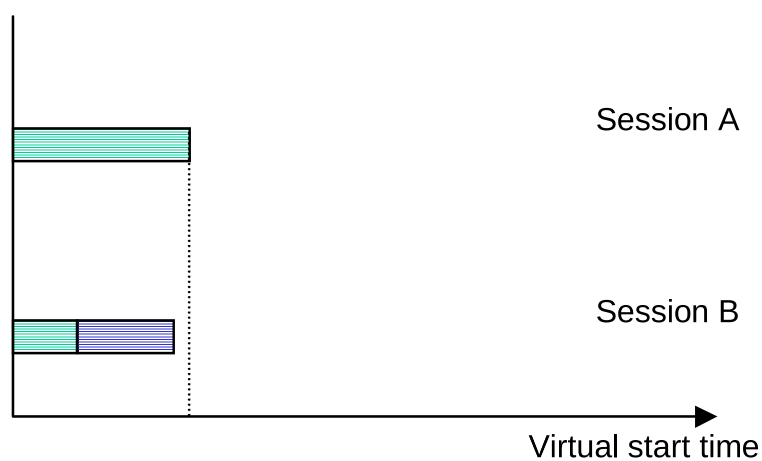


Error-free reference system S^r_{SFQ}

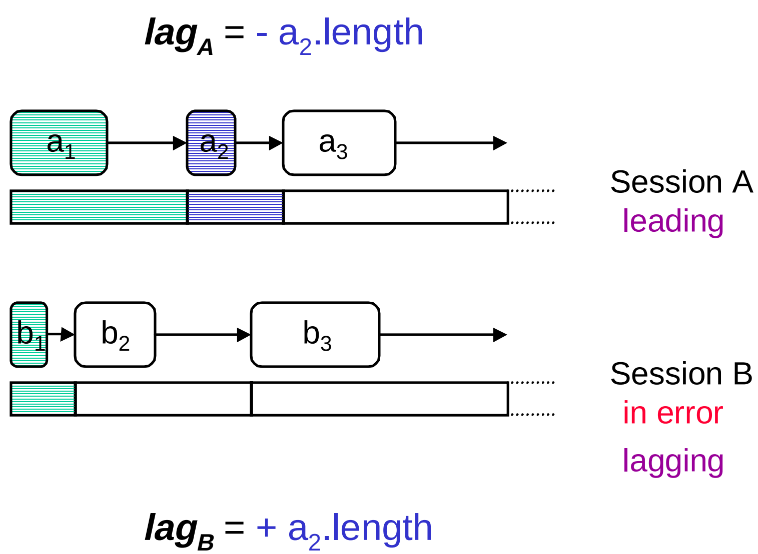


Real system S

Use lag_i to keep track of deviations

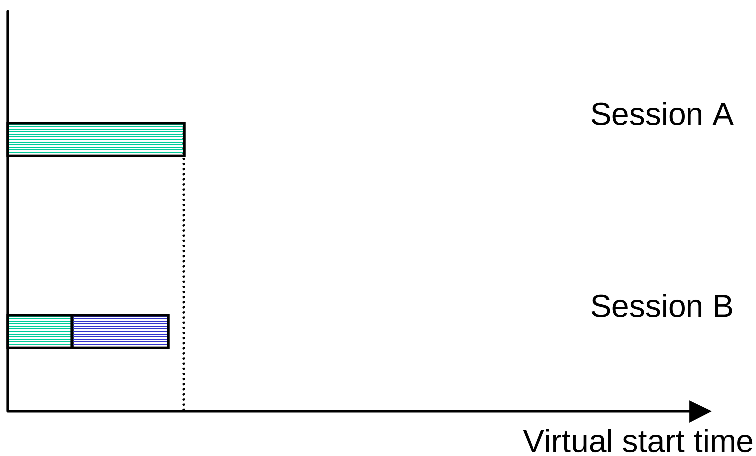


Error-free reference system S^r_{SFQ}



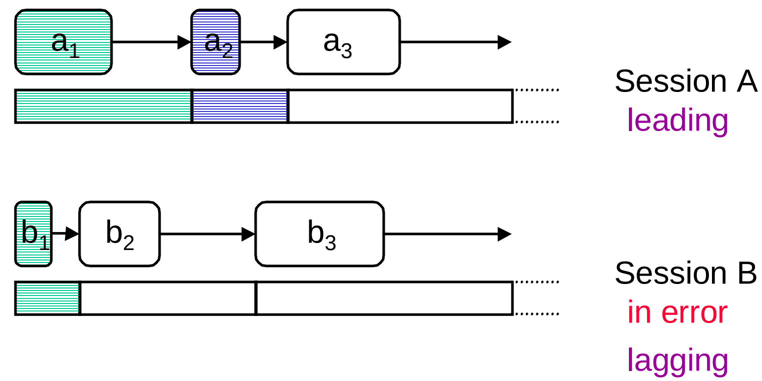
Real system S

Use lag_i to keep track of deviations



Error-free reference system S^r_{SFQ}

lag_A = - a_2.length



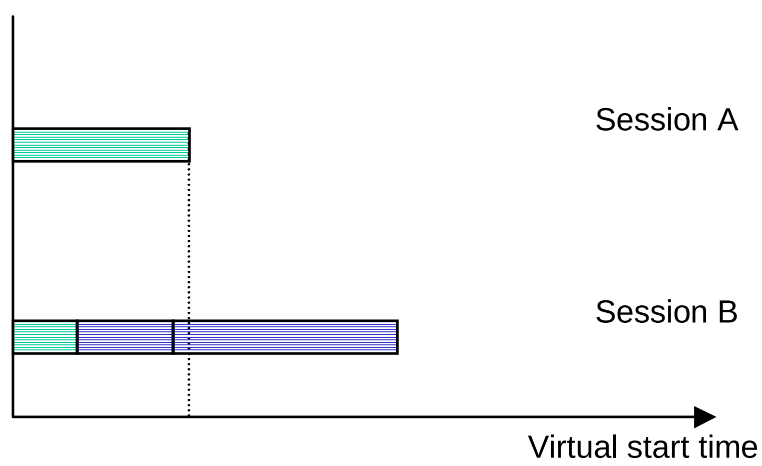
lag_B = + a_2.length

Invariant: $\sum_{i \in A} lag_i = 0$

A is the set of active sessions

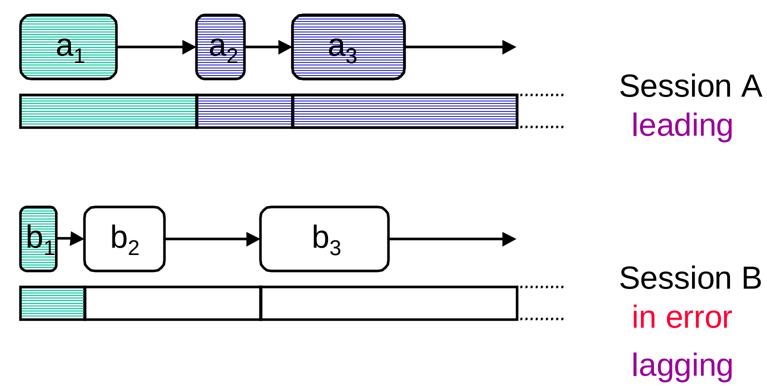
Real system S

Use lag_i to keep track of deviations



Error-free reference system S^r_{SFQ}

$\boldsymbol{lag_A}$ = - a_2.length



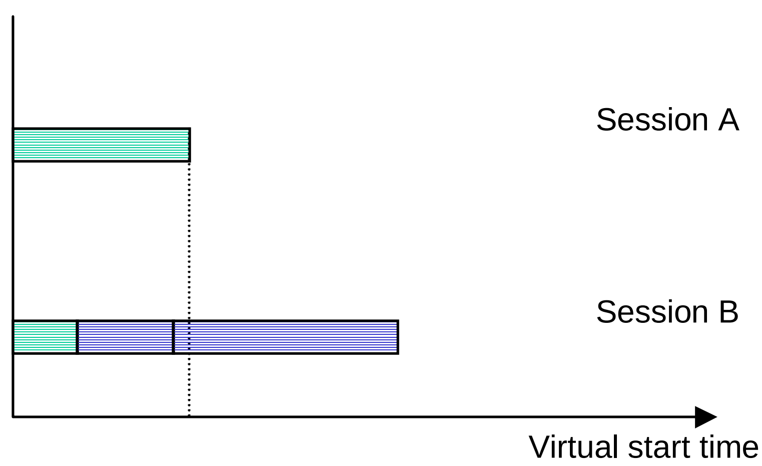
$\boldsymbol{lag_B}$ = + a_2.length

Invariant: $\sum_{i \in A} lag_i = 0$

A is the set of active sessions

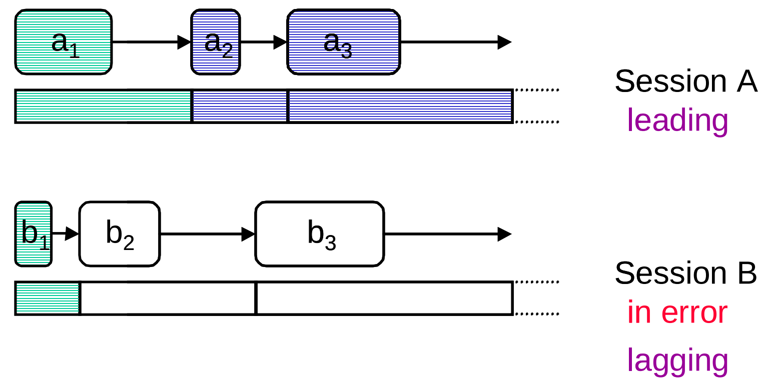
Real system S

Use *lag_i* to keep track of deviations



Error-free reference system S^r_{SFQ}

$\boldsymbol{lag_A}$ = - a_2.length - a_3.length



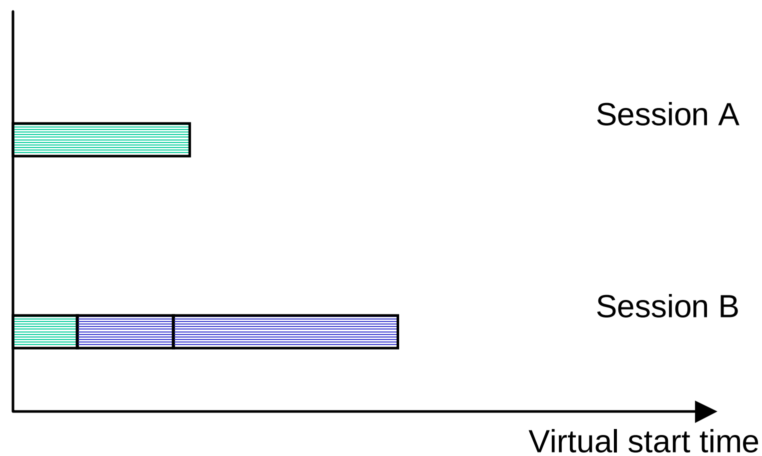
$\boldsymbol{lag_B}$ = + a_2.length + a_3.length

Invariant: $\sum_{i \in A} lag_i = 0$

A is the set of active sessions

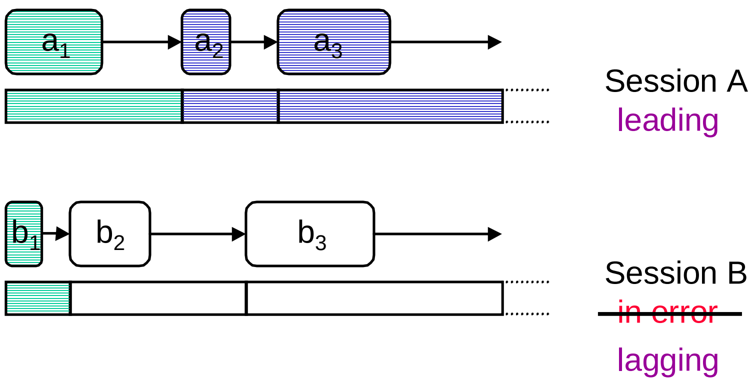
Real system *S*

Leading sessions give back service to lagging sessions



Error-free reference system S^r_{SFQ}

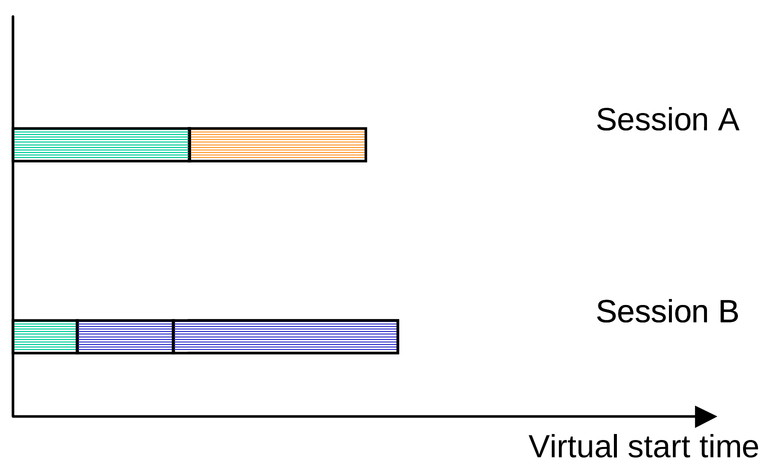
lag_A = - a_2.length - a_3.length



lag_B = + a_2.length + a_3.length

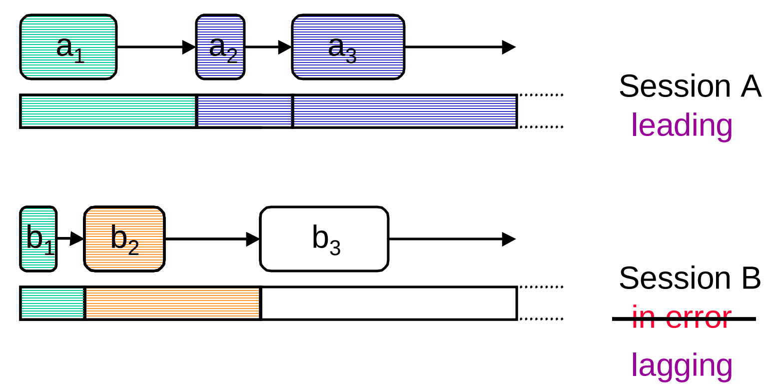
Real system S

Leading sessions give back service to lagging sessions



Error-free reference system S^r_{SFQ}

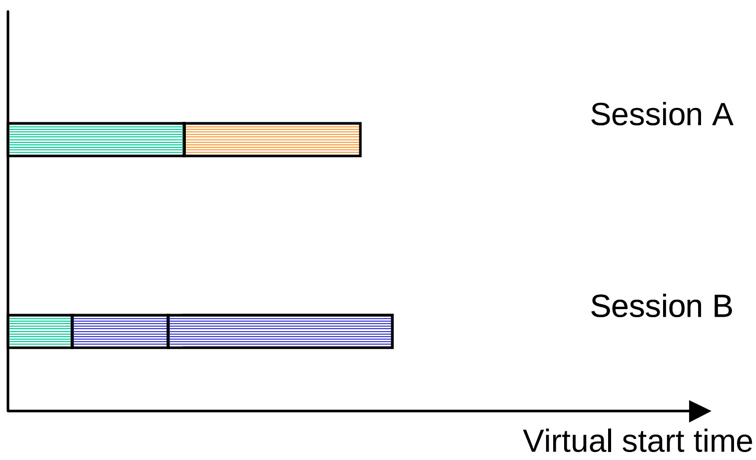
lag_A = - a_2.length - a_3.length



lag_B = + a_2.length + a_3.length

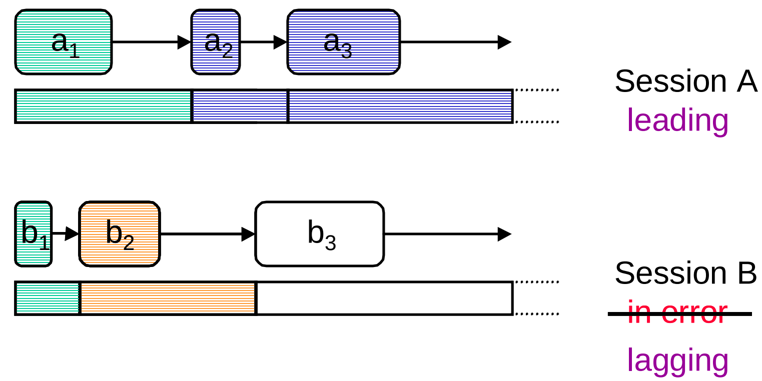
Real system S

Leading sessions give back service to lagging sessions



Error-free reference system S^{r}_{SFQ}

lag_A = - a_2.length - a_3.length + b_2.length



lag_B = + a_2.length + a_3.length - b_2.length

Real system S

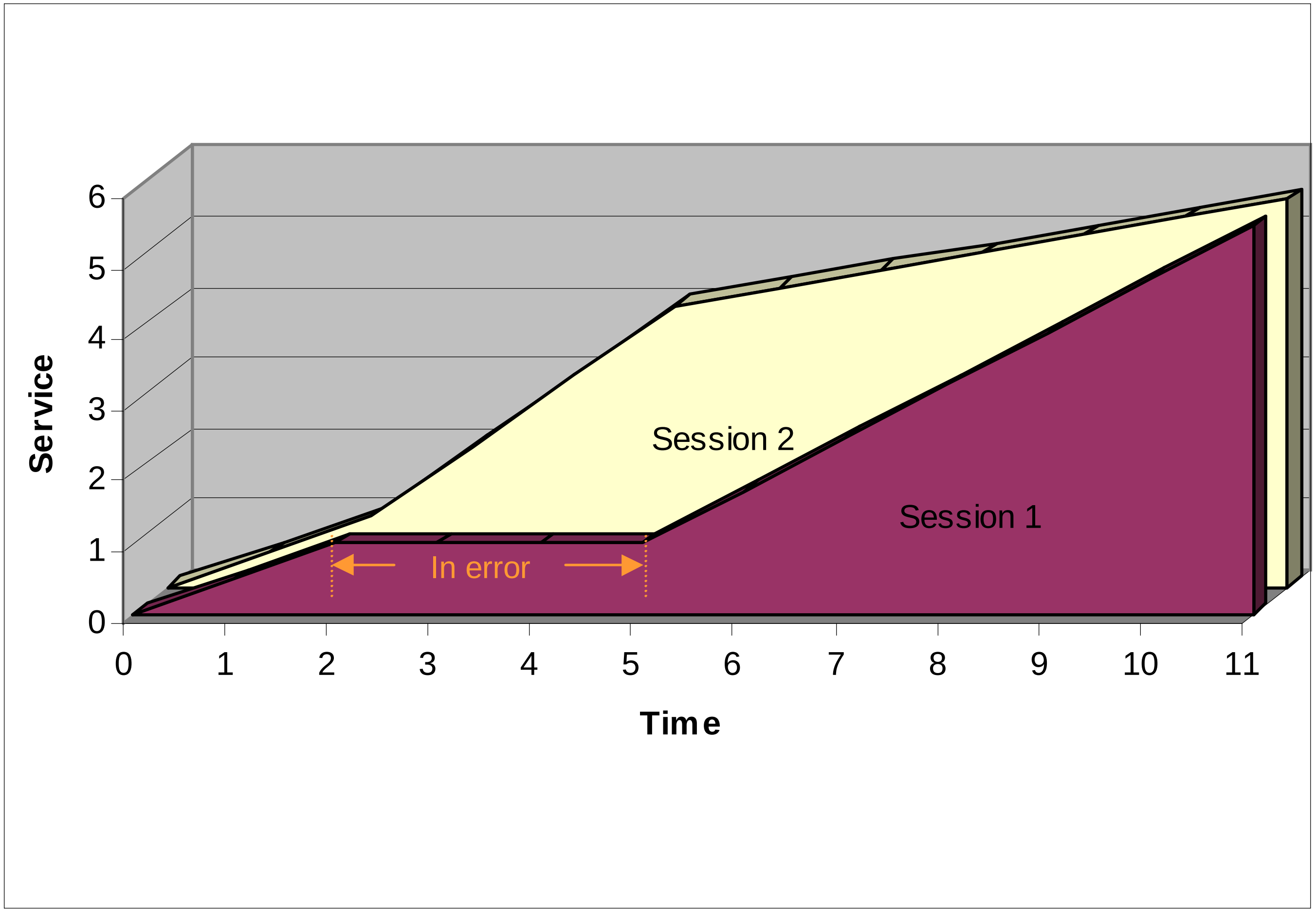
CIF-Q in a nutshell

- Select a session *i* based on reference system sessions' virtual times
- If session *i* is not in error and is allowed to keep this service, serve session *i* normally
- Otherwise, select session *j* with the largest normalized lag_j
  - Serve session *j* but **charge** service to session *i*
  - Adjust lag_i and lag_j accordingly

Leading sessions give back only a fraction of their service

- System parameter α controls how much service is retained by leading sessions
  - $0 \leq \alpha \leq 1$
  - At most (1 - α) of the service is given back
- Control the speed of compensation, **not** the amount of compensation
- Achieve graceful degradation in service for leading sessions

CIF-Q $\alpha = 0.5$



Leading unbacklogged sessions are not allowed to leave

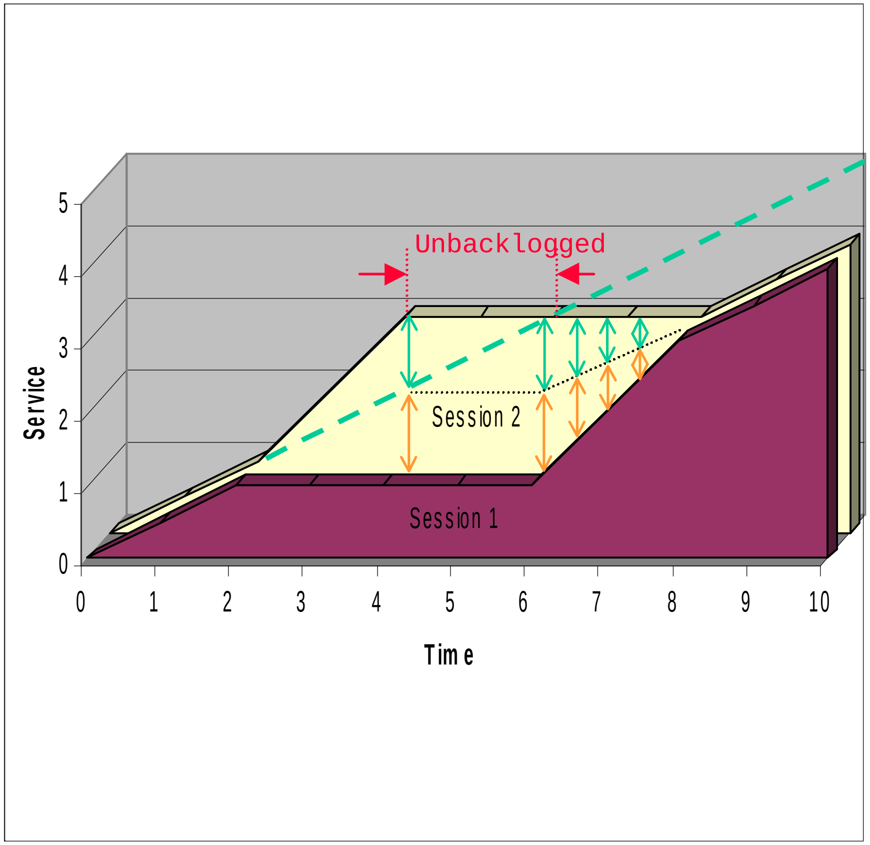
- Leading unbacklogged sessions have negative lag_i
- Must ensure that they give back their lead before removing them from the active set
  - Maintain invariant $\sum_{i \in A} lag_i = 0$
  - Prevent sessions from endlessly gaining extra services
  - Prevent sessions from getting penalized unnecessarily in the future

Use forced compensation when necessary

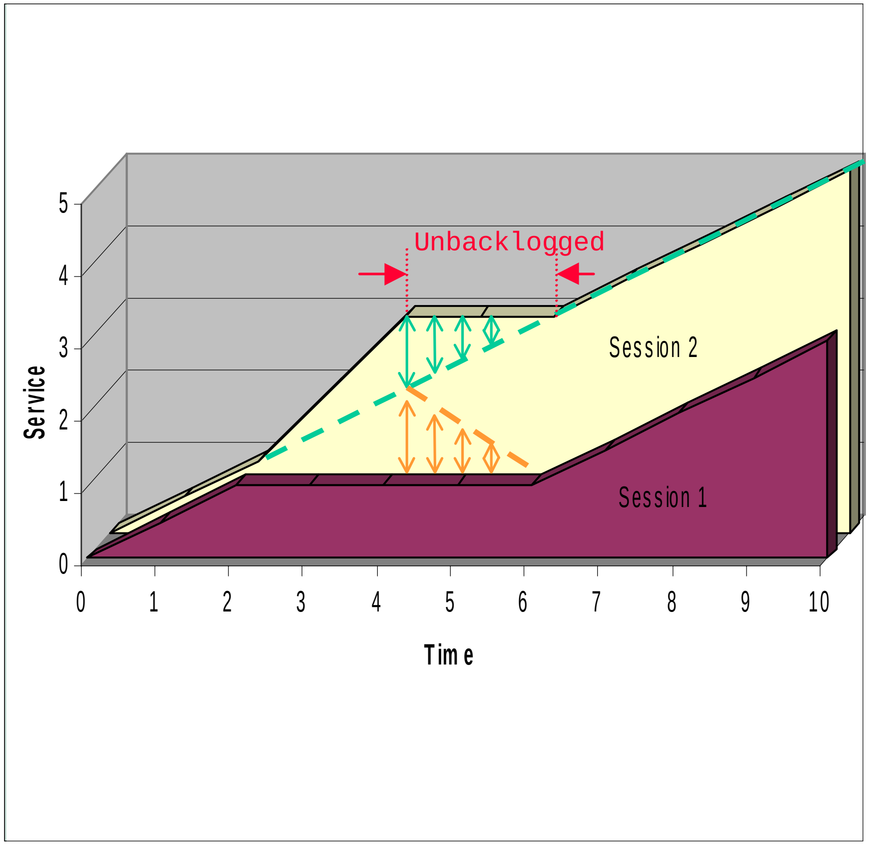
- When all backlogged sessions are in error and a leading session needs to give up its lead
- Must *force compensation* onto a lagging error session to ensure error-free sessions' delay bounds

Forced compensation

Without forced compensation



With forced compensation



Forced compensation

- Pick a lagging error session
- *Force* a small amount of compensation onto this error session
  - Charge the service to the leading unbacklogged session
  - Adjust virtual times and lag_i accordingly

Fair allocation of services

- Use extra virtual time variables to distribute services fairly to leading and non-leading sessions
  - Sessions in the same state (leading or non-leading) are treated the same way
  - As opposed to the session with the largest lag getting all compensation exclusively
- Ensure short-term fairness

What have we accomplished?

- Decoupled delay and fairness properties
  - Use an error-free reference system for scheduling, ensure no divergence in sessions' virtual times v_i
  - Use a second session parameter lag_i to keep track of the difference between the real system and the reference system, ensure long-term fairness
- Ensured graceful degradation
  - Leading sessions give back only a fraction of their service

Theoretical results on CIF-Q

- Delay bound for an error-free session is within one packet transmission time at the session's rate of the bound provided by SFQ
- A lagging session is guaranteed to get compensated for its lag when it becomes error-free
- The normalized amount of service received by two sessions in the same state (leading or non-leading) is tightly bounded

Simulation results

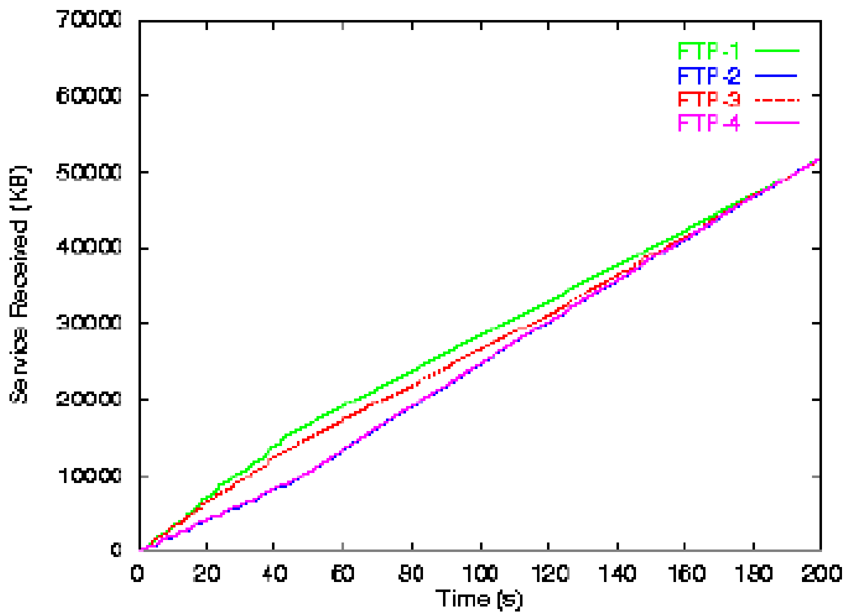
	Packet size	*Guaranteed rate*	*Source model*	*Error*
FTP-1	3 KB	2 Mbps	Greedy	None
FTP-2	3 KB	2 Mbps	Greedy	Pattern 1
FTP-3	8 KB	2 Mbps	Greedy	Pattern 2
FTP-4	8 KB	2 Mbps	Greedy	Pattern 1
Video	8 KB	1.25 Mbps	CBR	None

Pattern 1: Periodic error burst of 1.6 second with 3.2 seconds of error-free time

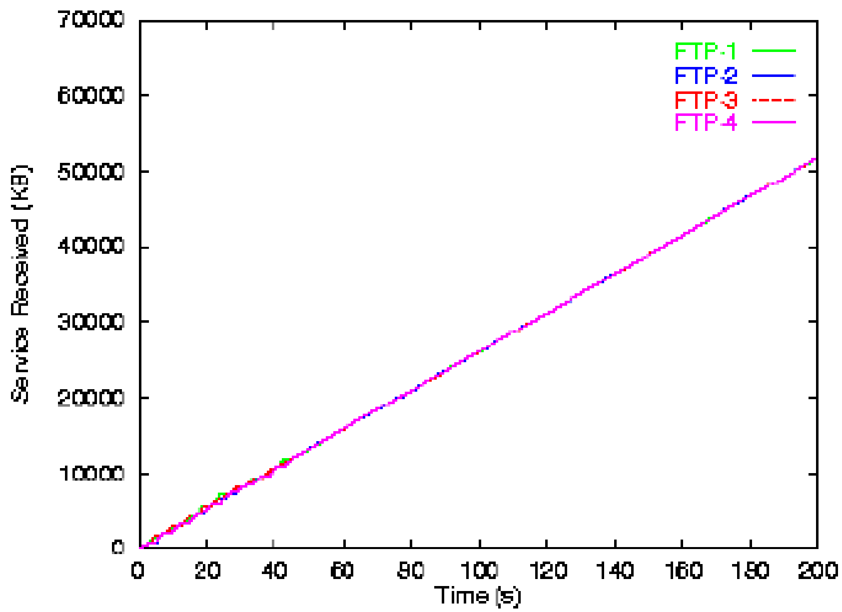
Pattern 2: Periodic error burst of 0.5 second with 5.5 seconds of error-free time

FTP sessions progression

α = 0.9



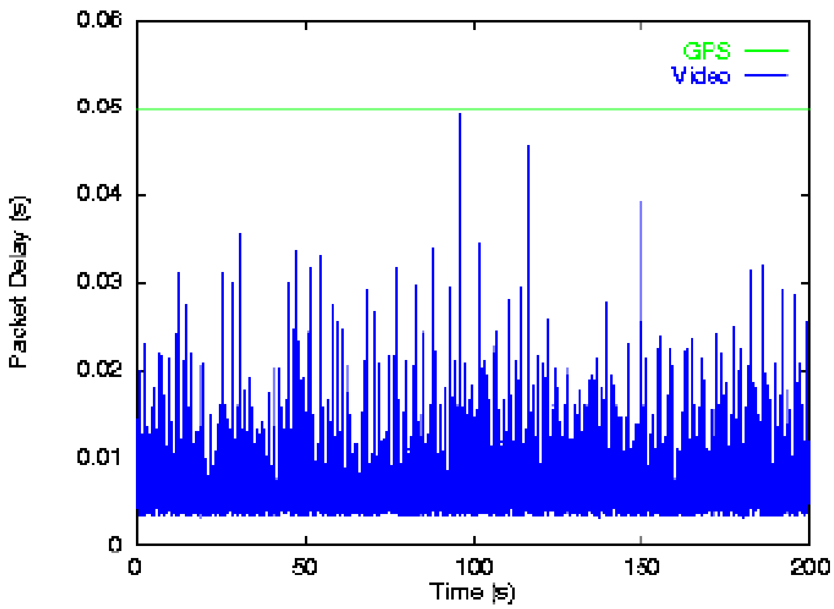
α = 0.0



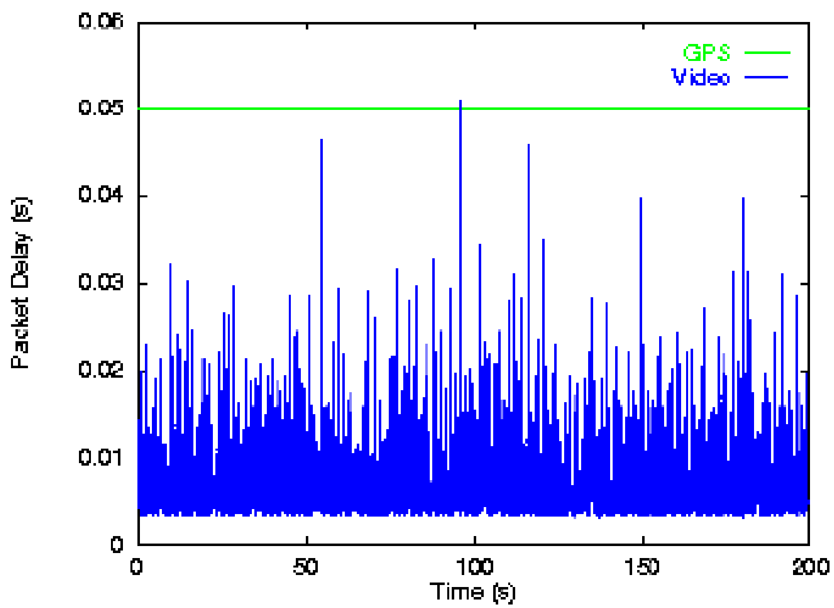
Error bursts from t = 0 to t = 45, error free for t > 45

Video packet delay

α = 0.9



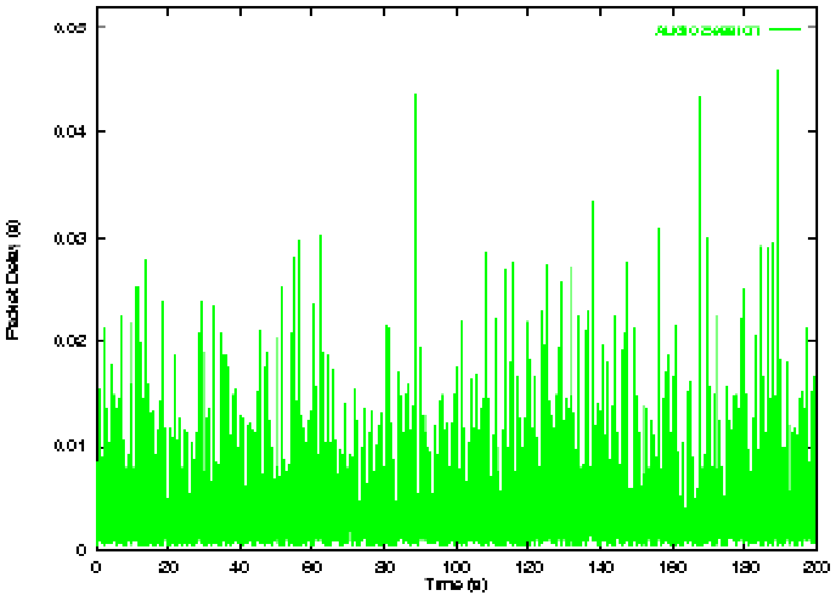
α = 0.0



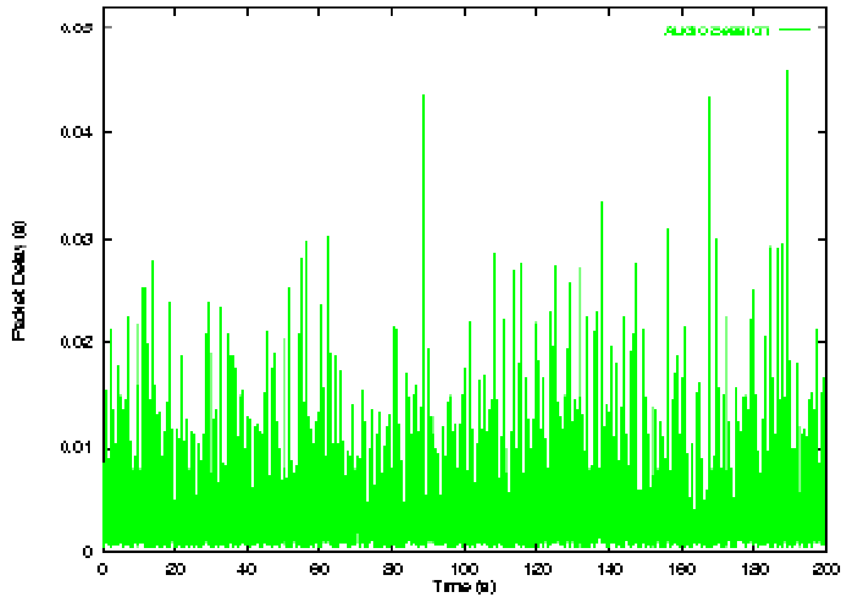
For comparison, GPS would guarantee a delay bound of 50 ms if the system is error-free

Audio packet delay

$\alpha = 0.9$



$\alpha = 0.0$



For comparison, GPS would guarantee a delay bound of 50 ms if the system is error-free

Other related work

- Channel State Dependent Packet Scheduling (CSDP) [Bhagwat et al. INFOCOM '96]
  - Defer link layer retransmission for error sessions
  - Eliminate head-of-line blocking
  - Does not focus on providing guarantee and fairness
- CBQ based CSDP [Fragouli et al. INFOCOM '98]
  - Modify CBQ to tune rates of sessions to achieve some fairness
  - Difficult to characterize service precisely

Other related work

- Server Based Fairness Approach [Ramanathan and Agrawal. MOBICOM '98]
  - General approach to augment any PFQ algorithm
  - Explicitly set aside fixed bandwidth for compensation
  - Essentially penalize all error-free sessions to obtain compensation service since some bandwidth is reserved
  - Require 2n queues and a hierarchical scheduler to achieve fairness for n sessions

Conclusions

- GPS cannot be applied directly to wireless networks
- CIF identifies the four desirable properties in a wireless environment
- Four novel algorithmic techniques introduced in CIF-Q to achieve all CIF properties
  - Use reference error-free system for scheduling
  - Use lag_i to keep track of deviations
  - Leading sessions are not allowed to leave the active set
  - Use forced compensation to ensure delay bounds
- CIF-Q provides delay bound similar to that of SFQ
- Low delay bound and long-term fairness can co-exist