End-to-End Congestion Control for InfiniBand

Jose Renato Santos, Yoshio Turner, John Janakiraman

HP Labs

Outline

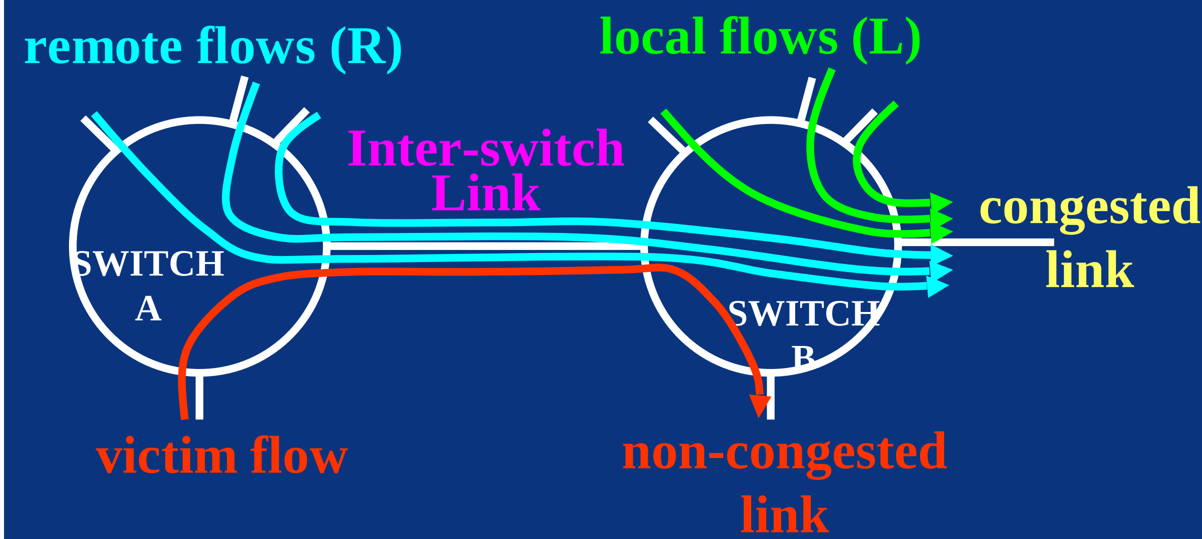
- Motivation: Unique System Area Network (SAN) characteristics require new congestion control approach
- Proposed approach appropriate for SANs:
  - ECN packet marking
  - Source response: rate control with window limit
- Focus: Design of source response functions
  - New convergence conditions, design methodology
  - New functions: LIPD and FIMD
- Performance Evaluation: LIPD, FIMD, AIMD
- Conclusions

System Area Networks Characteristics

- **InfiniBand example: Industry standard server interconnect – 2Gb/s(1x) to 24Gb/s(12x) links**
- **Characteristics: congestion control implications**
  - **No packet dropping**
    - à **Need network support for detecting congestion**
  - **Low network latency (tens of ns cut-through switching)**
    - à **Simple logic for hardware implementation**
  - **Low buffer capacity at switches (e.g., 2KB input buffer stores only four 512-byte packets)**
    - à **TCP window mechanism inadequate (narrow operational range)**
  - **Input-buffered switches**
    - à **Alternative congestion detection mechanisms**

Problem: Congestion Spreading

Flow not using congested link suffers performance degradation (victim flow)



Link BW: 8 Gb/s (4x link)
Packet Size: 2 KB
Buffer Size: 4 packets/port (8 KB)
Buffer Org.: Input port

Simulation (R=L=10)

- **Remote flows use only 30% of inter-switch link bandwidth**
- **Contention for root link à full buffer à prevents victim flow from using remaining inter-switch link bandwidth**

Our Congestion Control Approach

- **Explicit Congestion Notification (ECN) for input-buffered switches**
- **Source adjusts packet injection according to network feedback encoded in ECN returned via ACK**
  - **Combines window and rate control**
  - **New source response functions more efficient than AIMD**

Source Response: Rate Control with Window Limit

- **Window Control**
  - + **Self-clocked, bounds switch buffer utilization**
  - – **Narrow operational range (window=2 uses all bandwidth in idle network)**
  - – **Window=1 is too large if # flows > # buffer slots**
- **Rate Control**
  - + **Low buffer util. possible (< 1 packet per flow)**
  - + **Wide operational range**
  - – **Not self-clocked**
- **Proposed Approach: Rate control with a fixed window limit (w=1)**

Designing Rate Control Functions

- **Definition: When source receives ACK**

  Decrease rate on marked ACK: $r_{new} = f_{dec}(r)$

  Increase rate on unmarked ACK: $r_{new} = f_{inc}(r)$

- **$f_{dec}(r)$ and $f_{inc}(r)$ should provide :**
  - **Congestion avoidance**
  - **High network bandwidth utilization**
  - **Fair allocation of bandwidth among flows**
- **Develop new sufficient conditions for $f_{dec}(r)$ & $f_{inc}(r)$**
  - **Exploit differences in packet marking rates across flows to relax conditions**
    - **Requires novel time-based formulation**

Avoiding Congested State

- **Steady state: flow rate oscillates around optimal value in alternating phases of rate decrease and increase**
- **Want to avoid time in congested state**

Congestion Avoidance Condition:

$f_{inc}(f_{dec}(r))$ £ r

- **Magnitude of response to marked ACK is larger or equal to magnitude of response to unmarked ACK**

Fairness Convergence

- **[Chiu/Jain 1989][Bansal/Balakrishnan 2001] developed convergence conditions assuming all flows receive feedback and adjust rates synchronously**
  - **Each increase/decrease cycle must improve fairness**

- **Observation: In congested state, the mean number of marked packets for a flow is proportional to the flow rate.**
  - **bias promotes flow rate fairness**

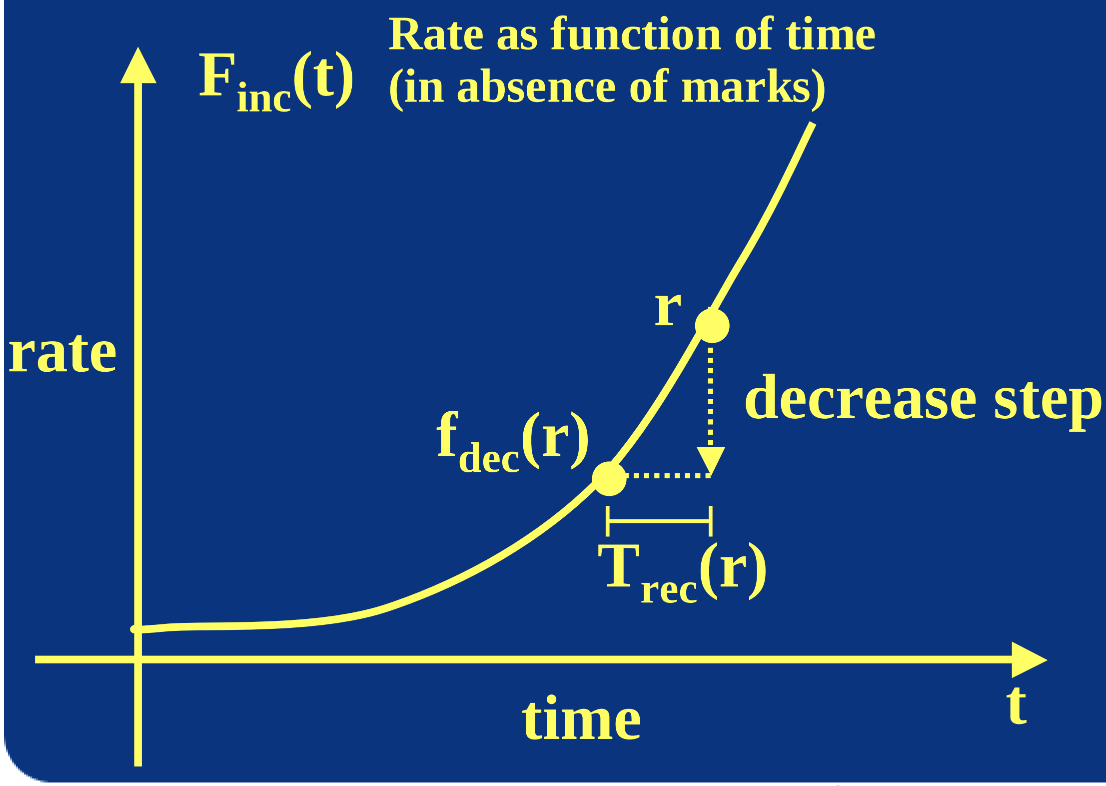
  à **Enables weaker fairness convergence condition**

  à **Benefit: fairness with faster rate recovery**

Fairness Convergence

Relax condition: rate decrease-increase cycles need only maintain fairness in the synchronous case

- **If two flows receive marks, lower rate flow should recover earlier than or in the same time as higher rate flow**



Fairness Convergence Condition:

$T_{rec}(r1)$ £ $T_{rec}(r2)$ for r1 < r2

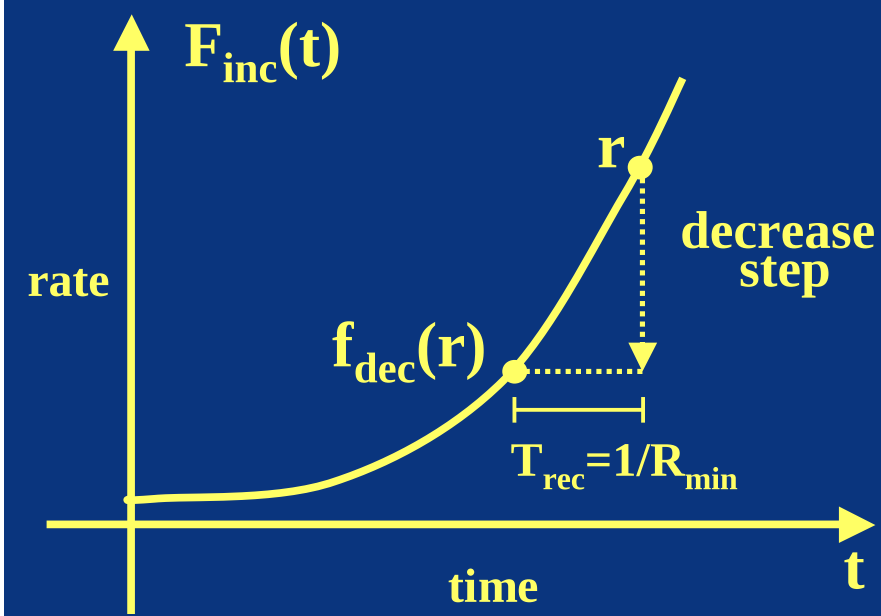
Maximizing Bandwidth Utilization

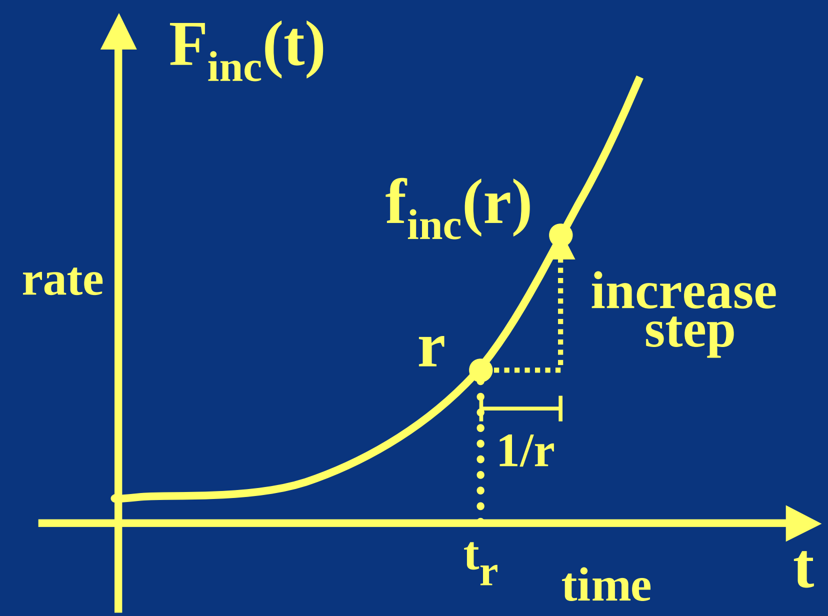
- **Goal: as flows depart, remaining flows should recover rate quickly to maximize utilization**
- **Fastest recovery: use limiting cases of conditions**
  - **Congestion Avoidance Condition $f_{inc}(f_{dec}(r))$ £ r**
    Use $f_{inc}(f_{dec}(r)) = r$ for minimum rate R_{min}
  - **Fairness Convergence Condition $T_{rec}(r1)$ £ $T_{rec}(r2)$**
    Use $T_{rec}(r1) = T_{rec}(r2)$ for higher rates

Maximum Bandwidth Utilization Condition:
$T_{rec}(r) = 1/R_{min}$ for all r

Design Methodology:

Choose $f_{dec}(r)$, find $f_{inc}(r)$ satisfying conditions





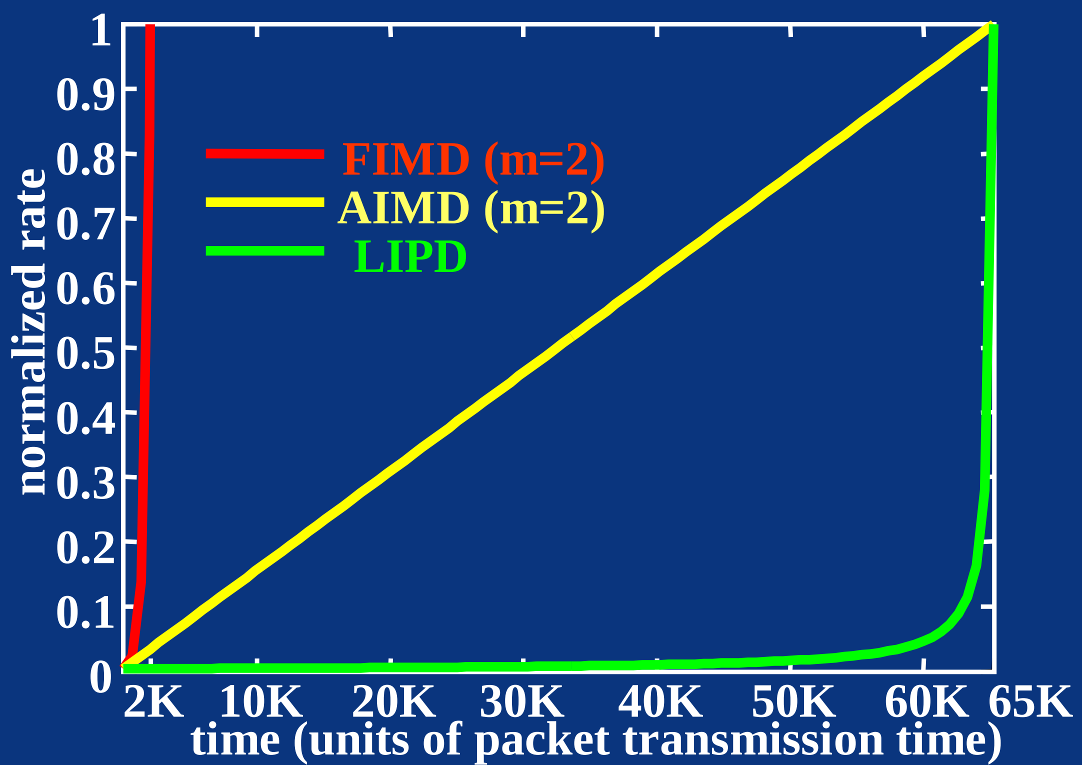
Use $f_{dec}(r)$ to derive $F_{inc}(t)$:
$F_{inc}(t) = f_{dec}(F_{inc}(t + T_{rec}))$,
$T_{rec}=1/R_{min}$

Use $F_{inc}(t)$ to find $f_{inc}(r)$:
$f_{inc}(r) = F_{inc}(t_r+1/r)$
where $F_{inc}(t_r) = r$

New Response Functions

- **Fast Increase Multiplicative Decrease (FIMD):**
  - **Decrease function: $f_{dec}^{fimd}(r) = r/m$, constant $m>1$ (same as AIMD)**
  - **Increase function: $f_{inc}^{fimd}(r) = r \cdot m^{Rmin/r}$**
  - **Much faster rate recovery than AIMD**
- **Linear Inter-Packet Delay (LIPD):**
  - **Decrease function: increases inter-packet delay (ipd) by 1 packet transmission time**
    $r = R_{max}/(ipd+1)$
  - **Increase function: $f_{inc}^{lipd}(r) = r/(1- R_{min}/R_{max})$**
  - **Large decreases at high rate, small decreases at low rate**
- **Simple Implementation: e.g., table lookup**

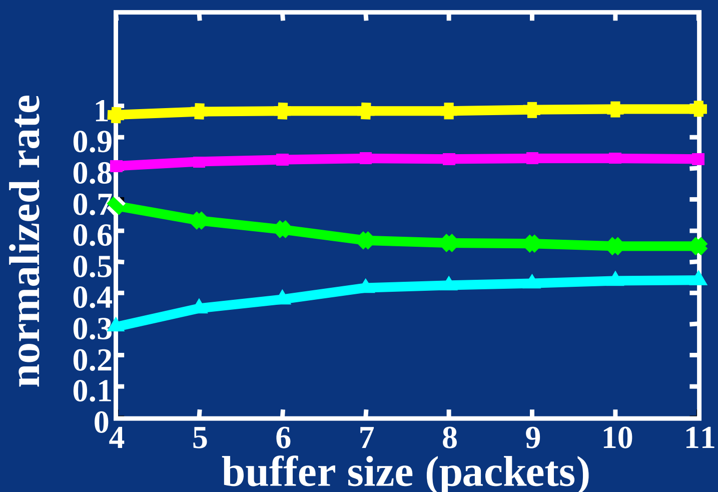
Increase Behavior Over Time : FIMD, AIMD, LIPD



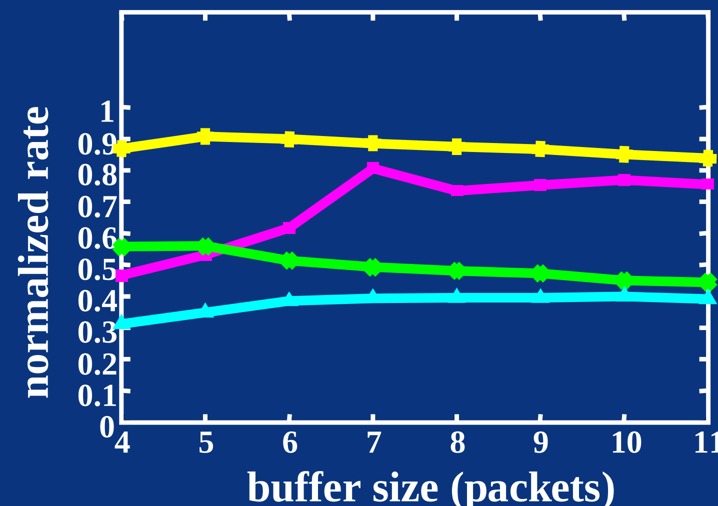
$F_{inc}(t)$

Performance: Source Response Functions

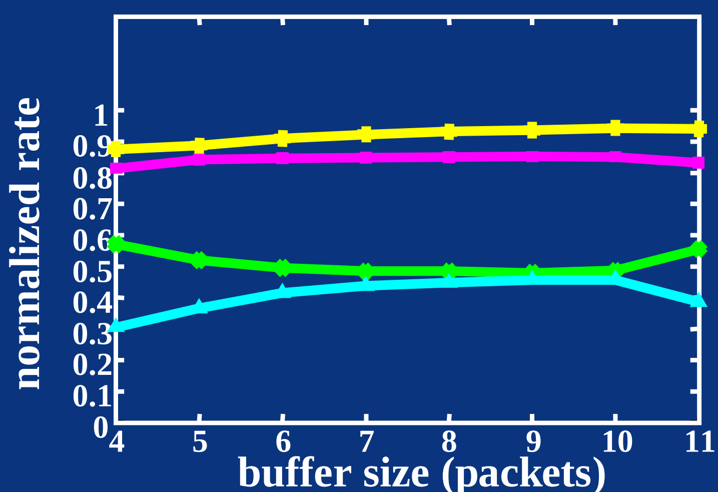
LIPD



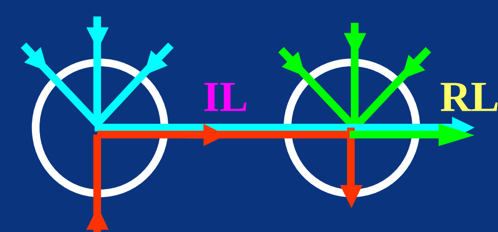
AIMD



FIMD



root link (RL)
inter-switch link (IL)
local flows (LF)
remote flows (RF)



Conclusions

- **Proposed/Evaluated congestion control approach appropriate for unique characteristics of SANs such as InfiniBand**
  - **ECN applicable to modern input-queued switches**
  - **Source response: rate control w/ window limit**
- **Derived new relaxed conditions for source response function convergence à functions with fast bandwidth reclamation**
  - **Based on observation of packet marking bias**
  - **Two examples: FIMD/LIPD outperform AIMD**
- **Future extensions:**
  - **Hybrid window-rate control (allow w > 1)**
  - **Evaluation with richer traffic patterns/topologies**