

96-0178R1

**Comments on
“Use-it or Lose-it”
(Annex F of TM4.0)**

**Raj Jain, Shiv Kalyanaraman, Rohit Goyal, Sonia Fahmy,
Fang Lu**

The Ohio State University

Saragur M. Srinidhi

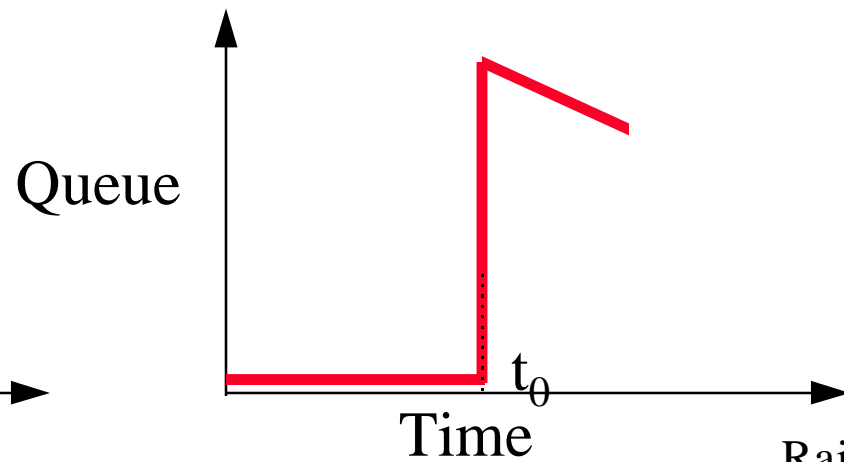
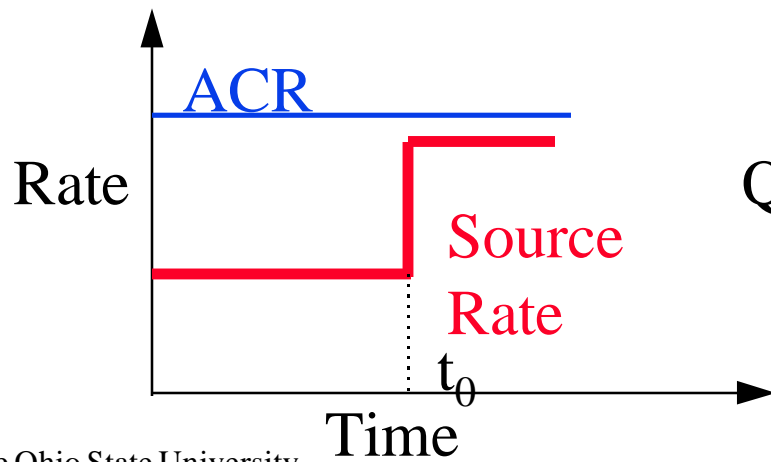
Sterling Software and NASA Lewis Research Center

Contact: Jain@CIS.Ohio-State.Edu

<http://www.cis.ohio-state.edu/~Jain/>

Use-it or Lose-it

- ❑ If a source does not use its allocation, the switch may give it to other sources. But if all of them start using it, there may be big queues and cell loss
- ❑ Queue growth = $(\text{ACR} - \text{Source Rate}) \times \text{Feedback delay} \times (\text{Number of Sources} - 1)$
- ❑ ACR Retention even for a small interval
⇒ Switches are exposed to “sudden arrivals”



Solutions

- ❑ How long can a source keep its rate allocation even though it is not using it?
 - ❑ Switch-Based: Keep ACR upto 500 ms max
 - ❑ Source-Based: Keep ACR upto next 32 cells, then 15/16th for next 32 cells, ...
- ❑ Switch-based
 - ⇒ Takes one feedback delay to control
 - ⇒ Switch must buffer excess traffic for one feedback delay
 - ⇒ Limit overbooking
 - ⇒ Under utilization
- ❑ ATM Forum has selected switch-based but NICs may optionally implement source-based policies

End-System Method 1: Forward RM Triggered

- When a in-rate forward RM is sent, if the ACR exceeds the recent transmission rate plus a fixed bound, the ACR is reduced by a multiplicative factor and the next increase inhibited. Specifically, the recent rate R , could be estimated by N_{rm} divided by the time since the last in-rate forward RM-cell was sent. If ACR exceeds $R + ICR$, the next increase is inhibited. If ACR exceeds $R + ICR$, the ACR could be set to the higher of $ACR * 15/16$ and $R + ICR$, and the next increase inhibited.

Pseudo-Code

□ Current:

$$R = N_{rm}/T$$

IF $ACR > R + ICR$ THEN

$$ACR = ACR * 15/16$$

ACR-ok = FALSE

ELSE ACR-ok=TRUE;

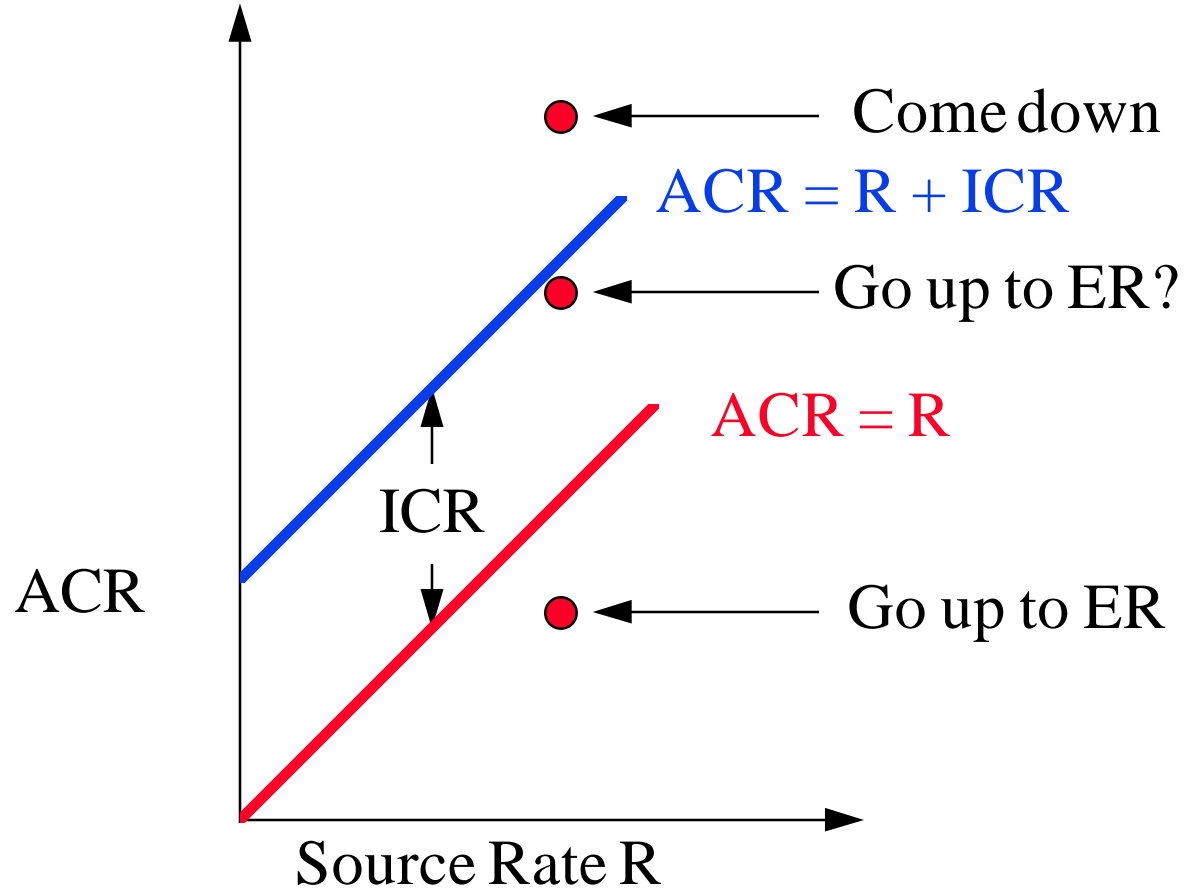
□ Proposed:

$$R = N_{rm}/T$$

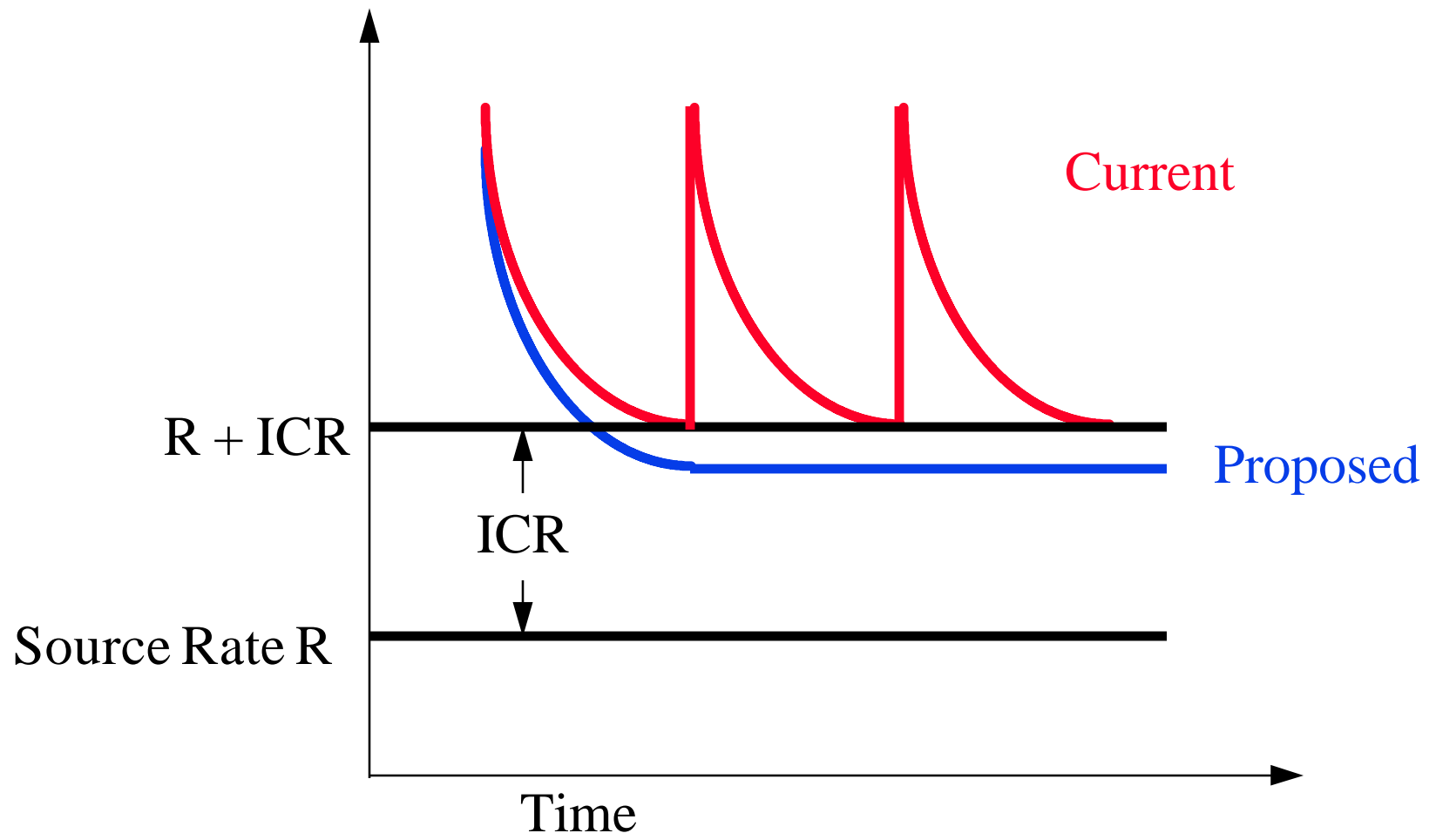
ACR-ok = $ACR \leq R$

IF $ACR > R + ICR$ THEN

$$ACR = ACR * 15/16;$$



- Our proposal introduces a “hysteresis zone” to avoid continuous oscillations between ACR-retention and non-retention



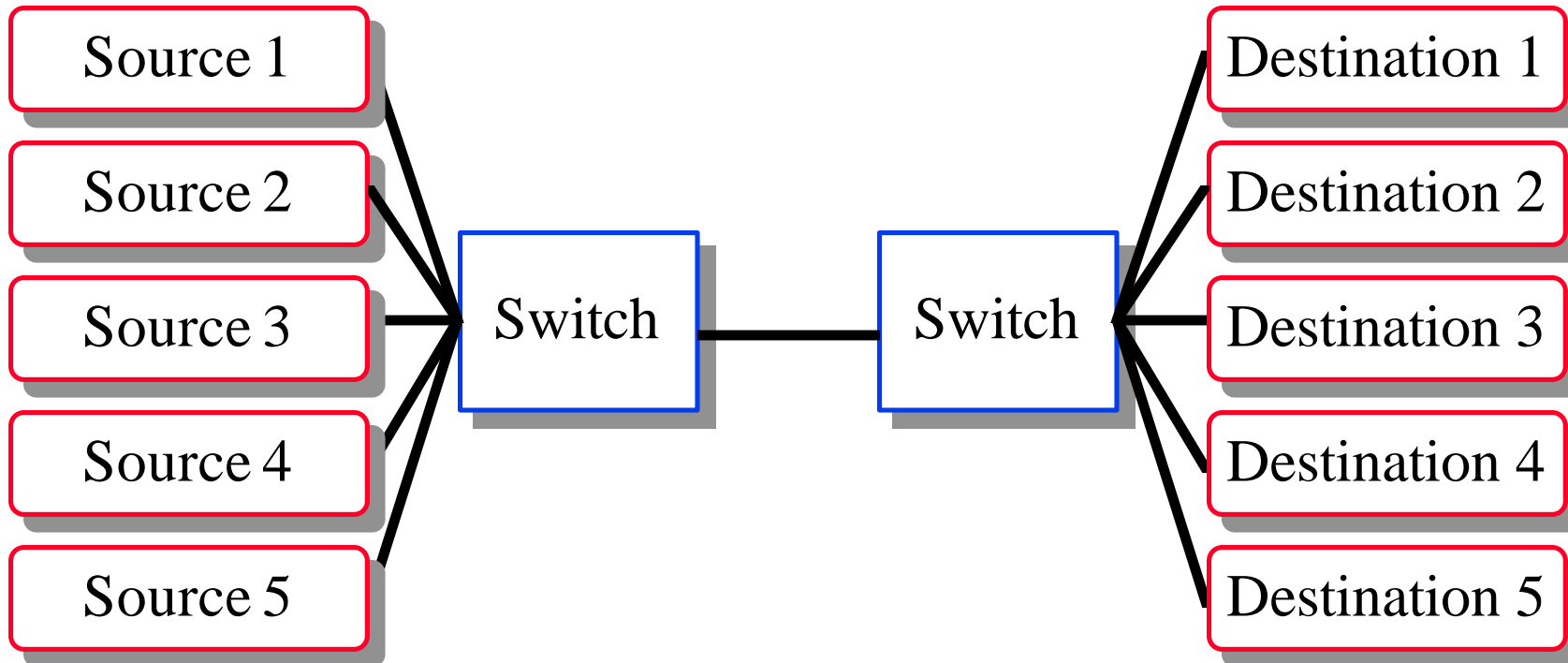
Motion 1

- Change the text to:

When a in-rate forward RM is sent, if the ACR exceeds the recent transmission rate the next increase is inhibited.

Further, if the ACR exceeds the recent transmission rate plus a fixed bound, the ACR is reduced by a multiplicative factor. Specifically, the recent rate R , could be estimated by N_{rm} divided by the time since the last in-rate forward RM-cell was sent. If ACR exceeds R , the next increase is inhibited. If ACR exceeds $R + ICR$, the ACR is set to the higher of $ACR * 15/16$ and $R + ICR$.

Five Source Configuration



← 1000 km → ← 1000 km → ← 1000 km →

- All links 155 Mbps
- All source bottlenecked at 10 Mbps upto 200 ms
At $t = 200$ ms, the sources are able to use their full ACR

Simulation Parameters

- Source: Parameters selected to maximize ACR
ICR = 1 Mbps
TBE = 4096 \Rightarrow Rule 6 disabled
CRM (Xrm) = TBE/Nrm
PCR = 155.52 Mbps, MCR = 0, RIF (AIR) = 1, Nrm = 32,
Mrm = 2, RDF = 1/512, Trm = 100ms, CDF (XDF) = 0,
TCR = 10 c/s
- Traffic: Bi-directional, infinite. Source bottlenecked initially.
- Switch: ERICA
Target Utilization = 90%
Averaging interval = $\min\{30 \text{ cells}, 200 \mu\text{s}\}$

Conclusions

- ❑ It is possible to have ACR retention for 500 ms.
The switch is vulnerable to “sudden arrivals” during this time
- ❑ Ignoring feedback just once causes oscillations.
Network is susceptible to large queues during oscillations.
Our proposal eliminates such oscillations.
- ❑ **Current:**
Ignore once.
As soon as you enter the allowed zone, you can go up
- ❑ **Our proposal:**
You should not go up as long as you have ACR retention

Pseudo-Code: Current

ACR-ok A flag indicating that the ACR passed the "TOF" test

ACR-ok = time \leq TOF * Nrm / ACR ! S5

If not ACR-ok and ACR $>$ ICR ! ACR is too high

ACR = ACR - ACR * time * TDF ! S5a: idle adjust

ACR = max(ACR, ICR)

if receive RM(DIR = backward, CCR, ER, CI, NI, BN) ! S8: adjust ACR

if CI = 1

ACR = ACR - ACR * RDF ! do MD

else if NI = 0 and (ACR-ok or PNI) ! S5b

ACR = ACR + Nrm * AIR ! do AI

ACR = min(ACR, PCR)

ACR = min(ACR, ER) ! S9

ACR = Max(ACR, MCR) ! S9

ACR-ok = true ! S5b

Pseudo-Code: Proposed

ACR-ok An optional flag indicating that source has no ACR retention

$R = N_{rm}/T$! Current rate

ACR-ok = (ACR \leq R) ! No retention

if (ACR $>$ R + ICR) ! ACR is too high

ACR = Max(R + ICR, ACR * 15/16) ! Optional adjustment

if receive RM(DIR = backward, CCR, ER, CI, NI, BN) ! S8: adjust ACR

if CI = 1

ACR = ACR - ACR * RDF ! Do multiplicative decrease

else if NI = 0 and ACR-ok ! Don't inhibit increase

ACR = ACR + AIR ! Additive increase

ACR = min(ACR, PCR)

ACR = min(ACR, ER) ! S9

ACR = Max(ACR, MCR) ! S9

Motion 2 **Motion 2**

