A Survey of Switch Rate Allocation Algorithms

Raj Jain

Raj Jain is now at Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/
Our Team

- **Current:**
  - Shivkumar Kalyanaraman
  - Rohit Goyal
  - Sonia Fahmy
  - Arun Krishnamoorthy
  - Manu Vasandani

- **Past:**
  - Fang Lu
  - Ram Viswanathan
Overview

- MIT Scheme, CAPC2, UCSC, OSU, and others
- ERICA
- ERICA+
- Unpublished modifications of ERICA
Disclaimer

- Some of the information presented here has not been published and is subject of a patent application to be filed.
- This information is being furnished under a non-disclosure agreement.
- Distribution is restricted.
MIT Scheme

- Fair Share = (Capacity - ∑ Underloading VCs' ER)/ (# of Bottlenecked VC's)
- Fair Share > VC's ER ⇒ Underloading VC
- Fair Share < VC's ER ⇒ Bottlenecked VC
- Fair share depends upon bottlenecked VCs and bottlenecked VCs depends upon fair share ⇒ Recursive definition
- ER at this switch = Min{VC's ER, Fair Share}
- Problem:
  - O(n) computation
  - No load measurement ⇒ Inefficiency
    Example: Two sources with ER of 77.5 Mbps
    One bottlenecked at 10 Mbps ⇒ Total load = 87.5 Mbps
OSU Scheme

- **Goals:**
  - $O(1)$ computation
  - Measured load (not just based on ER’s)

- **Key Innovations:**
  - Overload measured by rate and not by queue length
  - Introduced the concept of
    + Averaging interval
    + Target utilization
    + Target utilization band (TUB) $0.90 \pm 0.05$
Algorithm:
- Load = Input rate/Target Rate
- IF outside TUB
  THEN indicate Load factor
  [Now send Source rate/load factor in ER field]
- ELSE Compute fair share and
  Indicate Load/(1+\(\Delta\)) to underloading sources
  and Load/(1-\(\Delta\)) to overloading sources

Problem: Used time-based RM cell transmission
UCSC Scheme

- A modification of the MIT scheme
  1. Use minimum of ER\textsubscript{in\_Cell} and CCR
     \[ \text{Demand}_i = \text{Min}\{\text{ER\textsubscript{in\_Cell}}, \text{CCR}\} \]
  2. Instead of iterating on fair share computation right away, iterate on successive RM cells
- If a VC is currently "bottlenecked" assume unbottlenecked:
  \[ \text{Threshold} = \Sigma \text{Other bottleneck VCs'} \text{ ER}/(\# \text{ of Bottleneck VC's } - 1) \]
- If a VC is currently "not bottlenecked" assume bottlenecked:
  \[ \text{Threshold} = (\text{This VC's ER} + \Sigma \text{Other bottleneck VCs'} \text{ ER})/(\# \text{ of Bottleneck VC's } + 1) \]
3. Fair Share = \text{Max}\{\text{Fair Share, Threshold}\}
4. Adjust the VC's classification by comparing it with the new fair share:
   \begin{align*}
   \text{Bottlenecked}_i &= \text{Demand}_i > \text{Fair Share} \\
   \text{Allocation}_i &= \text{Min}\{\text{Demand}_i, \text{Fair Share}\} \\
   \text{ER}\_\text{in}\_\text{Cell} &= \text{Min}\{\text{ER}\_\text{in}\_\text{Cell}, \text{Fair Share}\}
   \end{align*}
5. Remember VC with the largest allocation. This should always be bottlenecked.

IF $\text{Allocation}_i > \text{max\_allocation}$

THEN

$\text{Max\_VC} = i; \text{max\_allocation} = \text{Allocation}_i$;

IF state $\neq$ bottlenecked

THEN State = Bottlenecked;

$\text{N\_Bottleneck} = \text{N\_Bottleneck} + 1$;

END IF

END IF

IF $\text{max\_VC} = i$ and $\text{Allocation}_i < \text{Max\_allocation}$

THEN $\text{Max\_allocation} = \text{Allocation}_i$

END IF
UCSC Scheme (Cont)

- **Problems:**
  - Sets ER in the forward direction
  - No load measurement
    - \(\Rightarrow\) May not work if source bottlenecked.
  - Need to measure active VC's
HKUST Scheme

- Modification of MIT Scheme
  - Use MIT scheme in both forward and reverse direction
  - Reset ER field at the destination
- **Claims**: Fast convergence. Fair.
- **Problems**:
  - O(n) complexity.
  - No load measurement ⇒ May not work if source bottlenecked.
  - Need to measure active VC's.
  - Not compatible with TM4.0
    (resetting ER to PCR at the destination is not allowed)
CAPC2 Scheme

- Congestion Avoidance Using Proportional Control Ver 2
- Borrows some concepts from OSU scheme and ERICA:
  - Monitor input rate.
  - Set target utilization
  - Underload $\delta = 1 - \text{Input Rate/Target Rate}$
- Fair Share is dynamically adjusted to get load close to one
  IF underload $> 0$
  THEN Fair Share = Fair Share $\times \text{Min}\{1+\delta R_{up}, \text{ERU}_{max}\}$
  ELSE Fair_share =Fair Share $\times \text{Max}\{1+\delta R_{down}, \text{ERD}_{Min}\}$
- $R_{Up}$ and $R_{Down}$ control the convergence rate.
  $\text{ERU}_{Max}$ and $\text{ERD}_{min}$ limit the oscillations.
Set CI if Queue > Threshold

Problems:
1. Four parameters
2. Slow convergence
3. Unfairness due to CI bit use
Explicit Rate Indication for Congestion Avoidance
Set target rate, say, at 95% of link bandwidth
Monitor input rate and number of active VCs
   Overload = Input rate/Target rate
This VC’s Share = VC’s Current Cell Rate/Overload
Fair share = Target rate/ Number of Active VCs
ER = \textit{Max}(\text{Fair share, This VC’s share})
ER in Cell = \text{Min}(ER in Cell, ER)
ERICA Features

- Uses measured overload
  ⇒ If sources use less than allocated capacity, all unused capacity is reallocated to others.
- Two parameters: Target utilization, Averaging interval
- Simple
- Order (1) computation
- Fast response due to optimistic design
- Fairness is improved at each step. Even under overload.
- Converges to efficient operation in most cases
- Max-min fair in most cases
Innovation: Use forward CCR

- **Problem**: CCR in backward direction is too old
- **Solution**: Read CCR in forward RM cells. Give feedback in backward RM cells.
- **Effect**: Shorter control loop for active VCs  
  ⇒ Faster convergence
Control vs Feedback Delay

- Fundamental principle of control theory:
- Control faster than feedback ⇒ Instability
  Control slower than feedback ⇒ non-responsiveness
- Ideal: Control rate ≈ Feedback rate
- Control delay = feedback delay = monitoring delay
Innovation: Same Feedback in One Interval

- **Problem**: Oscillations for high-rate sources
- **Reason**: Mismatched control and monitoring intervals
  - Control Interval = Inter-RM cell time = Feedback Interval
  - Monitoring Interval = Averaging interval
- **Solution**: Do not change feedback in one averaging interval.

Diagram:
- Source
- Switch
- Dest.
- Load Measurement Interval
Innovation: Fair Share First

- **Problem:** Transient overloads at state changes
- **Solution:** Source below Fair Share go only up to fair share first.
  
  IF $CCR < \text{Fair Share}$ and $ER_{\text{Calculated}} > \text{Fair Share}$
  
  THEN $ER_{\text{Calculated}} = \text{Fair Share}$

- **Example:** Two sources $\{10, 10\}$, $\{50,10\}$, $\{90,50\}$...
Option: Per-VC Rate Measurement

- **Problem**: Some VCs are bottlenecked at the source. CCR does not reflect source rate.

- **Solution**:
  - Count number of cells in each VC.
  - Source Rate = Number of Cells Seen/Averaging Interval.
  - This VC's Share = Source Rate/Overload.

- **Advantage**:
  - Also handles sources not using their allocation. ⇒ Switch based “use it or lose it”
Modification:
Time + Count Based Averaging

- **Problem**: Averaging over a fixed interval
  ⇒ Sudden overload can cause queue build up
- **Solution**: Average over $t$ ms or $n$ cells whichever happens first.
Innovation: ERICA with VBR

- Monitor VBR usage
- ABR capacity = Target Rate - VBR input rate
- Overload factor = ABR input rate/ABR capacity
- This VC’s share = VC’s CCR/overload factor
- Fair share = ABR capacity/Number of active ABR VCs
- ER = Max{Fair share, This VC’s share}

NOTE: Target utilization applies to total link load
ABR capacity = Target Util. \times Link Rate - VBR output rate
and not
ABR capacity = Target Util. \times(\text{Link Rate} - \text{VBR output rate})
⇒ VBR Output rate < Target utilization
Out-Of Phase Effect

- Bursty load and backward RM (BRM) cells are often out of phase.
- When there is load in the forward direction, there are no BRMs.
- By the time the switch sees BRMs, there is no load in the forward direction.
- The above effect disappears when the bursts become larger than RTT

\[\text{\begin{tikzpicture}[baseline,-,thick]
  \draw[<->] (0,0) -- (1,0);
  \draw[<->] (2,0) -- (3,0);
  \draw[<->] (4,0) -- (5,0);
  \draw[<->] (6,0) -- (7,0);
  \draw[<->] (0,-1) -- (1,-1);
  \draw[<->] (2,-1) -- (3,-1);
  \draw[<->] (4,-1) -- (5,-1);
  \draw[<->] (6,-1) -- (7,-1);
\end{tikzpicture}}\]
Innovation:
Bidirectional Counting

- **Problem**: Data cells or RM cells may not be seen in one direction. Resulting in undercount and overallocation.

- **Solution**: A VC is active if any of the following holds:
  - data cells seen in the forward direction in the last averaging interval
  - Data cells seen in the forward direction in this averaging interval
  - BRMs seen in the reverse direction

- **Option**: Reset CCR = 0 for all inactive sources at the beginning of an averaging interval
  - Not necessary if per-VC source rate measurement is used
Unfairness in ERICA

- \( \text{ER}_{\text{Calculated}} = \text{Max}\{\text{Fair Share}, \text{CCR/overload}\} \)
- ERICA becomes unfair if ALL of the following conditions hold true:
  - Overload = 1
  - Some VCs are bottlenecked at other switches and therefore have CCRs below fair share
  - All VCs that are not bottlenecked at other switches have a CCR greater than the fair share
- Under the above condition, the CCRs do not change at all. The allocation stabilizes. But the stable operating point may not be max-min fair.
Fairness Problem: Example

- Max-Min Allocation of 150 Mbps: \{10, 10, ..., 10, 70, 70\}
- With \{10, 10, ..., 10, 60, 80\}, Link 2 Fair Share = 50, Load = 1
  \[\text{Max}\{\text{Fair share, CCR/load}\} = 60\text{ and } 80\] for VC16 and VC17.
Innovation: Fairness Fix

Solution:
- All VCs that are bottlenecked at this switch must get the same allocation = maximum allocation
- Remember maximum ER in the previous interval
- IF overload $\leq 1+\delta$
  THEN $ER_{\text{Calculated}} = \text{Max}\{\text{Fair Share, CCR/Overload, Max\_ER}\}$
  ELSE $ER_{\text{Calculated}} = \text{Max}\{\text{Fair Share, CCR/Overload}\}$

Example: On Link 2, Fair Share = 50
- \{10, 10, ..., 10, 60, 80\}, Load = 1, ER=10, 80, 80
- \{10, 10, ..., 10, 80, 80\}, Load = 17/15, ER=10, 70.6, 70.6
- \{10, 10, ..., 10, 70.6, 70.6\}, Load = 1.008, ER=10, 70.03, 70.03
Is Low Queue Length Good?

- Queue length is close to 1.
  Not good if bandwidth becomes available suddenly
  You can’t use BECN to ask sources to increase
  Low rate sources may have long inter-RM cell times
- Link utilization is 90% or below
  May not be acceptable for high-cost WAN links.
- Very high queue length is also bad.
Innovation:
ERICA with Queue Control

- Target utilization is dynamically changed.
- During steady state: Target utilization = 100%
- During overload the target may be low, e.g., 80%
- During underload the target may be high, e.g., 110%
- Available Bandwidth = fn(Unused bandwidth, Queue length, queue length goal)
- Unused bandwidth = Link Rate - VBR output rate
- Rest is similar to ERICA
Innovation: Use Queue Delay Threshold

- Since available bandwidth (AB) varies dynamically, a queue of 30 may be too big when AB is 1 Mbps but too little when AB is 100 Mbps.
- Use queue delay instead of queue length
  Queue Delay = Queue length / Available bandwidth
- Available Bandwidth = fn(Unused bandwidth, Queue length, queue delay goal)
Innovation: Target Utilization Function

- The function should be monotonically non-increasing and have a lower bound.

\[ q \]

Queue Delay \( T \)

Factor = \( F_{\text{min}} \)

Available Bandwidth = Unused Bandwidth \( \times \) Factor
Sample Queue Control Function 1

Parameters: \{a, b, T_0, F_{\text{min}}\} = \{1.15, 1.05, 5 \text{ ms}, 0.5\}

Factor = \frac{b T_0}{(b-1)T + T_0}

Factor = \frac{a T_0}{(a-1)T + T_0}

Factor = F_{\text{min}}
Sample Queue Control Function 2

Parameters: \{\{a_1, T_1\}, \{a_2, T_2\}, \ldots, \{a_{n-1}, T_{n-1}\}, a_n\}
Sample Queue Control Function 3

Parameters: \{\{a_1, T_1\}, \{a_2, T_2\}, ..., \{a_n, T_n\}\}
Advantage of Q-Control

- Can tolerate errors in measurements:
  - Number of active sources
  - VBR load
  - ABR input rate
- Allows n-VC TCP operation with buffers $\times 1 \times$ RTT
Both input rate and queue measurements are required. Cannot rely on declared CCRs only. Per-VC source rate measurement required in some cases.

Queue control helps overcome measurement errors.

ERICA has been thoroughly tested by us and others. Source bottleneck, VBR, Bursty TCP sources

Modified ERICA solves the fairness problem.
References


