Ethernet Congestion Manager (ECM)

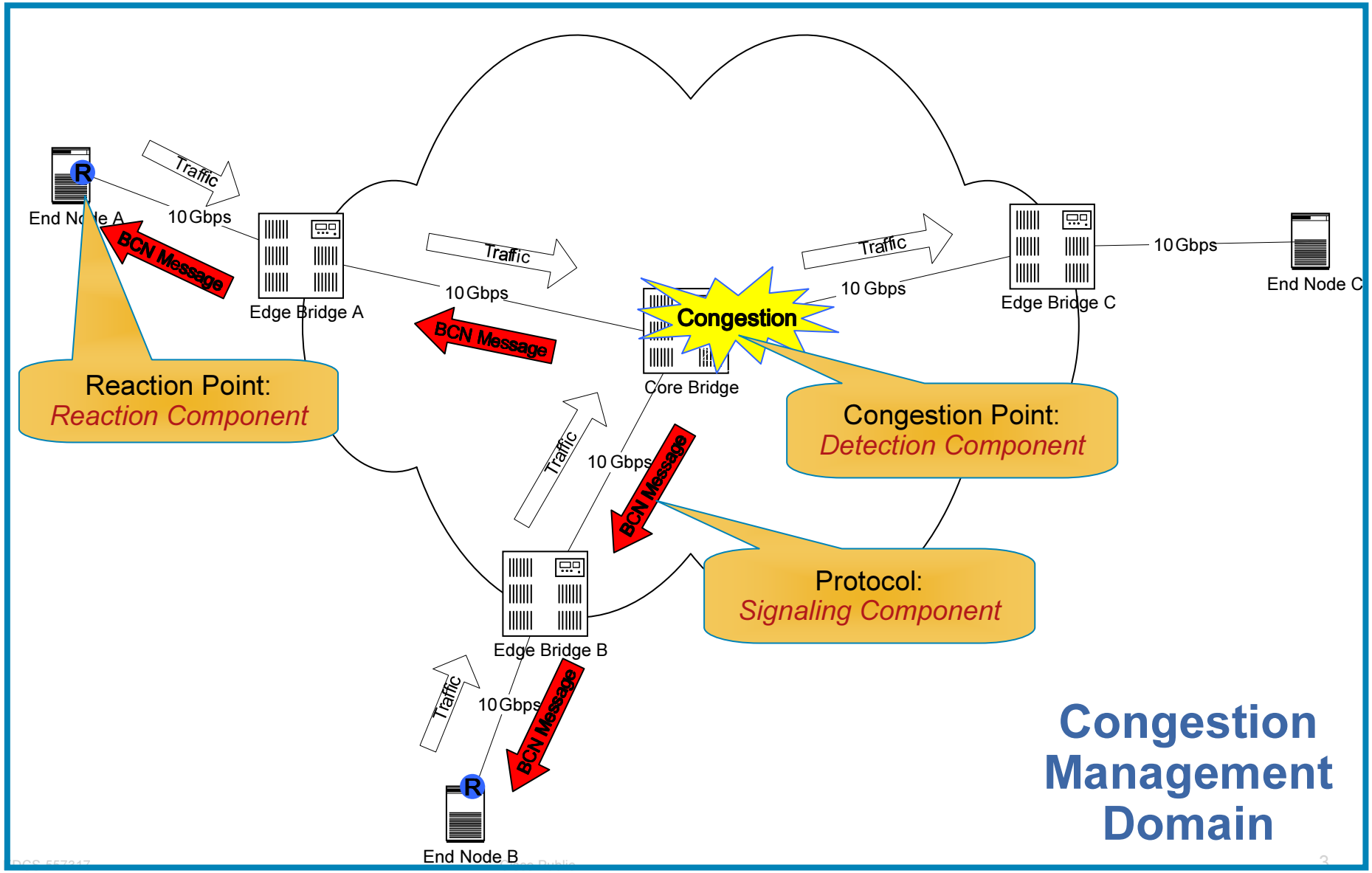
Davide Bergamasco

IEEE 802 Plenary Meeting
Orlando, FL
March 13th, 2007

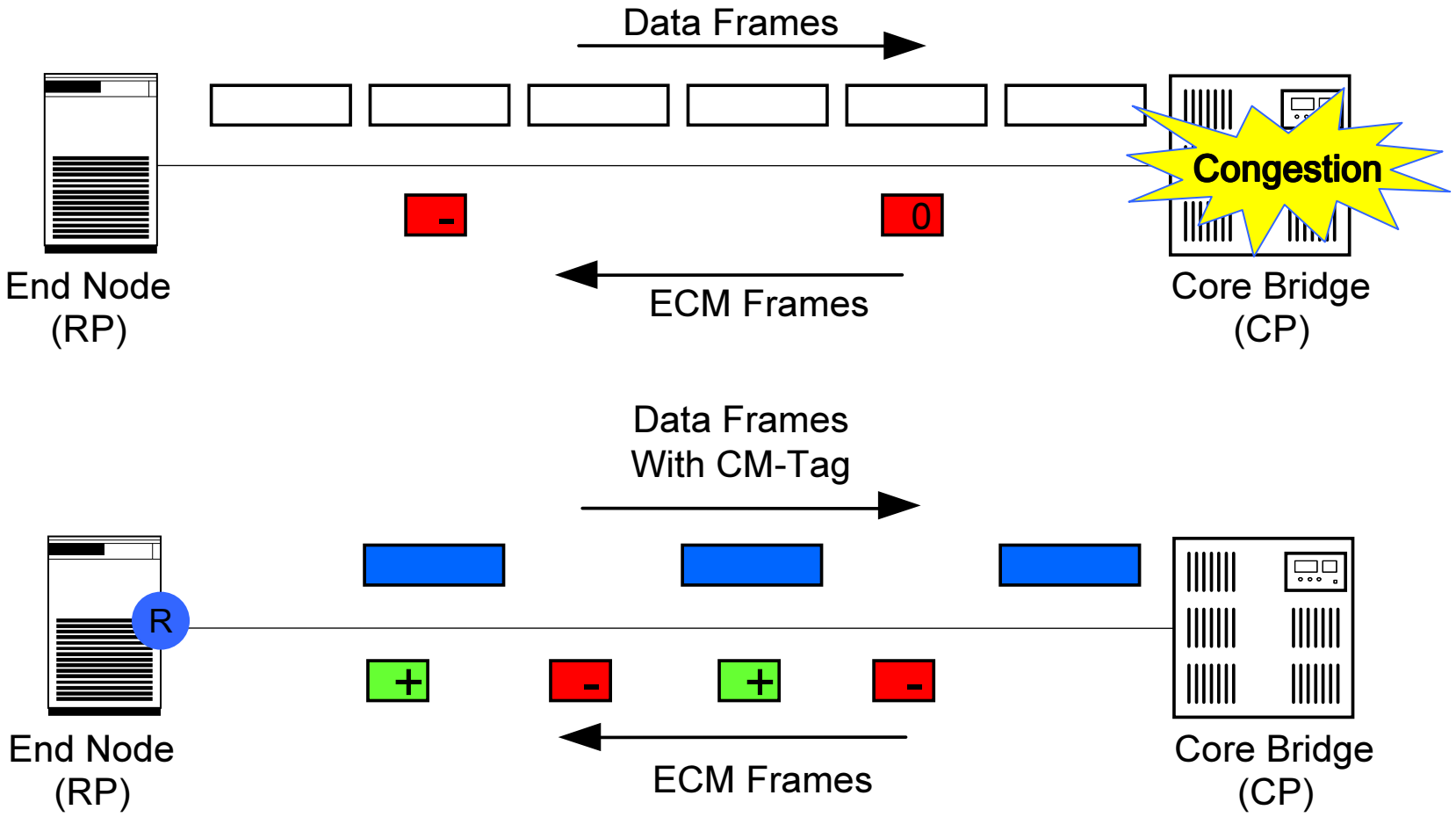
What is ECM?

- ECM is a *Layer 2 congestion management mechanism*
- Formerly known as BCN
- "Comprehensive" document available on IEEE 802.1 website:
  http://www.ieee802.org/1/files/public/docs2007/au-bergamasco-ecm-v0.1.pdf
- Principles
  - Push congestion from the core towards the edge of the network
  - Use rate-limiters at the edge to "shape" flows causing congestion
  - Control injection rate based on feedback coming from congestion points
- Inspired by TCP
  - AIMD rate control
    - TCP window increases linearly in absence of congestion
    - Decreases exponentially (gets halved) at every congestion indication (either implicit or explicit)
  - Self-Clocking Control loop (acknowledgements)

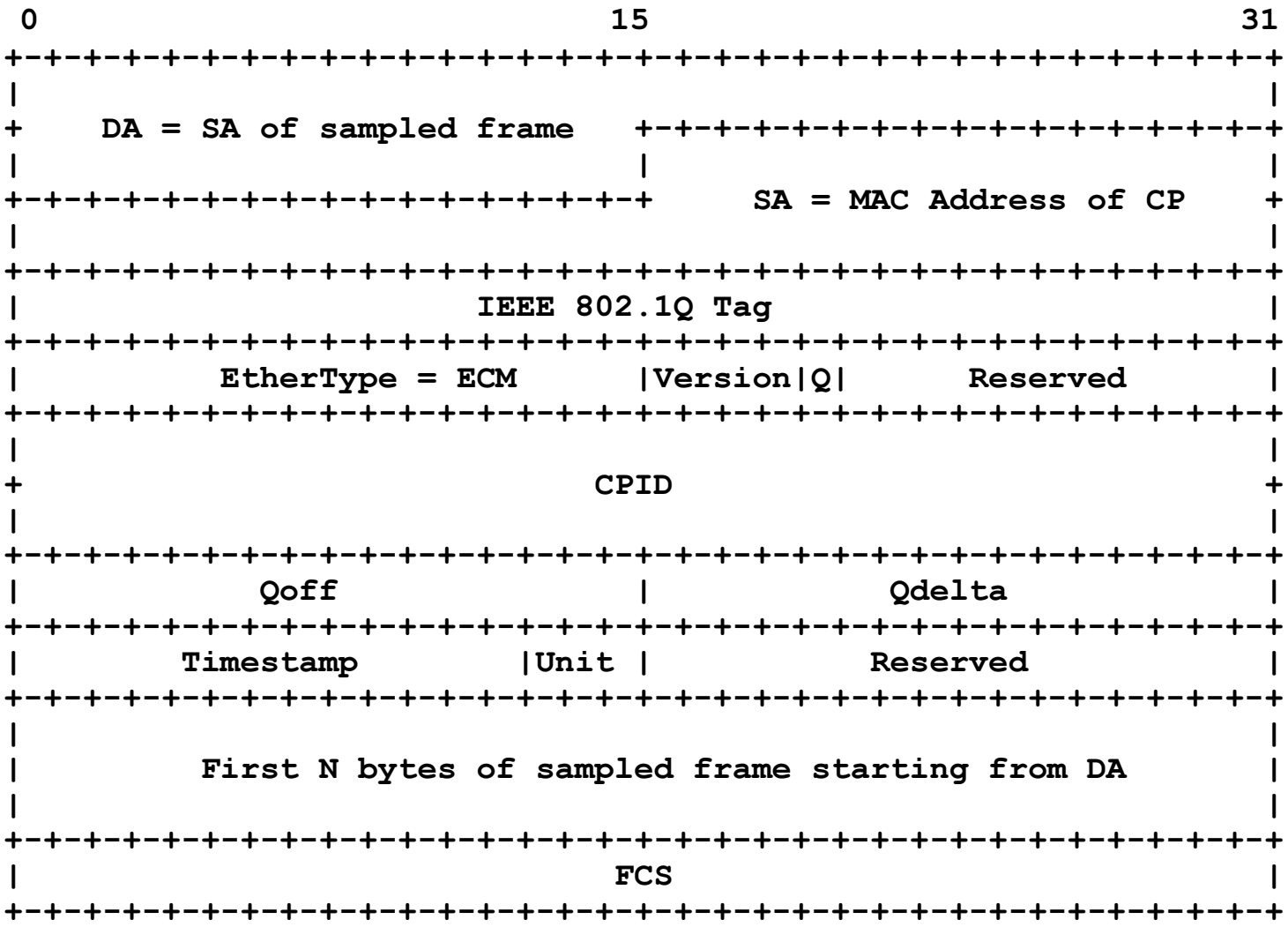
ECM Concepts



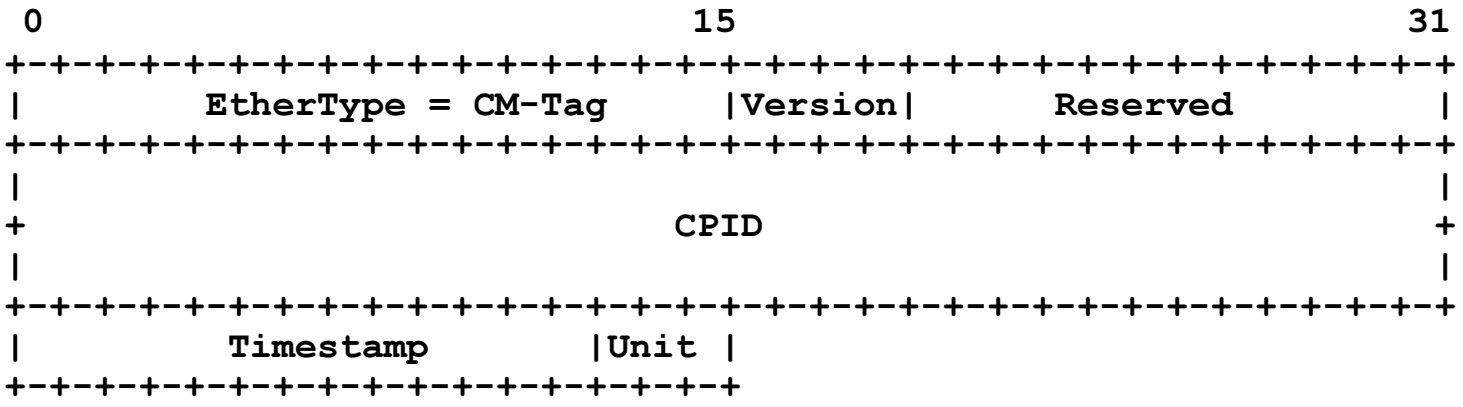
ECM Concepts: Signaling



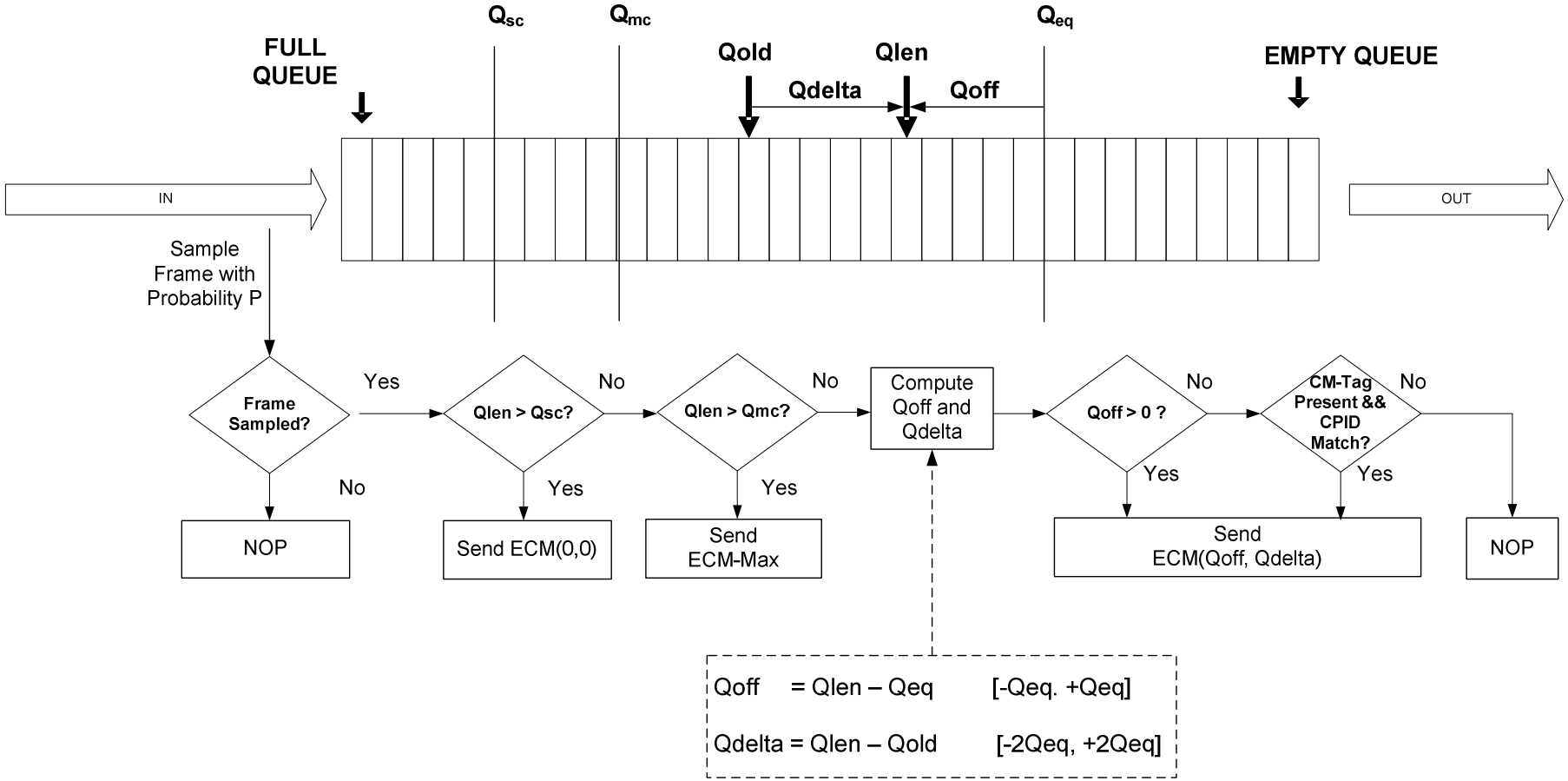
ECM Concepts: Signaling



ECM Concepts: Signaling

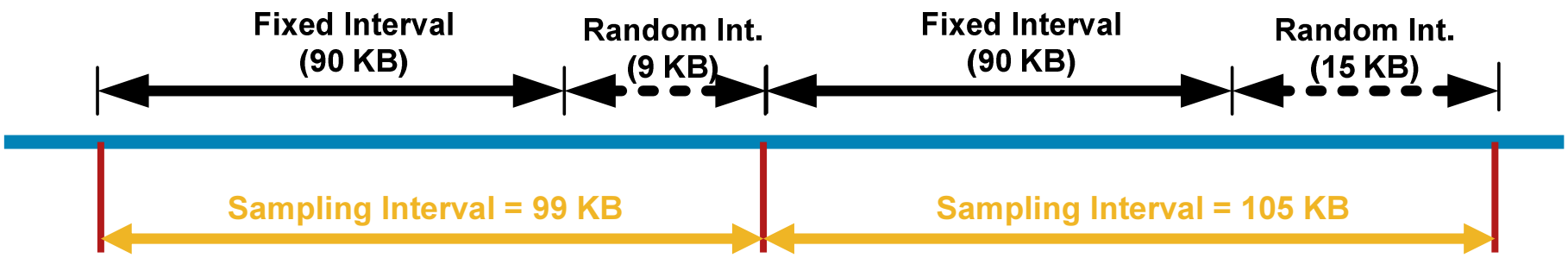


ECM Concepts: Detection



ECM Concepts: Detection

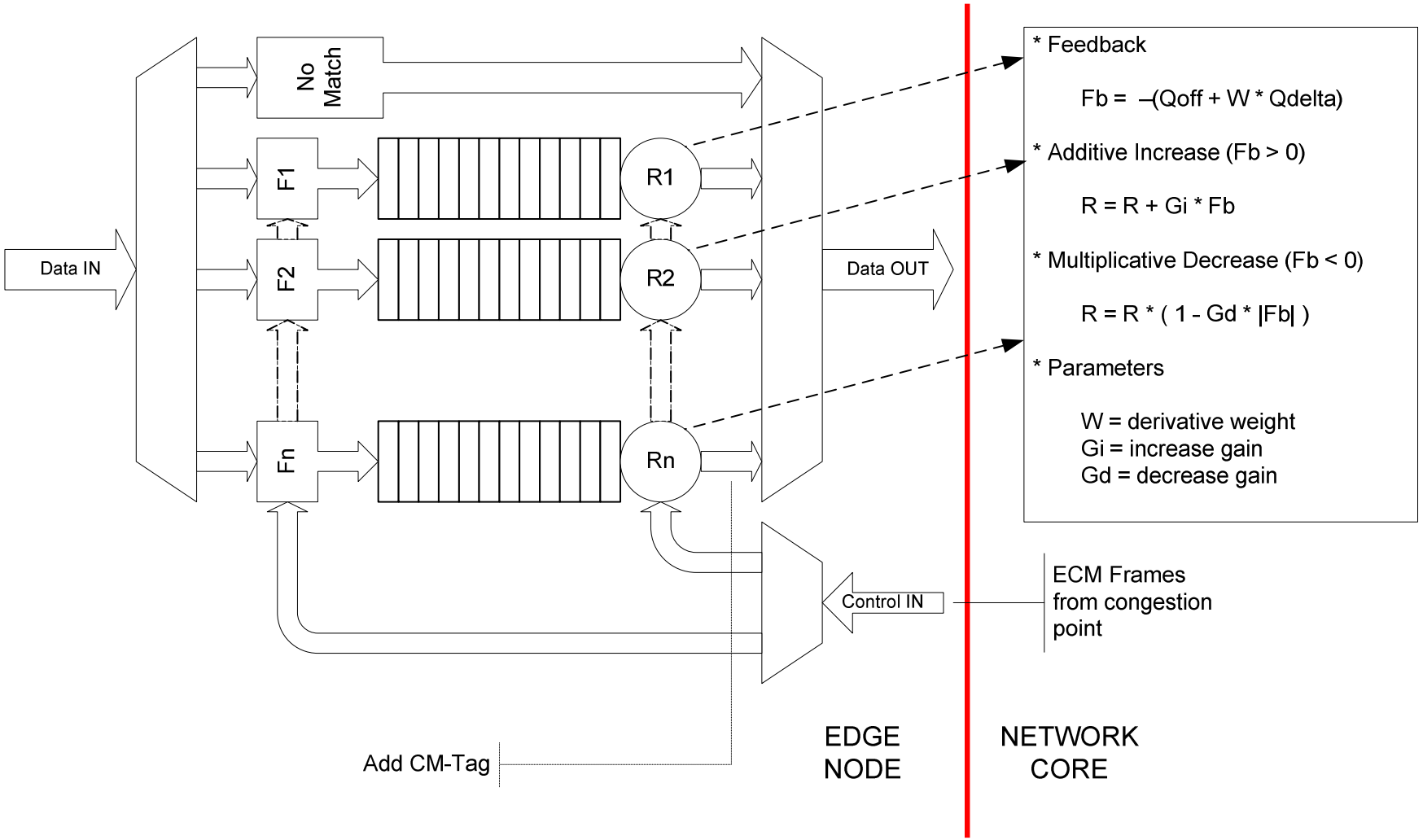
- Byte-based Sampling
  - P desired frame sampling probability
  - E[L] is the average frame length
  - Sampling interval is S = E[L] / P
  - E.g., P = 0.01, E[L] = 1 KB ➔ Sample a frame every 100 KB received
- To avoid bias, small random component added to sampling interval
  - S = Sf + Sr ➔ E[S] = Sf + E[Sr] ➔ E[Sr] = E[L]/P – Sf
  - E.g., if E[S] = 100 KB and Sf = 90 KB ➔ E[Sr] = 10 KB ➔ Sr ∈ [0, 20] KB



ECM Concepts: Detection

- Congestion Point state and configuration variables
  - ***State***:
    - Qlen: current queue length
    - Qold: queue length at previous sample
    - Bytecount: # bytes arrived since last sample
  - ***Configuration***:
    - Qeq: equilibrium threshold
    - Qmc: medium congestion threshold (BCN-MAX trigger)
    - Qsc: severe congestion threshold (BCN(0,0) trigger)
    - Sf: fixed sampling interval
    - Sr: random sampling interval range

ECM Concepts: Reaction



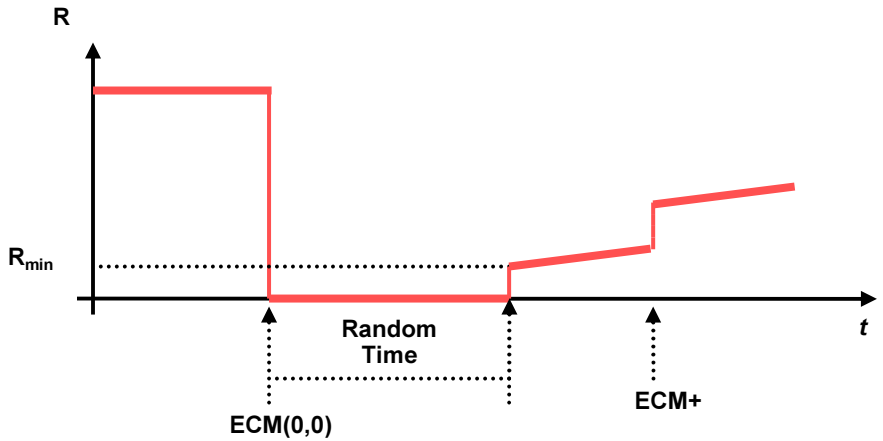
ECM Concepts: Reaction

- Rate adjustment

```
handle_bcn_frame( rate_lim, bcn_frame )
{
  Fb = calc_feedback( bcn_frame );
  if (Fb < 0)
  {
        rate_lim->R = rate_lim->R * ( 1 – min( - Gd * Fb, alpha ) );
        rate_lim->CPID = bcn_frame->CPID;
  }
  else
  {
        if ( bcn_frame.CPID == rate_lim->CPID )
        {
              rate_lim->R = rate_lim->R + min( Gi * Fb, beta );
        }
  }
}
```

ECM Concepts: Reaction

- Severe congestion reaction: ECM(0,0)
  - Current rate *R* is set to 0
  - Start random timer T $\in$ [0,T_{Max}]:
  - When timer T expires $R \leftarrow R_{Min}$
  - Next ECM(0,0) causes exponential back-off:
    - $T_{Max} \leftarrow T_{Max} * 2$ and $R_{Min} \leftarrow R_{Min} / 2$
  - Next positive feedback resets T_{Max} and R_{Min}



ECM Concepts: Reaction

- Self-increase (aka drift)
  - At regular intervals Td current rate R ← R + Rd
  - Purposes:
    - Speedup recovery from
      - ECM(0,0)
      - Loss of signaling from CP
    - Improve fairness
    - Reclaim a rate limiter at flow termination
- RTT estimator
  - $RTT_{avg} \leftarrow (1 - 2^{-Wrtt}) RTT_{avg} + 2^{-Wrtt} * RTT$
  - RTT may be used for adjusting ECM parameters as network condition change (still experimental)

ECM Concepts: Reaction

- Reaction Point state and configuration variables
  - ***State (per rate-limiter)***:

Req
- R: current rate
- CPID: current CPID

Opt
- RTTavg: last measured RTT
- Rmin: current minimum back-off rate
- Tmax: current maximum back-off time

  - ***Configuration***:

Required
- W: weight of derivative component
- Gd: decrease gain
- Gi: increase gain
- α: maximum rate decrease (fraction of current rate)
- β: maximum rate increase (fraction of link capacity)

Optional
- Wrtt: weight of the EWMA RTT filter
- Td: self-increase timer
- Rd: self-increase amount
- Rmin: minimum back-off rate
- Tmax: maximum back-of time

Questions?


CISCO