Details of the IKN-Fairness Mechanism: CQMA

Jon Schuringa, Günter Remsak, Harmen R. van As

September 2002

Vienna University of Technology

Contents

- Fairness operation
  - Access
  - Reservation
- Rate estimation
- Conclusions

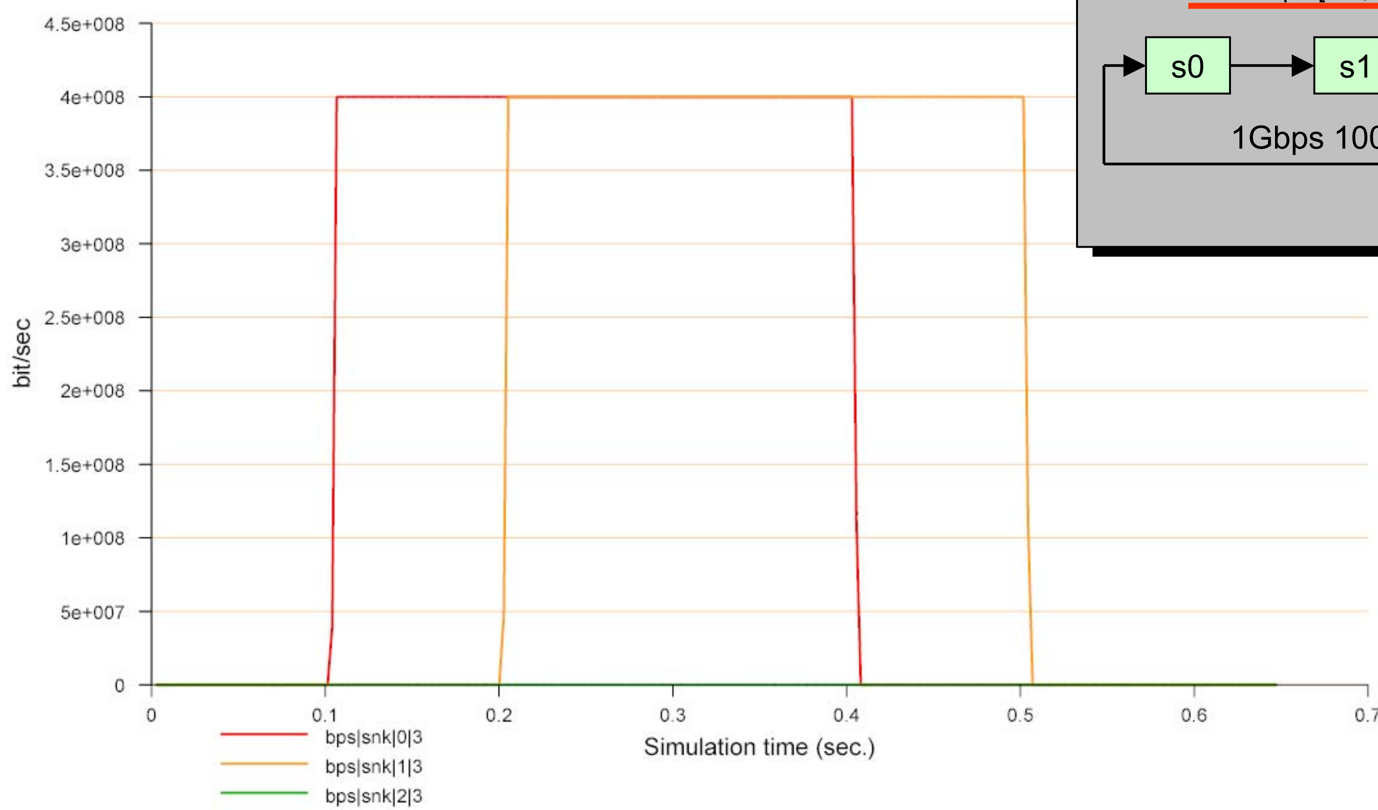
- Goals:
  - Demonstrate CQMA operation
  - Present the main ideas in jhs_clause9_01.pdf

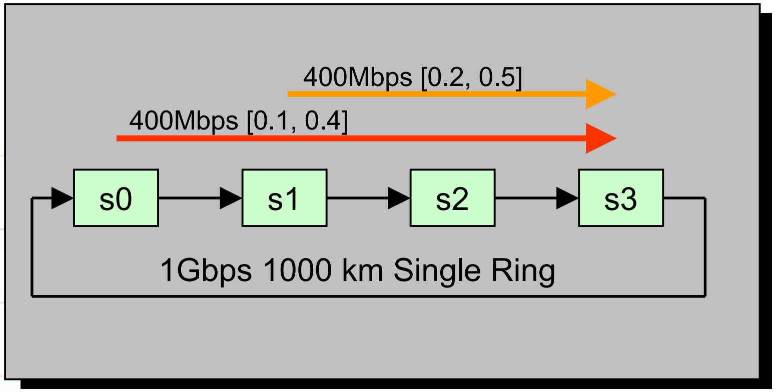
Fairness Operation

- Access:
  - Stations need a reservation for bottleneck links
  - Immediate access for non-bottleneck links
- Reservations:
  - Are done per cycle
  - Allocated by the fairness algorithm
  - Based on rate estimations or MAC-queue sizes

Immediate Access

- Immediate access to non-reserved capacity

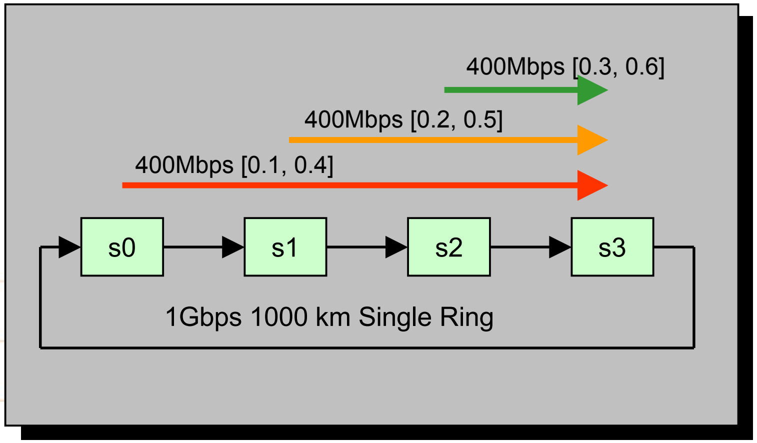


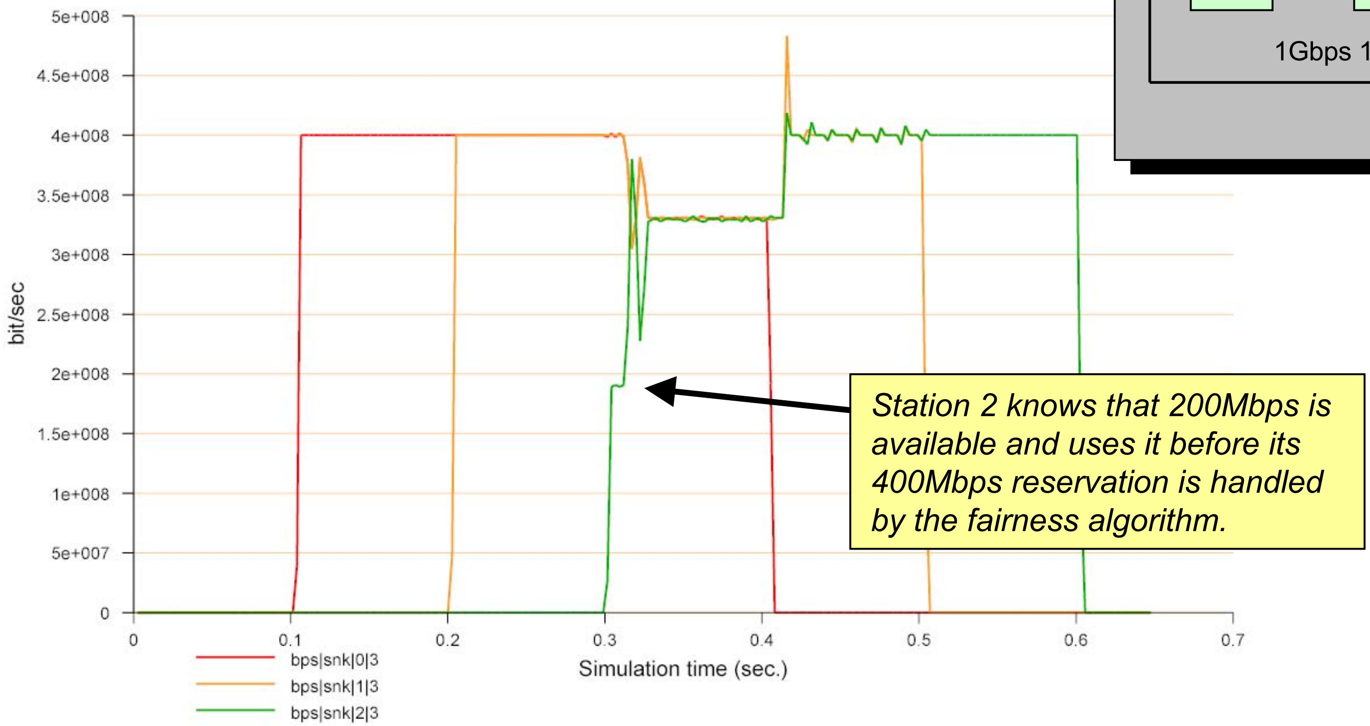


jhs_fairness_02.pdf

Reserved Access

- Reservation needed for the bottleneck link

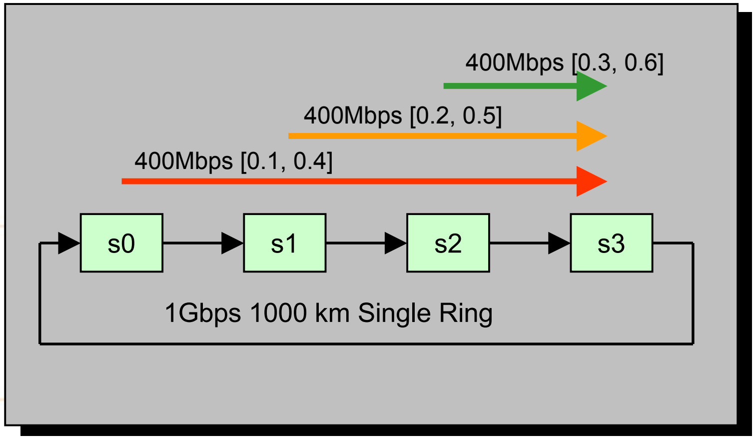


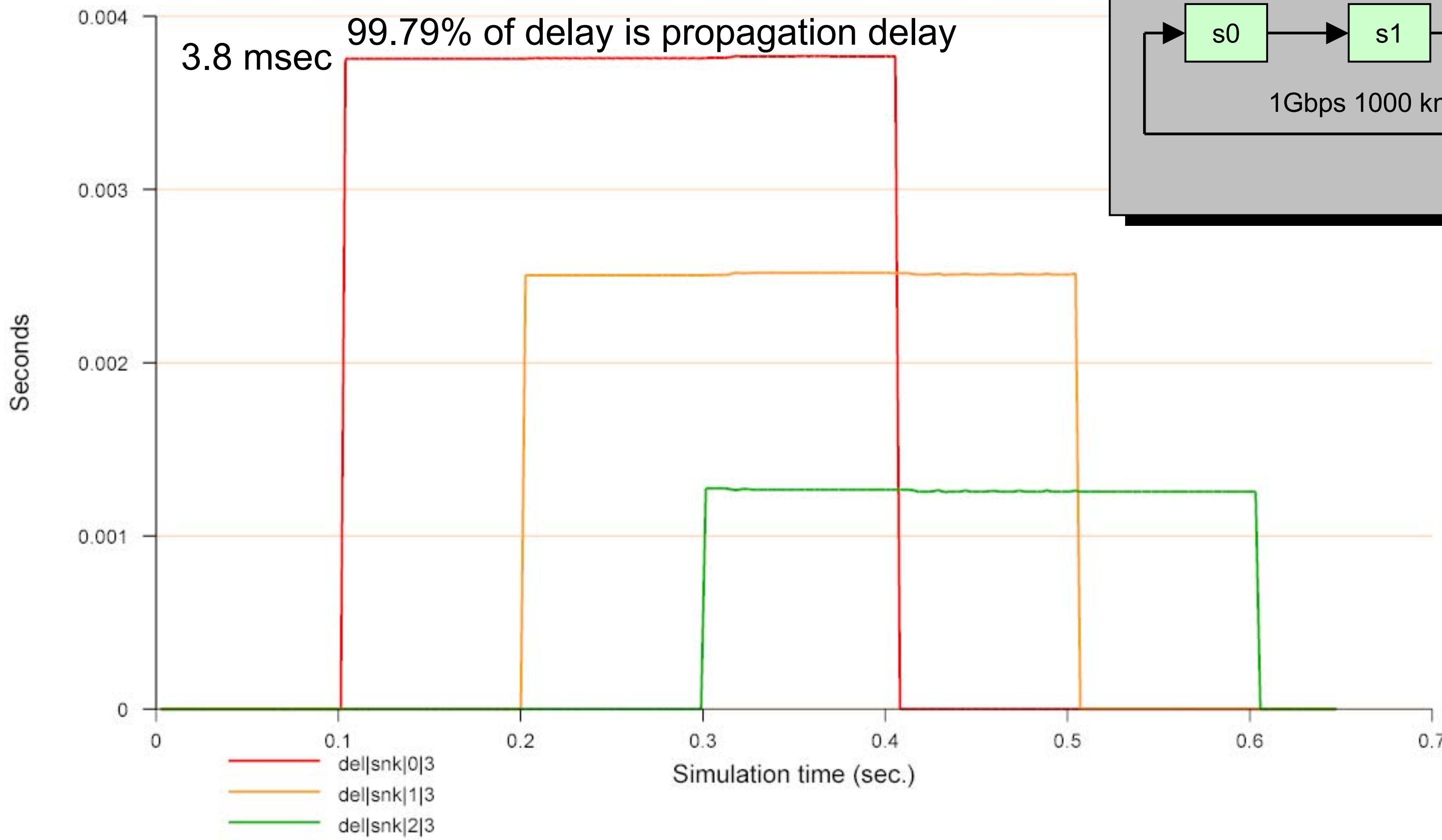


jhs_fairness_02.pdf

CQMA MAC end to end Delay (1000 km ring)

- No scheduling on transmission path → low delays

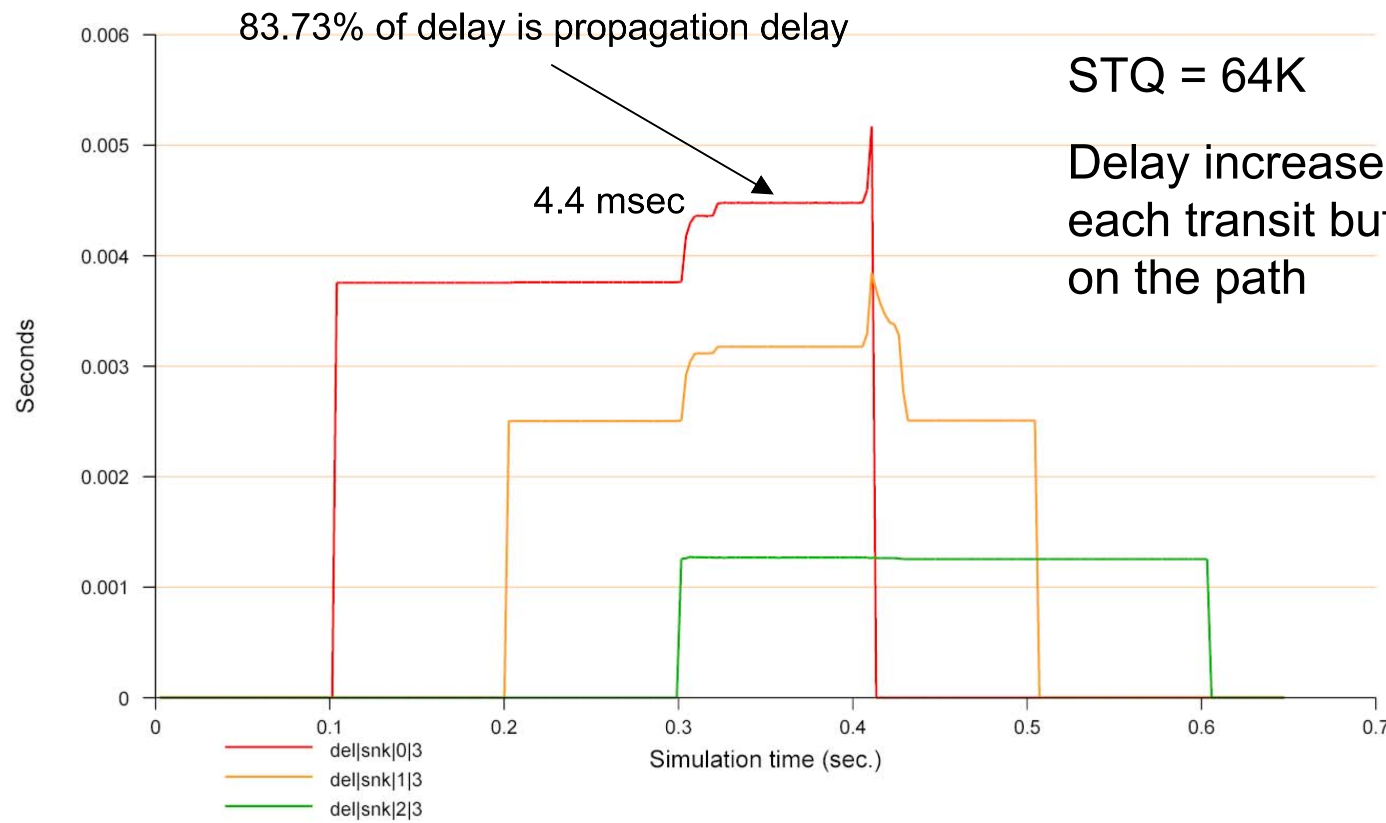




jhs_fairness_02.pdf

802.17 MAC end to end Delay (1000 km ring)

- Same scenario with current draft P802.17/D1.0 :



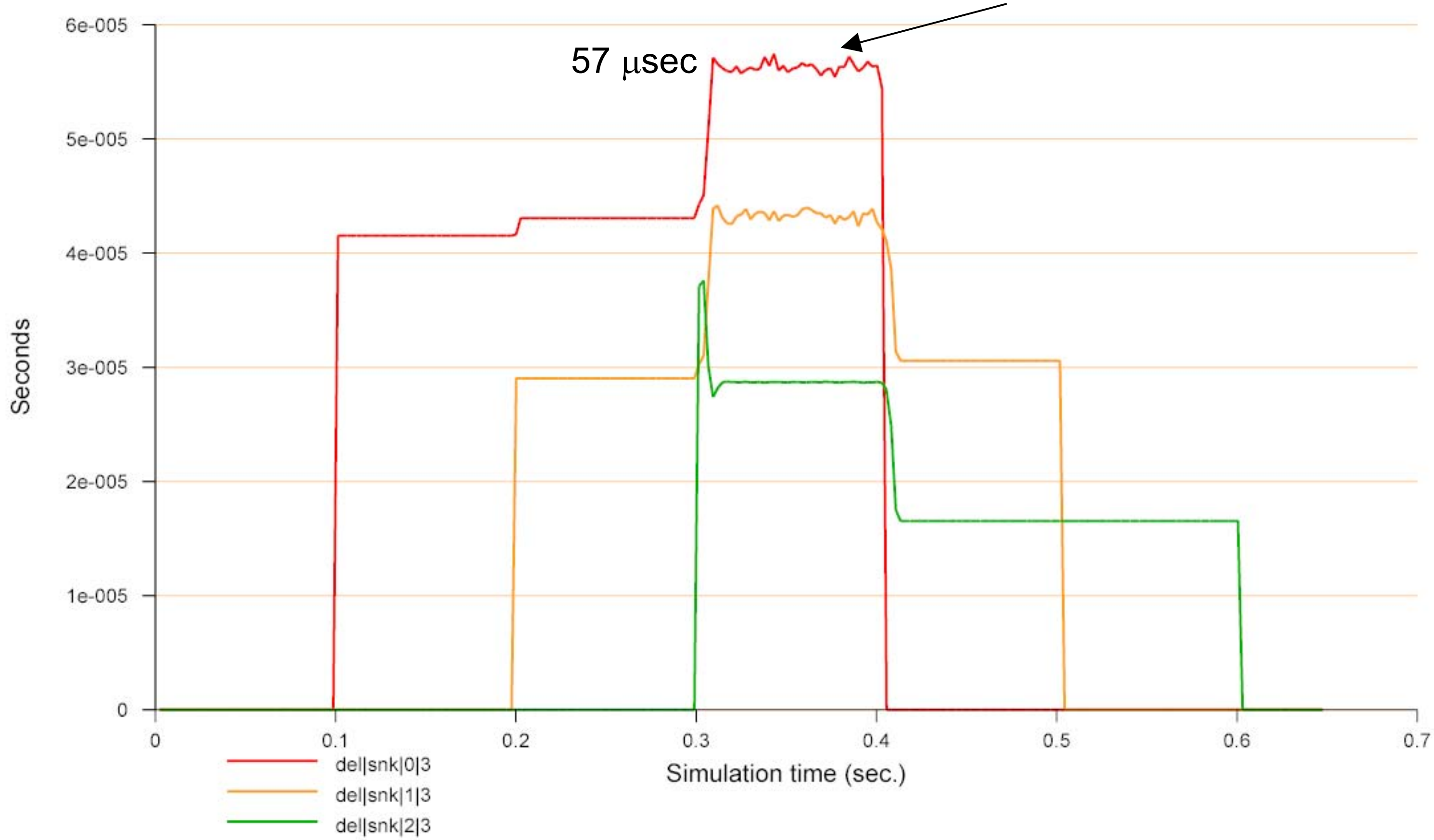
STQ = 64K

Delay increases for each transit buffer on the path

jhs_fairness_02.pdf

CQMA MAC end to end Delay (10 km ring)

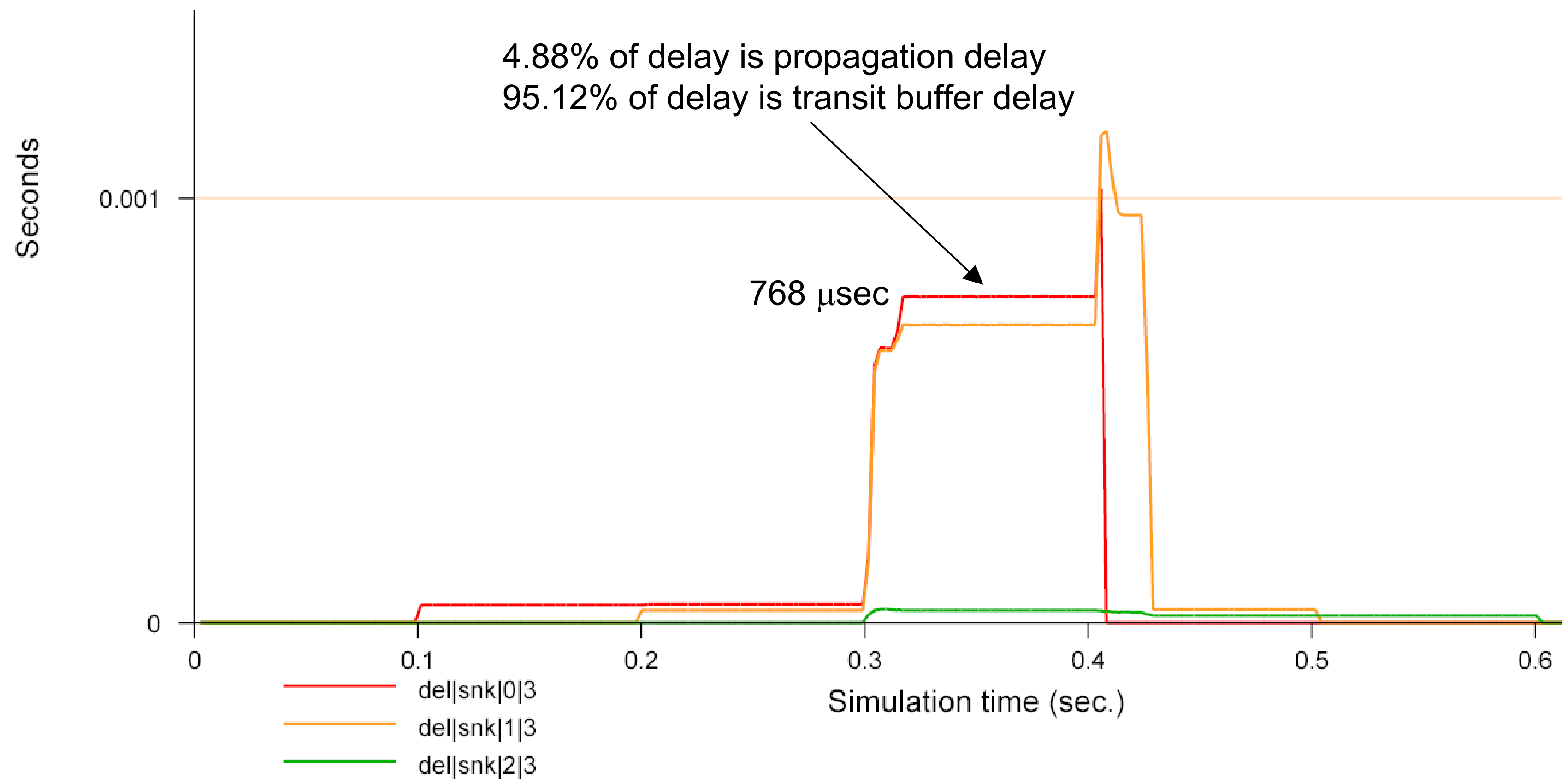
67% of delay is propagation delay
33% of delay is transit buffer delay



jhs_fairness_02.pdf

802.17 MAC end to end Delay (10 km ring)

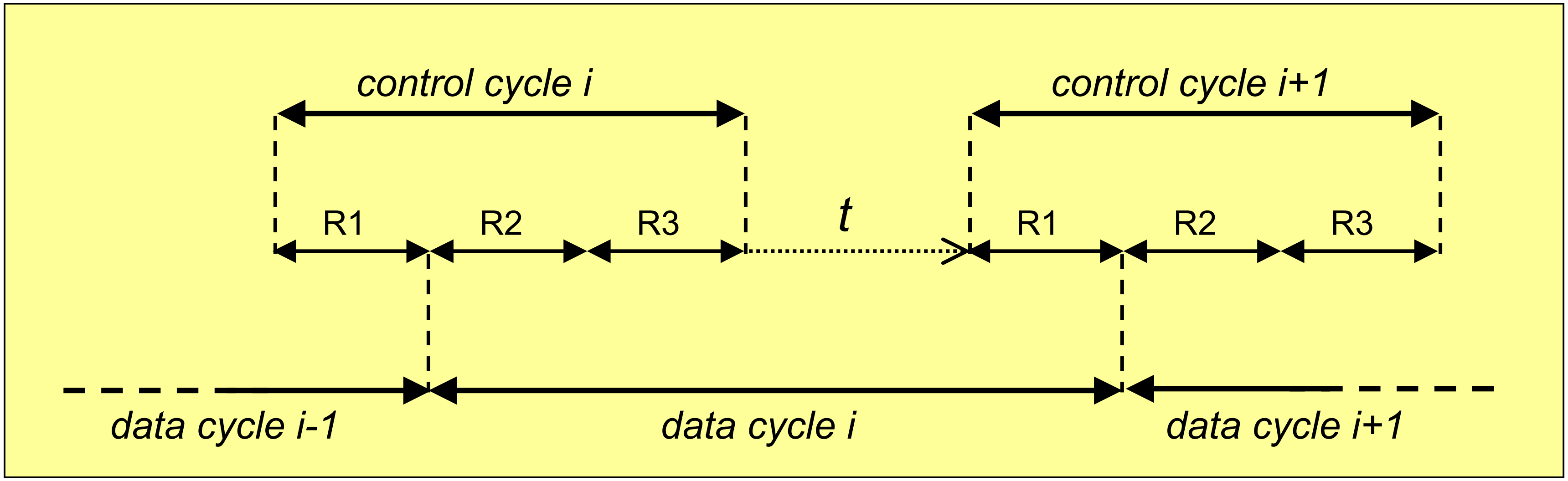
- Same scenario with current draft P802.17/D1.0 :



jhs_fairness_02.pdf

Fairness Operation: Reservation

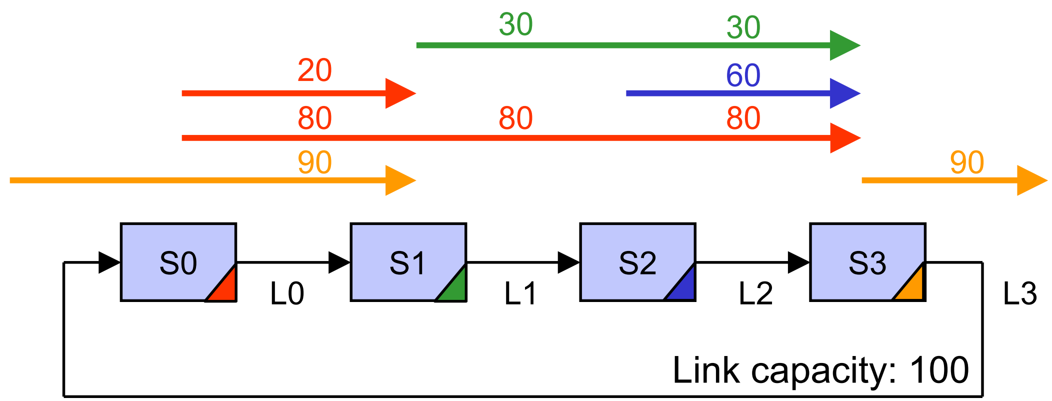
- Three rounds of the Fairness Control Message (FCM) are needed for each control cycle:
  - R1: Information round
    - Uses rate estimation and/or real queue size
  - R2: Calculation round
  - R3: Remaining capacity advertisement round (used for immediate access)
- Successive control cycles may overlap (*t* < 0), but must have at least 1 RTT distance to each other



jhs_fairness_02.pdf

Example

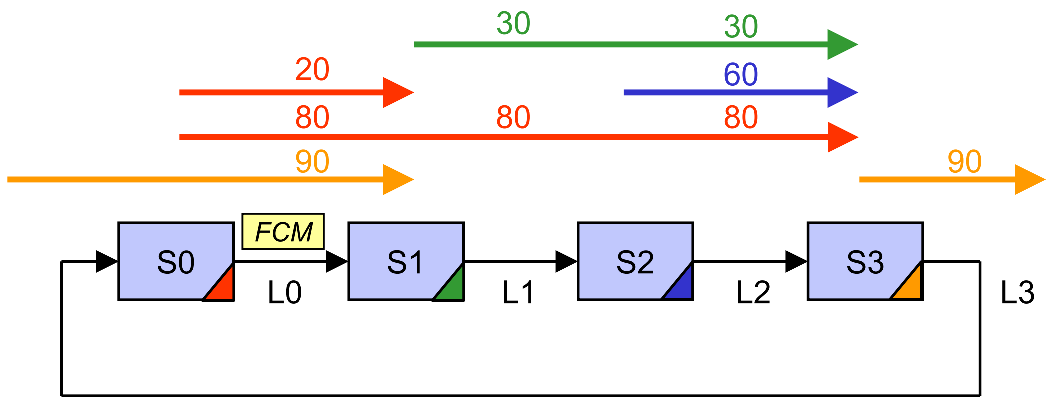
Scenario:



Empty FCM:

FCM	Demand	Remaining Capacity
L0		
L1		
L2		
L3		

Round 1: Information Round (cont.)

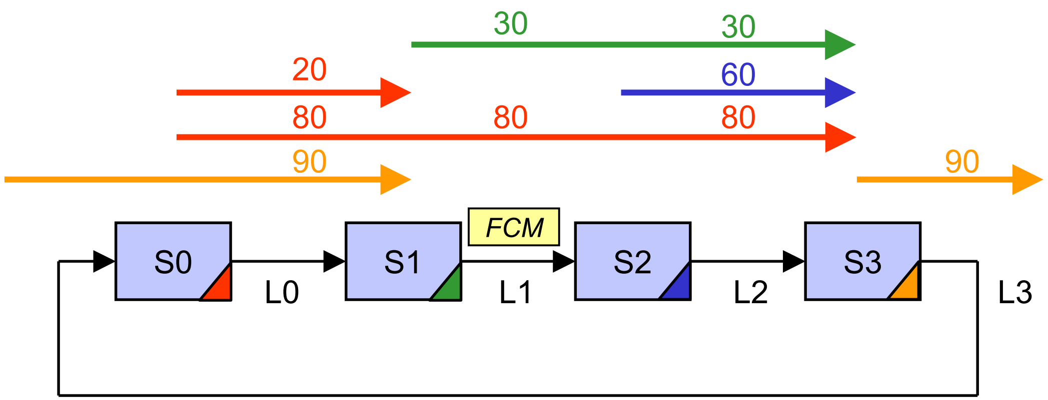


FCM	Demand	Remaining Capacity
L0	20+80	100
L1	80	
L2	80	
L3		

FCM leaving station 0

- Each station adds its own demand for each link to the corresponding demand field in the FCM
- The remaining capacity is set to the link capacity minus all provisioned traffic over that link
- Link capacities may differ on the same ringlet
- Small FCM size: N*2*2 bytes = 1024 bytes for 256 Stations.

Round 1: Information Round (cont.)

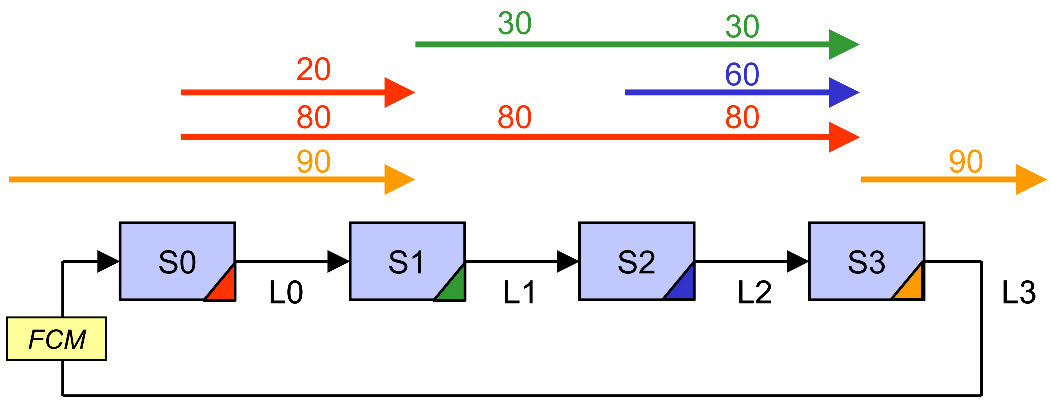


FCM	Demand	Remaining Capacity
L0	20+80	100
L1	80+30	100
L2	80+30	
L3		

FCM leaving station 1

jhs_fairness_02.pdf

Round 1: Information Round (cont.)



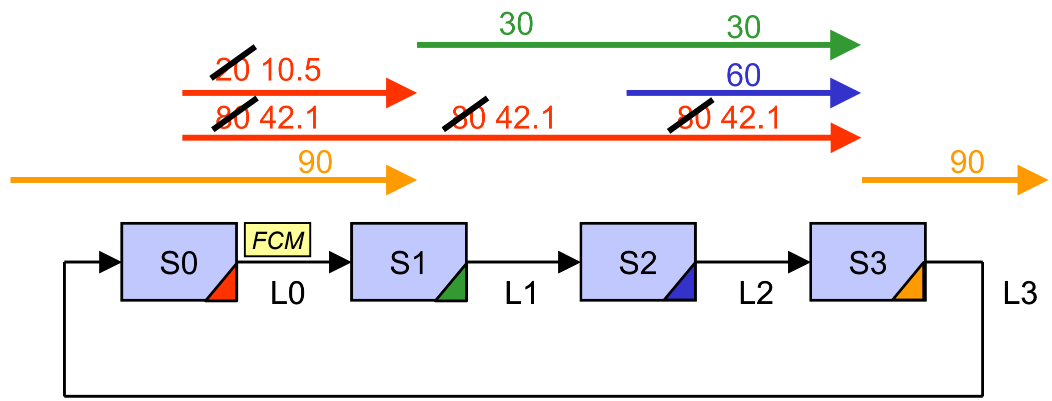
FCM	Demand		Remaining Capacity	
L0	20+80+90	= 190	100	☆
L1	80+30	= 110	100	☆
L2	80+30+60	= 170	100	☆
L3	90	= 90	100	

FCM arriving at station 0

☆ Bottleneck links

Round 2: Calculation Round

- Function: makeFair()
  - Calculates the fair rates
  - Updates the FCM
  - Supports source and flow fairness
  - Scalable
  - Predictable
  - Small (about 30 lines of C code)
- Each station forwards the updated FCM after calculation and can start/continue data transmission

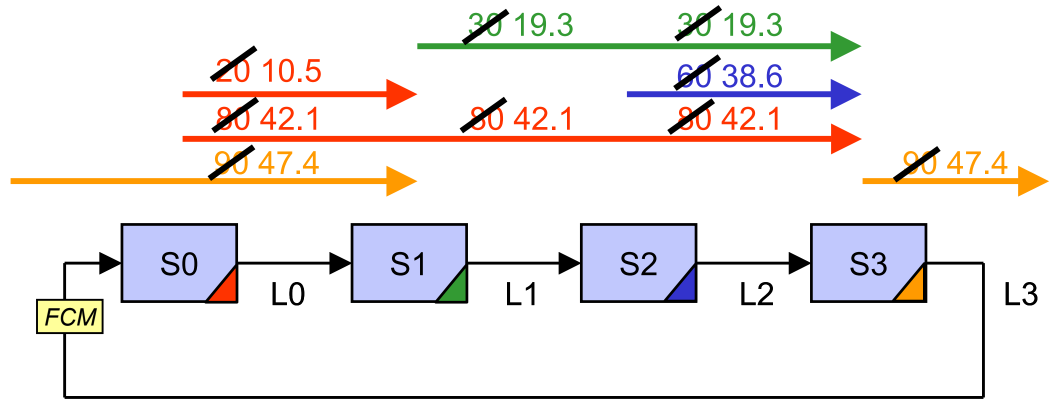


FCM	Demand	Remaining Capacity
L0	~~20~~+~~80~~+90 = 90	100-10.5-42.1 = 47.4
L1	~~80~~+30 = 30	100-42.1 = 57.9
L2	~~80~~+30+60 = 90	100-42.1 = 57.9
L3	90 = 90	100 = 100

FCM leaving station 0, after makeFair()

Round 2: Calculation Round (cont.)

- Calculation round finished:
  - All stations know their own fair rates to each destination.
  - Note that the stations don‘t know (and don‘t need to know) the fair rates of the other stations.

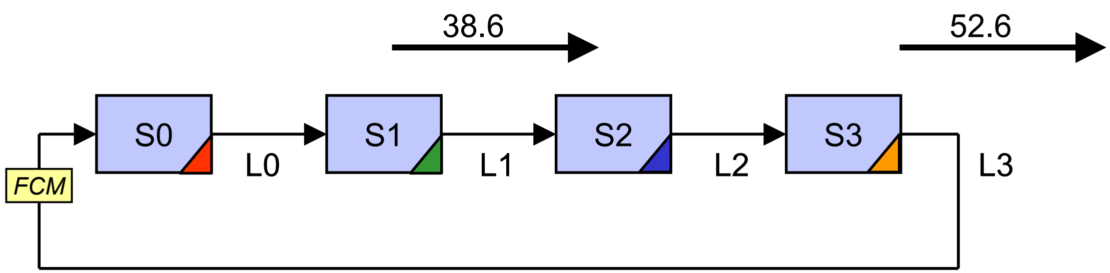


FCM	Demand	Remaining Capacity
L0	~~20~~+~~80~~+~~90~~ = 0	0
L1	~~80~~+~~30~~ = 0	38.6
L2	~~80~~+~~30~~+~~60~~ = 0	0
L3	~~90~~ = 0	52.6

FCM at the end of the calculation round

Round 3: Remaining capacity round

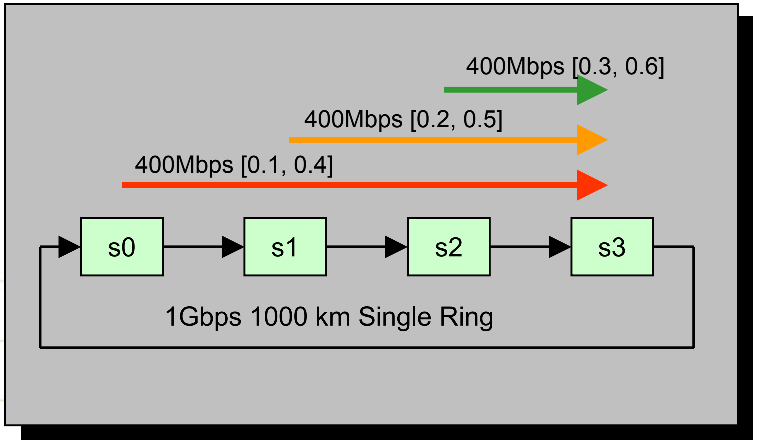
- FCM travels the last round to inform all stations about the remaining capacity
- Stations are free to use the remaining capacity for immediate access in the current cycle

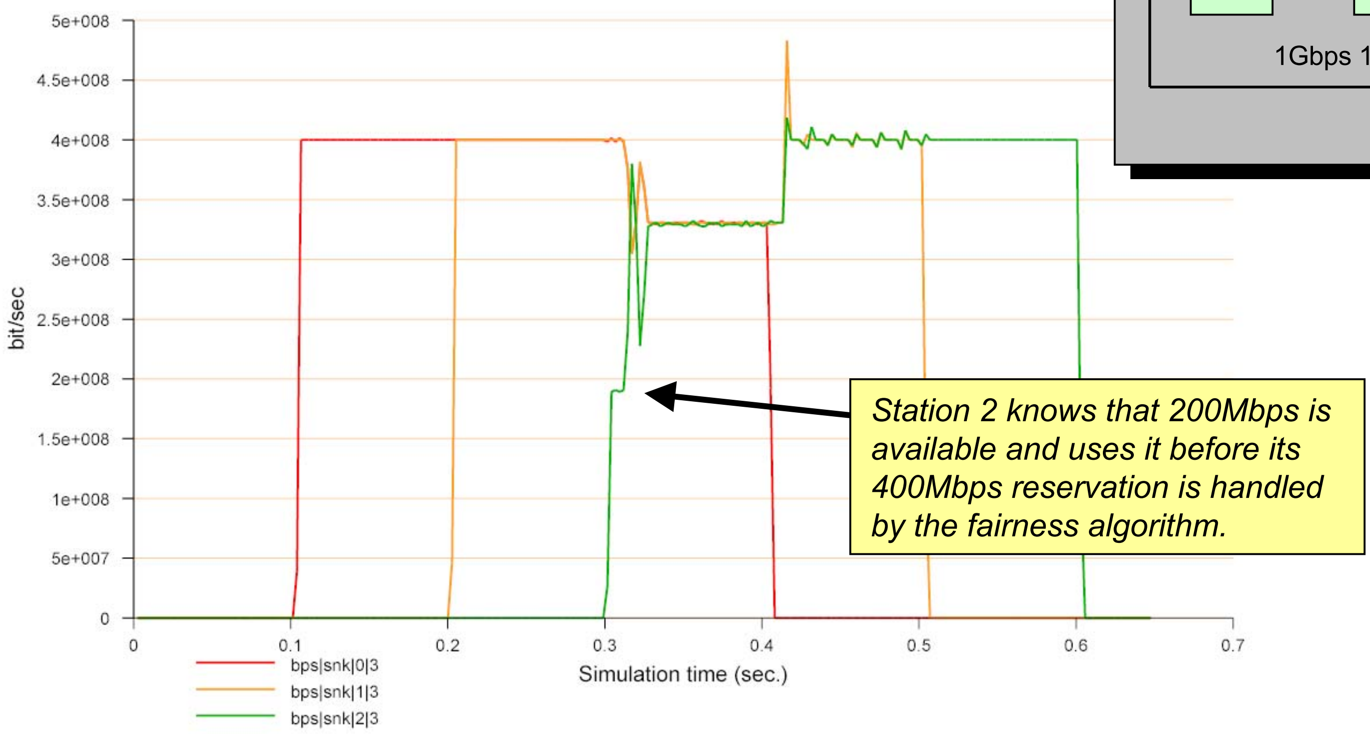


FCM	Demand	Remaining Capacity
L0	0	0
L1	0	38.6
L2	0	0
L3	0	52.6

Remaining Capacity Example (copy of slide 5)

- Reservation needed for the bottleneck link





jhs_fairness_02.pdf

Rate Estimation

- Local variables:
  - Estimated[#stations] : Array with the current estimation to each destination
  - Measured[#stations] : Array with the total amount of fairness eligible traffic sourced to each destination in the current cycle
- Executed at each stations at the end of each cycle:

```
If (isOutputQueueEmpty(dest))
    Estimated[dest] = Measured[dest]
Else
    Estimated[dest] = A * max( Estimated[dest], Measured[dest], B)

If (Estimated[dest] > MAX_ALLOW)
    Estimated[dest] = MAX_ALLOW
```

A and B are constants and control a trade-off between throughput and response time during transitions.

Conclusion

- Adapted CQMA:
  - Simple operation
  - To be combined with existing MAC protocol
  - Removes current performance weaknesses
  - Detailed description in jhs_clause9_01.pdf