



# RPR Bandwidth Management

---

Sanjay Agrawal: [Luminous](#)

Jeanne De Jaegher: [Alcatel](#)

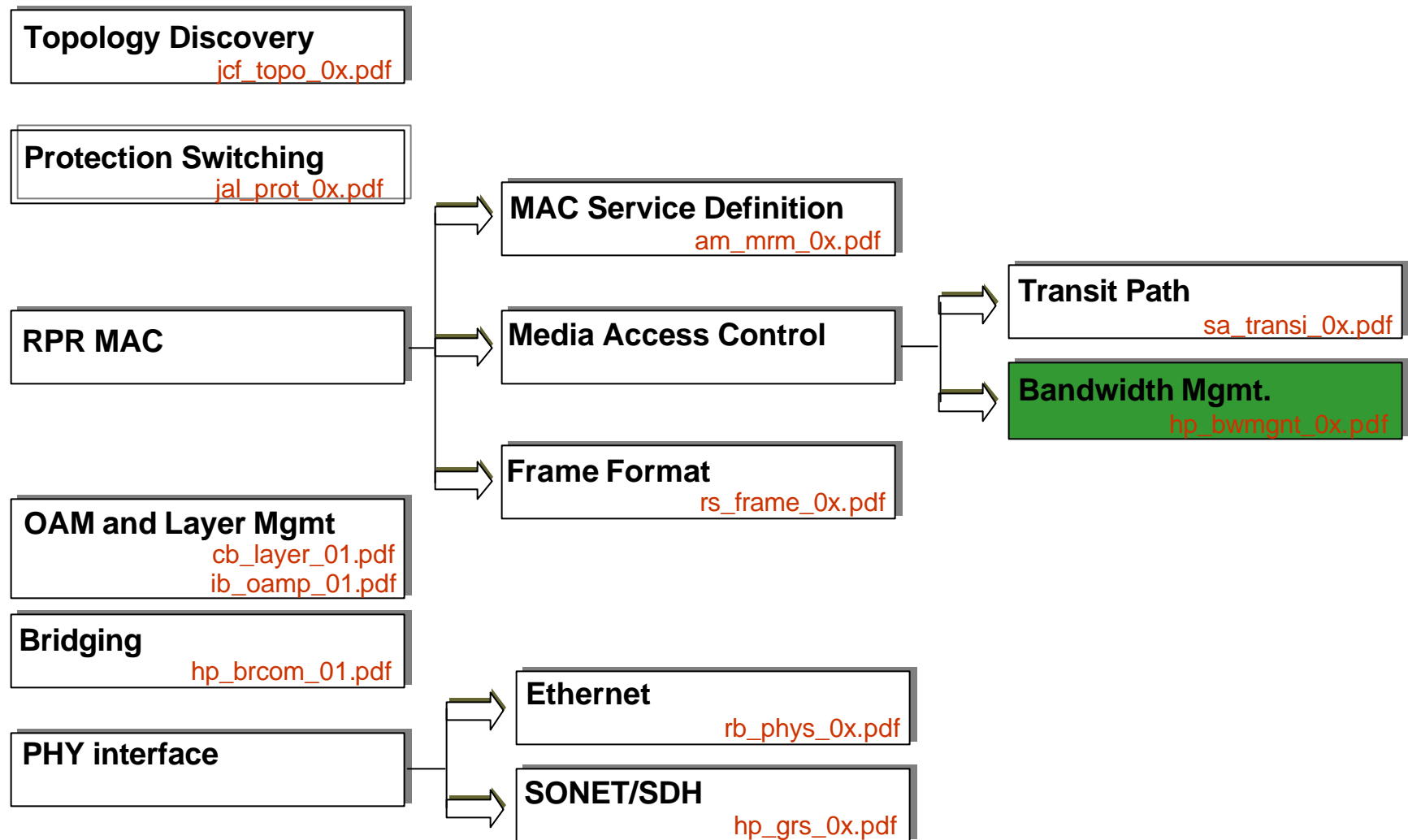
Henry Hsiaw: [NEC](#)

Adisak Mekkittikul (presenter): [Lantern](#)

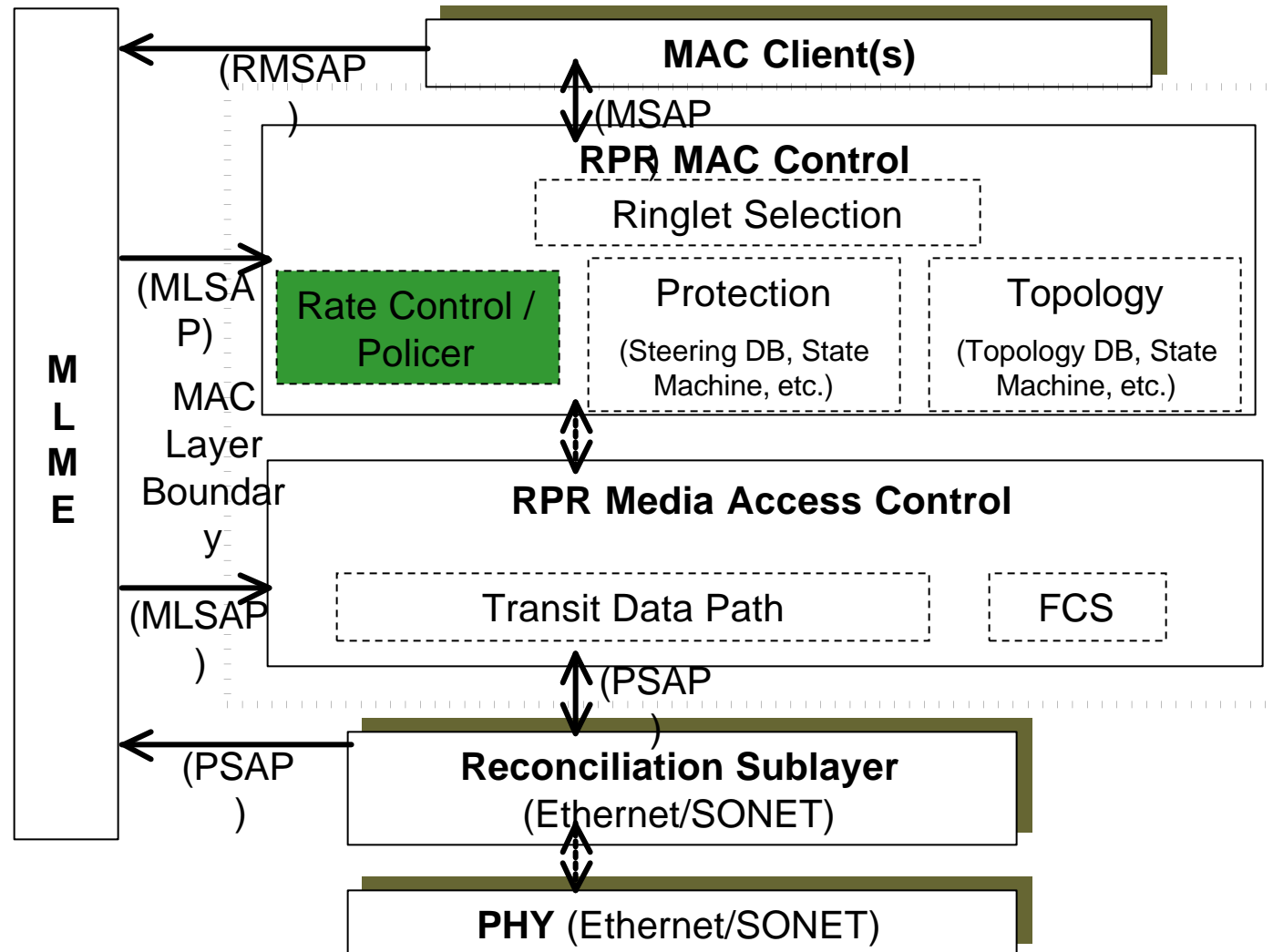
Robin Olsson: [Vitesse](#)

Harry Peng (presenter): [Nortel](#)

# Components of a complete RPR proposal



# Components of an RPR MAC





# Contents

---

- RPR Attributes
- Services
- BW Management Protocol
- BW Management Entities
  - Link BW Allocation Entity
  - Fairness Message Management Entity
  - Media Access Rate Policing Entity
- Rate Control Message Format
- Conclusions



# RPR Attributes

---

- Shared Medium
- Contention domain
- Source packet is temporal and spatial unaware of contention
- Parking lot problem
- Downstream disadvantage
- Spatial reuse

An 802.17 Objective: “Dynamic weighted BW distribution”



# Services

---

- Local Area and Metropolitan Area Networks
- Customers: Carriers and Service providers
- Applications: VPN, Internet Access, Future Apps.
- Support SLAs attribute is important for RPR
  - Delay, jitter, BW, and loss
  - Guaranteed Class
    - No BW reclamation
  - Committed Class
    - Excess BW are reclaimed
    - Committed BW may be reclaimed
  - Opportunistic Class
    - Best Effort

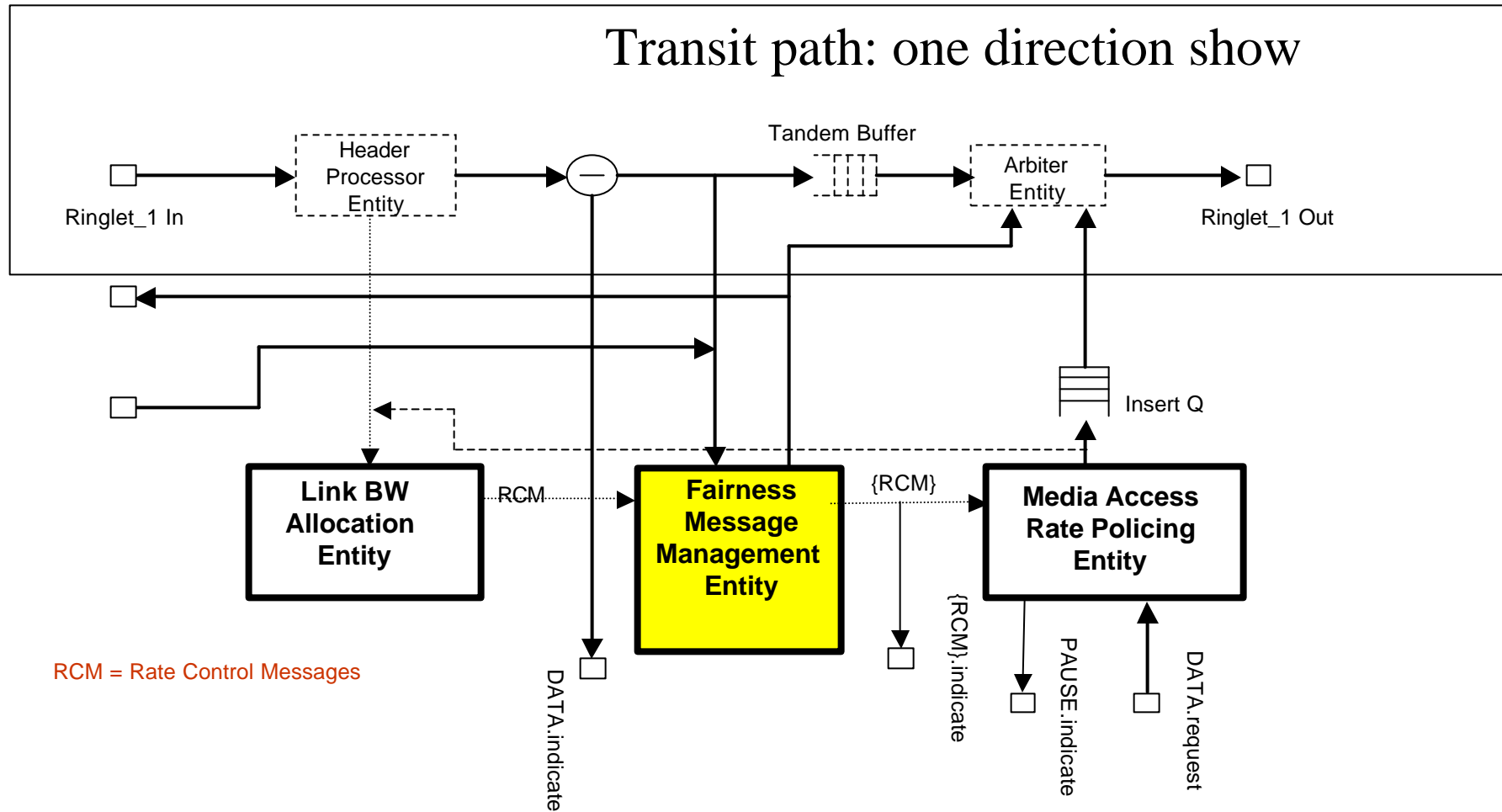


# RPR BW management

---

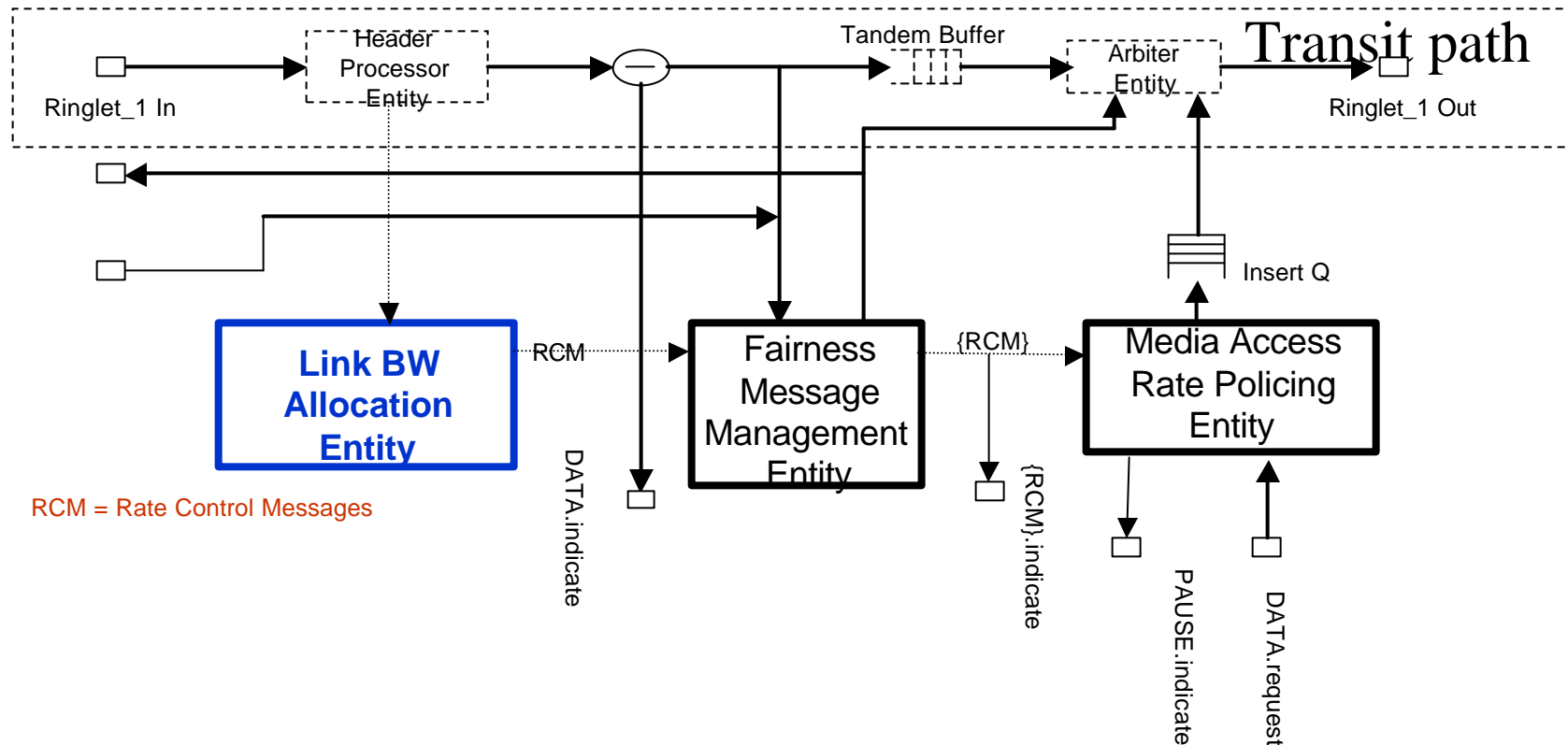
- Features (what it can do)
  - Support for weight fairness (unfairness), both Non-VOQ and VoQ
  - Support guaranteed, committed rate and best effort
  - Scalable to speed, size, multiple ringlets
- Performance (how well)
  - Fairness (index): BW, delay, loss
  - Utilization, Spatial reuse,
  - Convergence time (micro, milli-second, second)
  - Stability and robustness
- Complexity (at what cost)
  - Memories , logics, codes

# BW Management entities in MAC

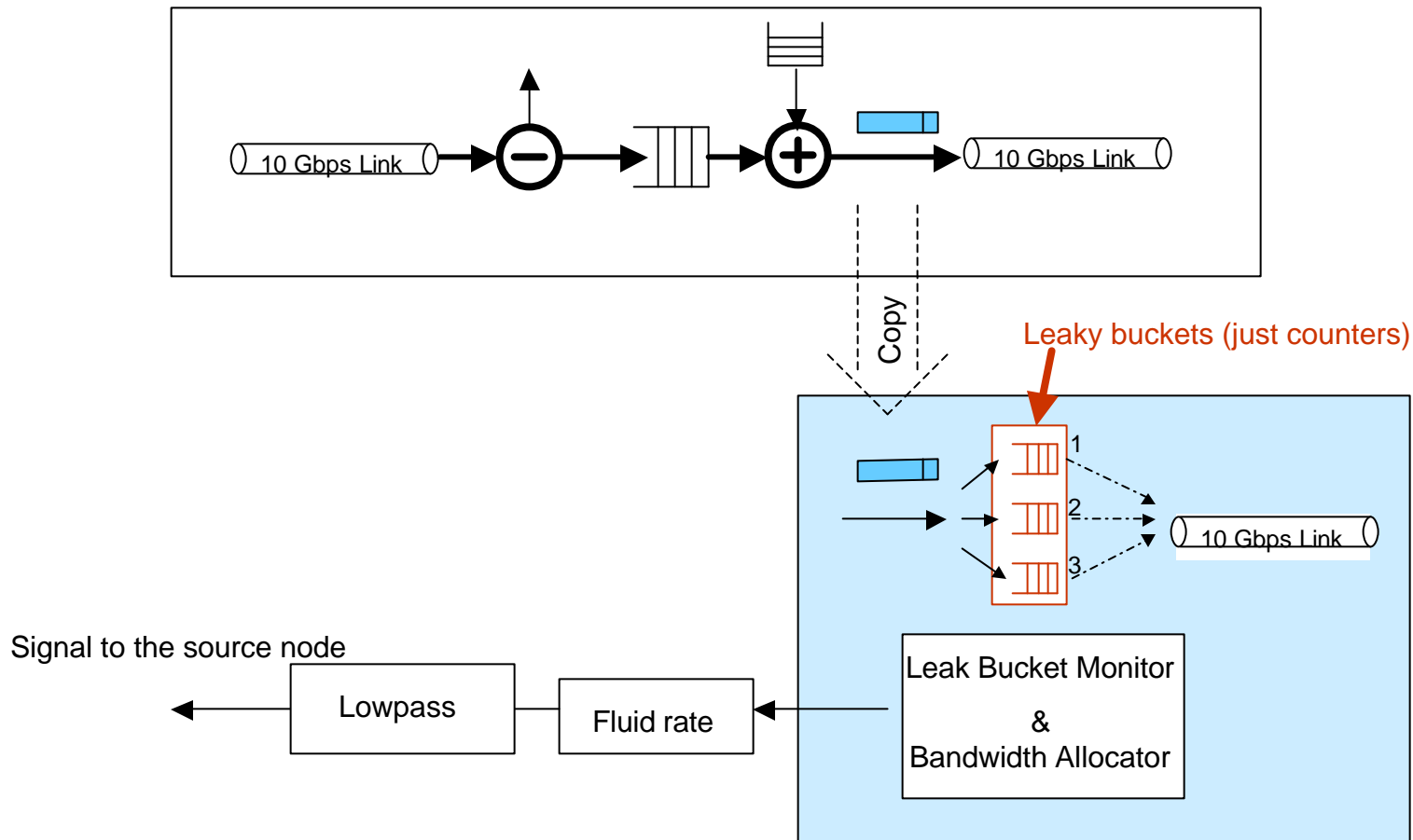




# BW Management Entities: Link BW Allocation Entity



1. Measure output link utilization:
2. Monitors active sources and passthru utilization





# BW Monitor

## @packet arrival

bucket(source node) += packet length

## @calc\_interval (1us)

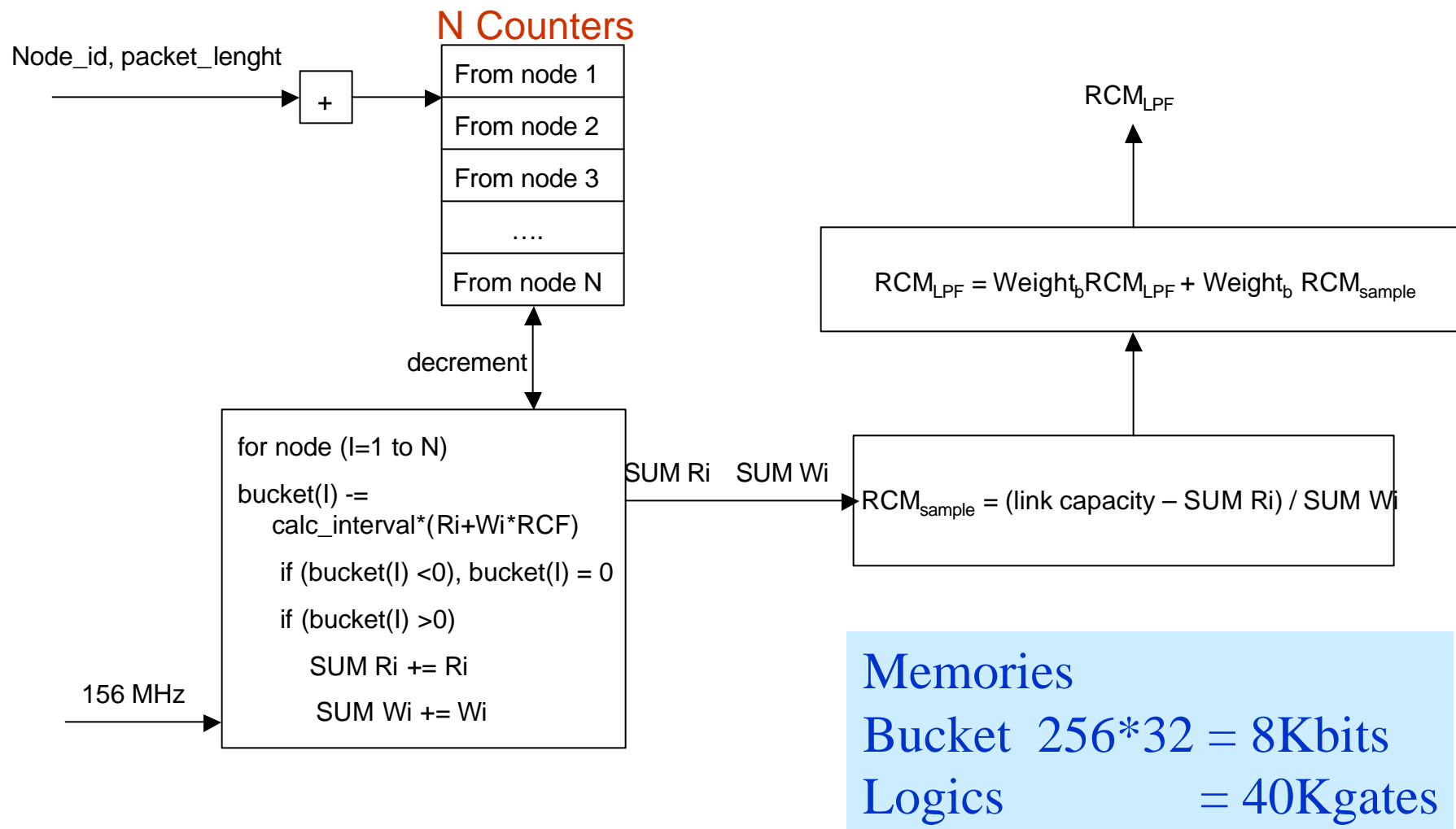
```
for each source node (l=1 to 256)
  if (bucket(l) >0)
    SUM Ri += Ri
    SUM Wi += Wi
  //drain bucket(i)
  bucket(l) -= calc_interval*(Ri+Wi*RCF)
  if (bucket(l) <0), bucket(l) = 0
end FOR
RCF = (link capacity – SUM Ri) / SUM Wi
SUM Ri = 0
SUM Wi = 0
```

## Register File

Ri 16 bits	wi 16 bits	Leaky Bucket 32 bits
1		
2		
.		
.		
.		
256		

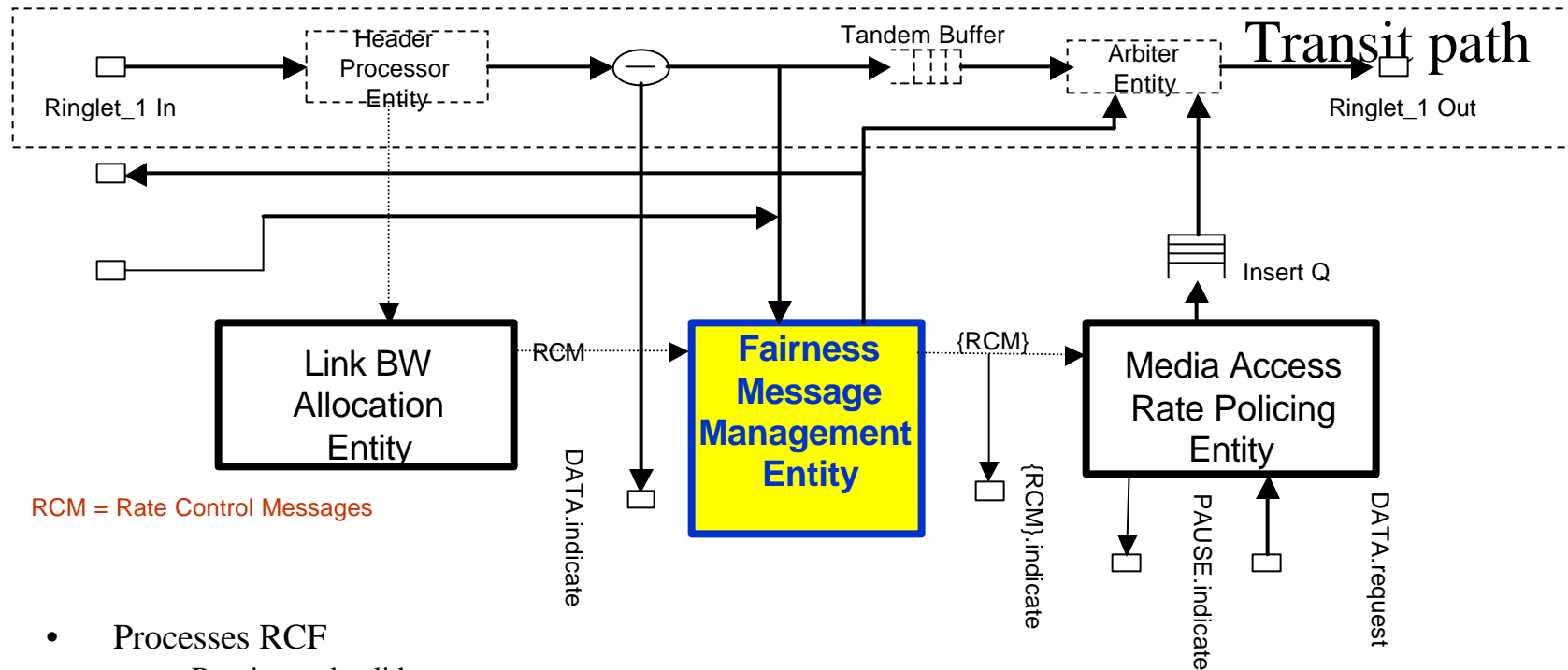
SUM Ri
SUM Wi
RCF
Calc Interval
Link Capacity

# Implementation Complexity (Example)



# BW Management Entities:

## Fairness Message Management Entity



- Processes RCF
  - Receive and validate message
  - Calculates local RCF, weighted scheme
- Generates advertised RCF:
  - triggers
  - Soft state: Periodic transmit RCF



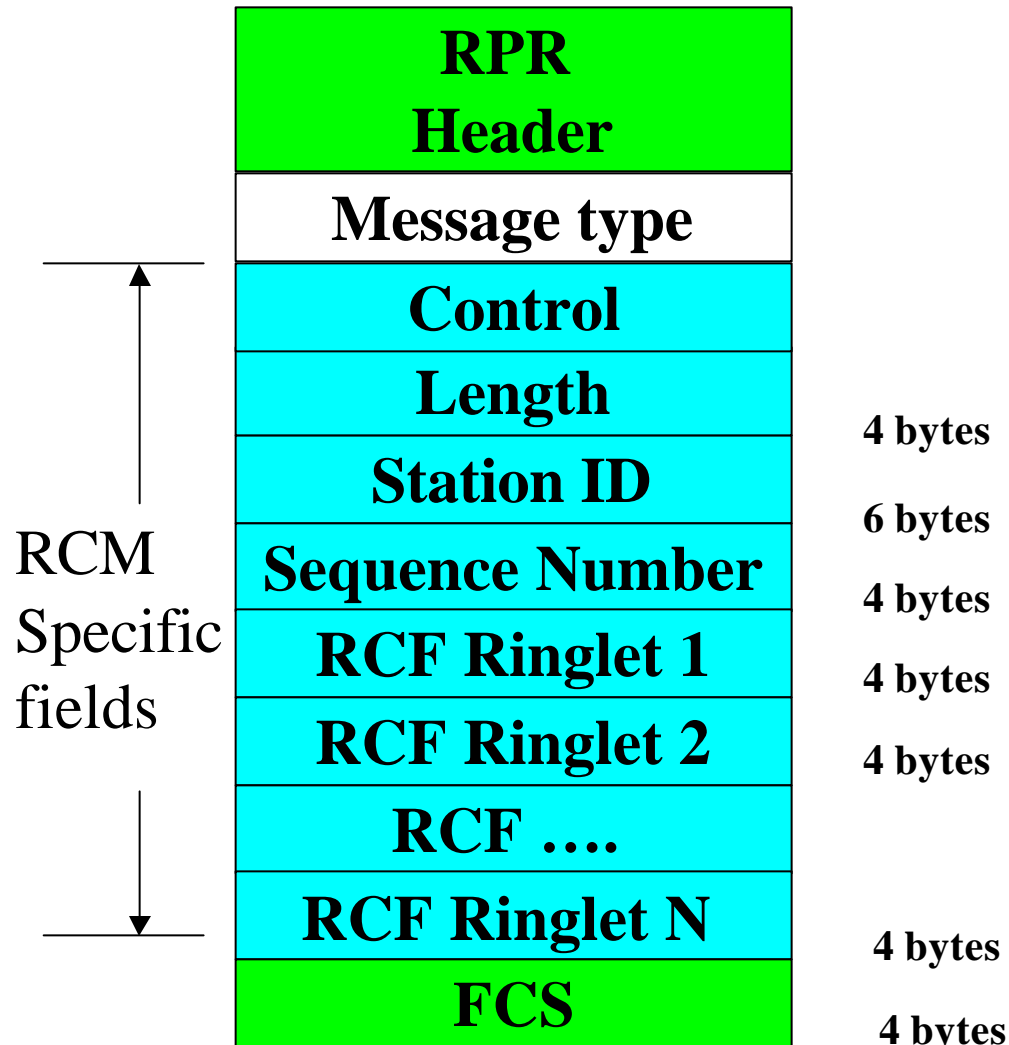
# Rate Control Factor

---

- Rate estimator on output link per source
- $RCF_{sample}$ : usage per sample window
- Filtering:

$$RCF_{(t)} = weight_1 RCF_{(t-1)} + weight_2 RCF_{sample}$$

# Rate Control Message Format

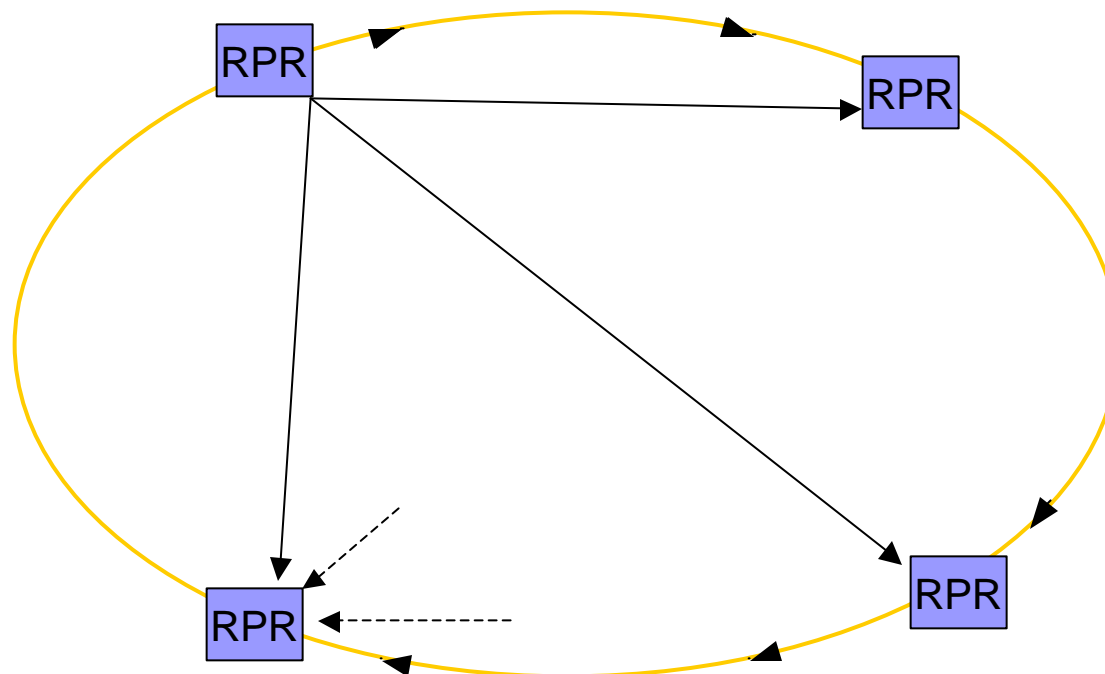


- **Common control frame and message type=RCM:**
  - Control: specific control bits-version etc.
  - Length: length of RCM packet
  - Station ID: packet source station address
  - Sequence number: message synchronization
  - RCF: rate control factors. One for each ringlet
  - FCS: error detection for RCM

# Broadcast RCM

---

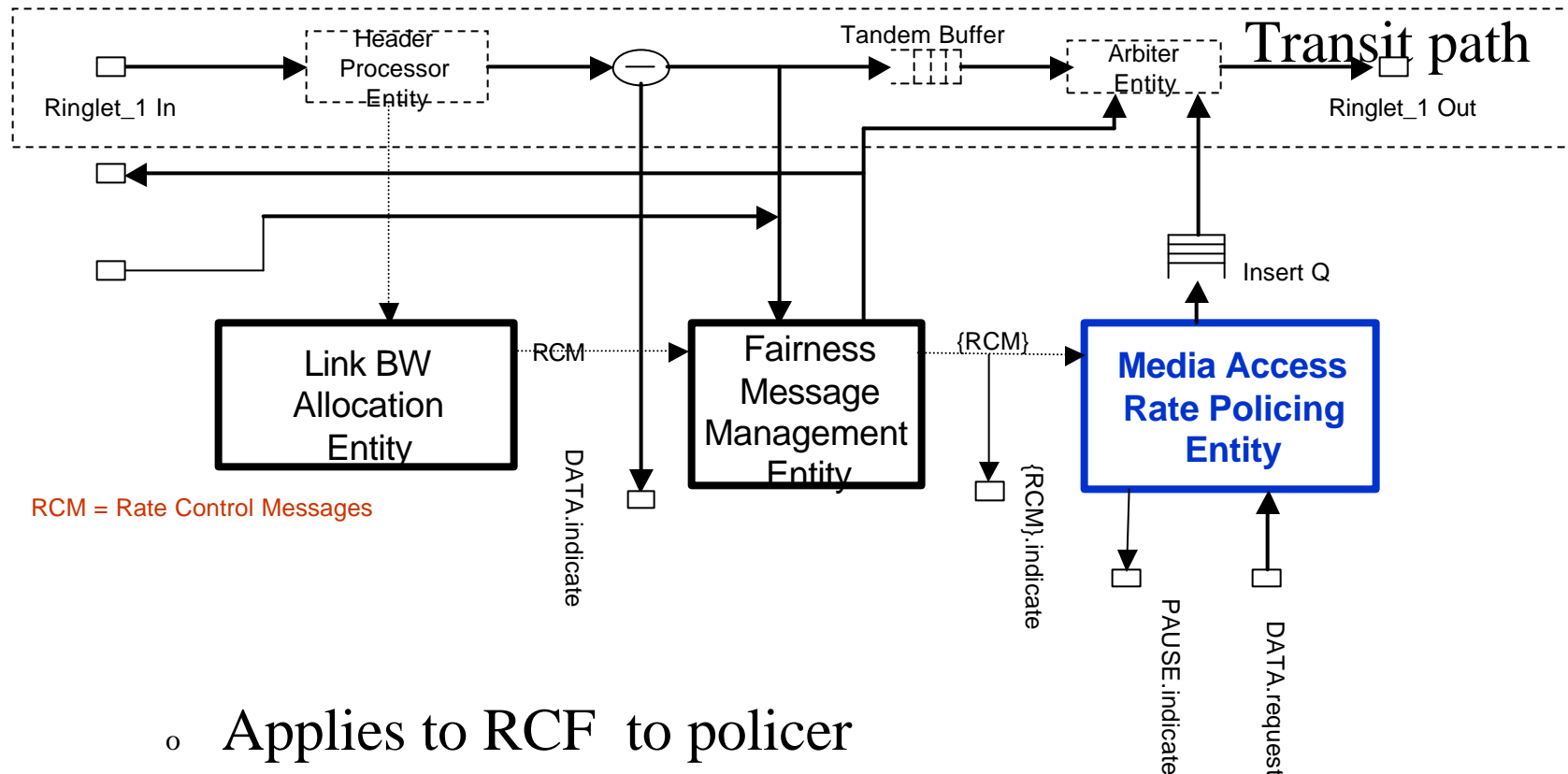
Fast and Robust against failures





# BW Management Entities:

## Media Access Rate Policing Entity



- Applies to RCF to policer
- Generates PAUSE.indicates when over transmit

# Media Access Rate Control (VOQ)

@ each pacing\_interval (256 clks)

```

for each link segment
  calculate the node (for this MAC) allowed BW, fj.
  fj = rj + wj*RCF
  give credit for each segment
  if ( segment_credit) < 1 MTU
    segment_credit += fj * interval_value
  if ( segment_credit) < 0 // client BW exceeds limit
    assert PAUSE.indicate
end FOR
  
```

@ each DATA.request

```

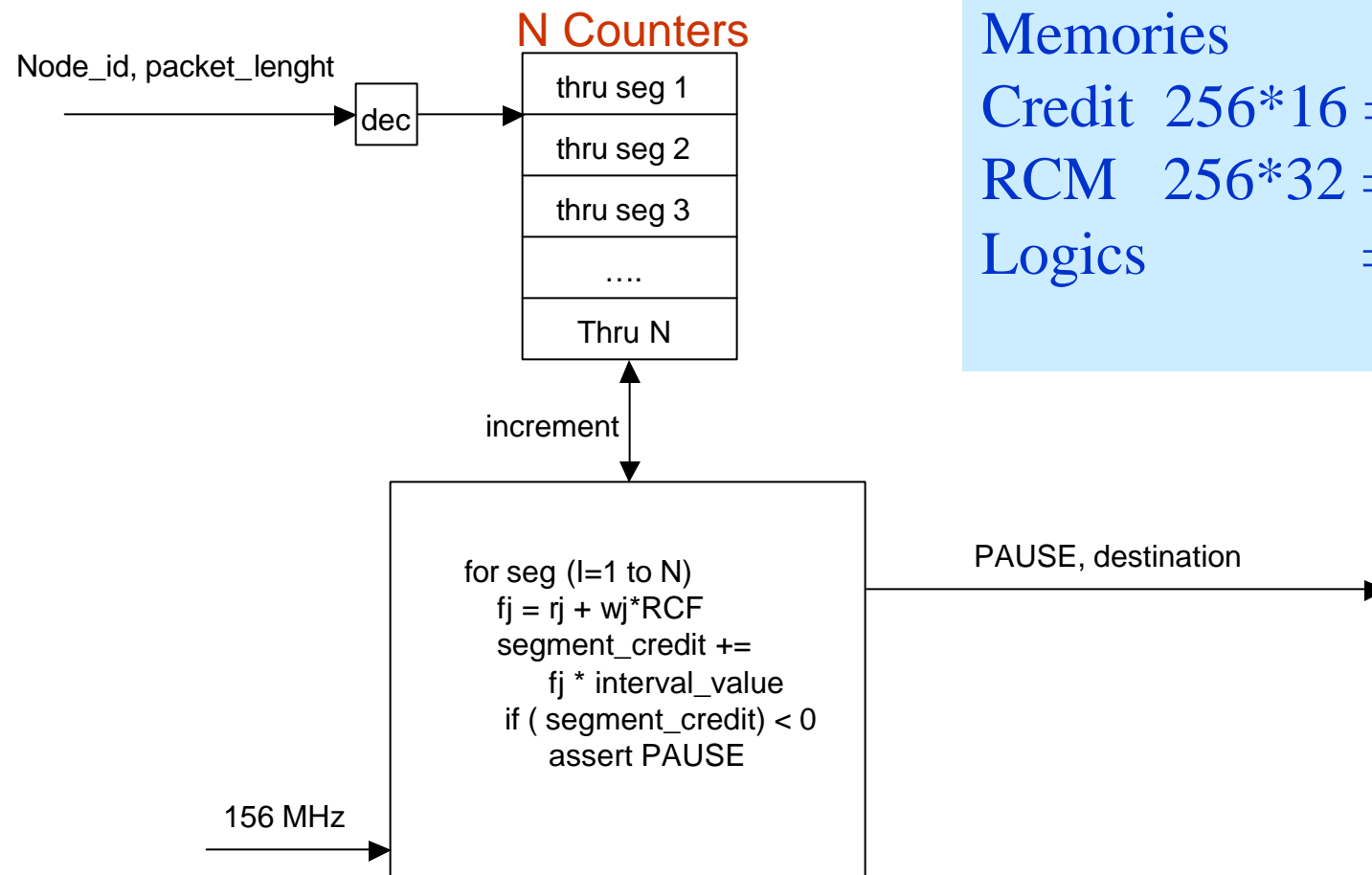
if no PAUSE.indicate asserted, the accept DATA.request
  for each segment between this and the dest nodes
    deduct segment credit
    segment_credit -= packet_length
  end FOR
  
```

Register File

RCM 32 bits	Segment Credit 16 bits
1	
2	
.	
.	
.	
256	

Rj
Wj
Interval_value

# Implementation Complexity



## Memories

Credit  $256 \times 16 = 4\text{Kbit}$

RCM  $256 \times 32 = 8\text{Kbit}$

Logics = 30Kgate



# Conclusions

---

- Started with a clean sheet
  - designed for carrier, service provider, and enterprise requirements.
- Complete functional BW management protocol that is simple, scalable and logical
  - Support a broader range of higher layer protocols
  - No HoL blocking
  - Stable, robust, predicable performance
  - Satisfies customer requirements



# RPR Bandwidth Management



## This Proposal:

Comprehensive features and capabilities for broad-applications: LAN, MAN, SAN, carriers, enterprise.  
Revolutionary and clean slate design to meet all requirements.

## Summary:

KISS: Simple and efficient (less memories, less logics, less cost)