Reactive Congestion Management (using Backward Congestion Notification)

Hugh Barrass (Cisco Systems)

Reactive Congestion Management

Why reactive? What target application?

- Prescriptive mechanisms (i.e. traffic management) not scaleable, require significant expertise
  - Needs predictable data flows
- Block data transfers (apparently random)
  - e.g. ftp, tftp, rdma, iSCSI, etc.…
  - Logically meshed topology (any source to any destination)
- Life of flow >> network latency
  - Otherwise reaction ineffective
  - Buffering requirement proportional to delay b/w product

BCN components

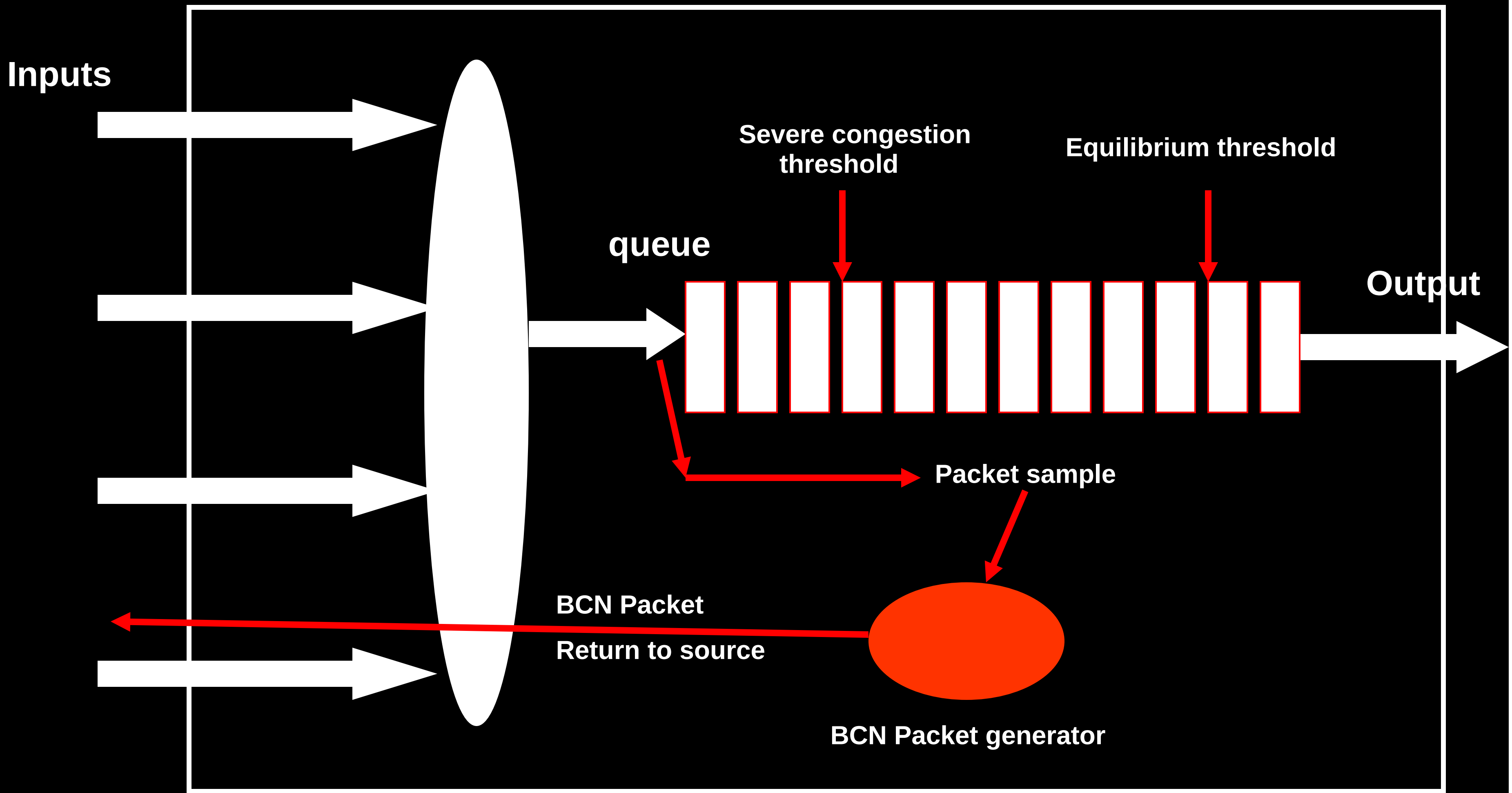
Congestion Point

Located in a bridge – where flows collide

- Queue structure unchanged from 802.1D / Q
  - CM operates orthogonally to priorities
- Number of queues unchanged
- New requirement for thresholds
  - Similar behavior to current QOS implementations
- New requirement to generate backward notification
  - Sample incoming traffic, generate packet on threshold
- New requirement to detect forward tagged packets
  - Some state change

Congestion Point

Component architecture



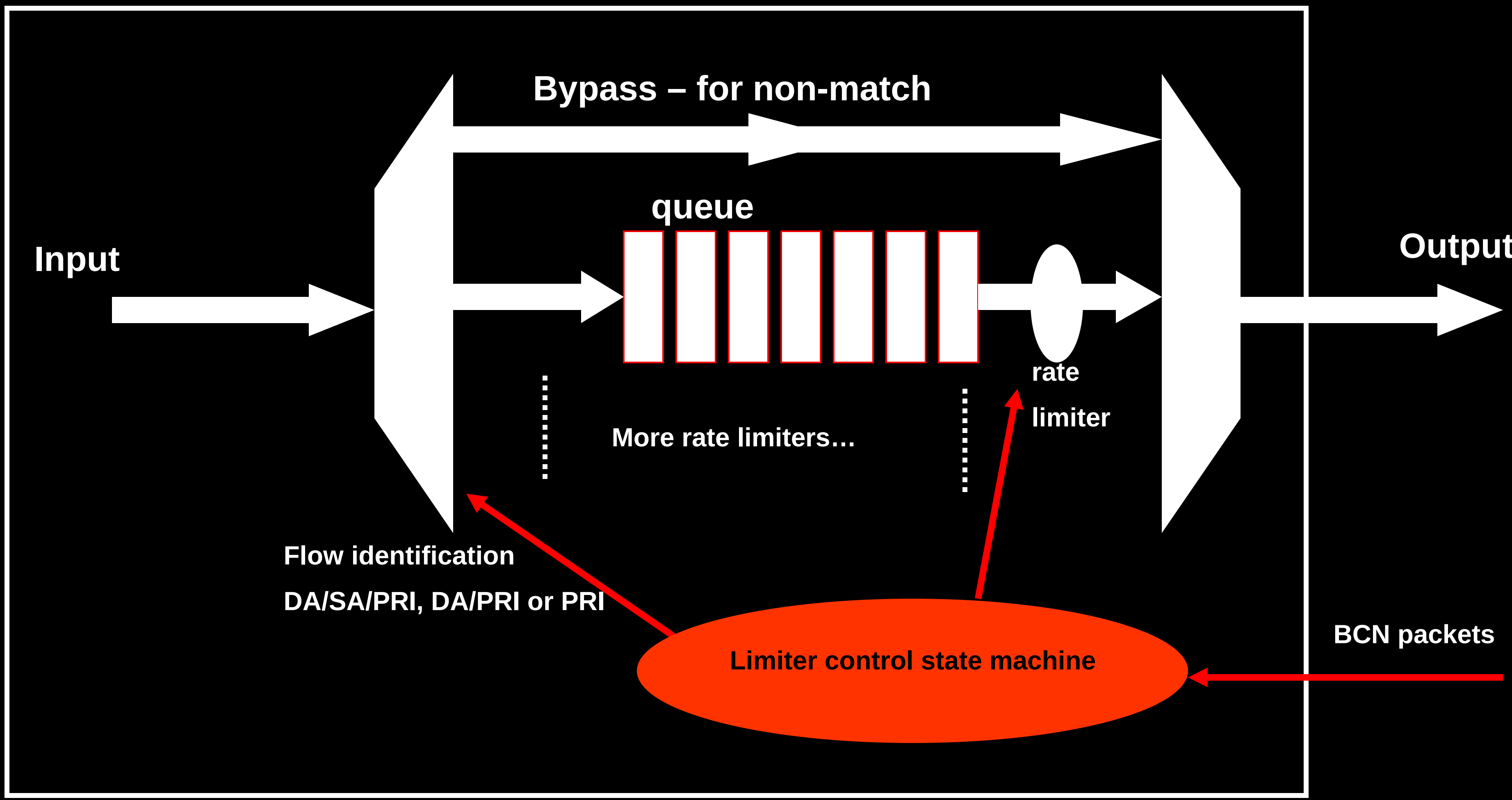
Reaction Point

Located at edge – where flows enter the network

- **New queue, with rate limiter mechanism**
  - **Multi-path (run around) may be needed**
- **State preserved, based on notifications received**
- **Granularity dependant on implementation**
  - **Could be SA/DA/PRI, DA/PRI, PRI-only, or entire link**
- **Suggest multiple rate limiters, with fall-back**
  - **React to multiple congestion points**
  - **If # congestion points exceeds # rate limiters…**
  - **… fall-back to coarser granularity**
- **More than 2 or 3 simultaneous congestion points unlikely**

Reaction Point

Logical architecture



Reaction Point

Logical and physical architecture may vary

- Best place for reaction point is in end station
- The reaction point may interact with data source
    - e.g. integration may allow application awareness of congestion
- Back-signaling from rate limiter may travel up the OSI stack
- Alternative implementation may be in “edge of cloud”
    - Rate limiter in edge bridge would behave like constricted link
    - Could use WRED or mark-down or other congestion response…
    - … must tie in with external congestion management
- “Ideal” architecture always places reaction points as near to data sources as possible

How it works…

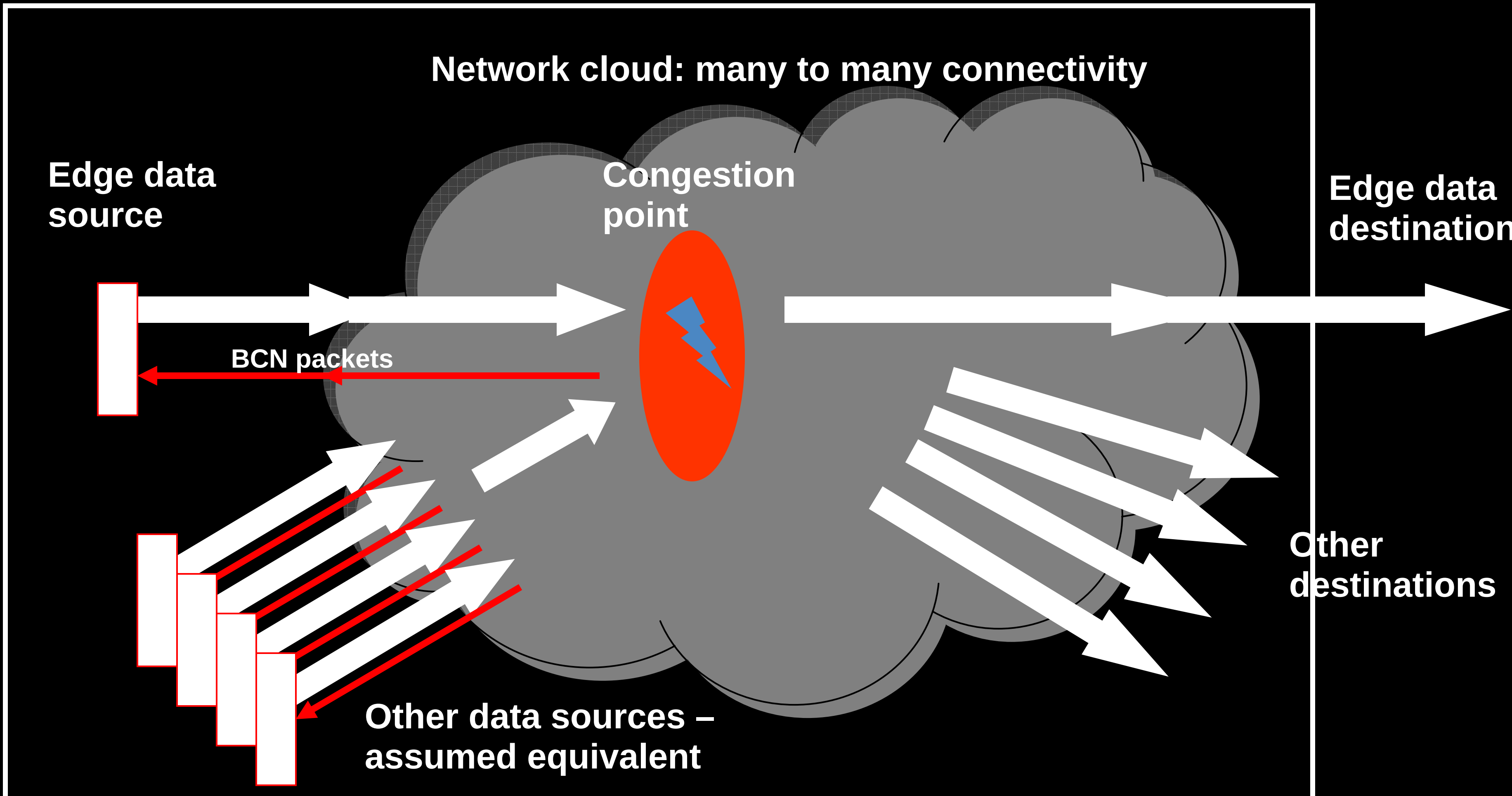
BCN behavior

Detailed simulations & analysis of 1 proposal

- **Davide's presentations have details – please reference them!**
  - http://www.ieee802.org/1/files/public/docs2005/new-bergamasco-bcn-july-plenary-0705.ppt
  - http://www.ieee802.org/1/files/public/docs2005/new-bergamasco-bcn-september-interim-rev-final-0905.ppt
- **Note – behavior is reactive, not intended for managed flows**
- **Relies on flow lifetime long enough to allow reaction**
  - Otherwise, has no effect (equivalent to no management)
- **Generally, goal of CM to keep queue length short**
  - Minimize latency, minimize (or eliminate) packet loss
- **Buffer size requirement (for low or no packet loss)**
  - Dependant on bandwidth.delay product
  - i.e. amount of data to be absorbed before CM starts to work

Sample system (for description)

1 congestion point, 1 reaction point considered



1. Congestion builds…

Multiple data sources start sending data through a congestion point (sources & destinations vary)

- **Queue in congestion point starts to grow**

  Eventually queue depth crosses equilibrium point

- **Sample incoming traffic (Pm sample probability)**
- **Generate BCN (Qoffset, Qdelta) packet**

  Contents: DA = SA of sampled frame; SA = address of CP; Q-TAG (high priority); Ethertype = BCN; Congestion Point ID (CPID); Qoffset = offset of queue depth from equilibrium at time of sample; Qdelta = change in queue depth since last sample; timestamp (for optimization); first N bytes of sampled frame – to allow reaction point to see higher layers

- **BCN packets sent back to source (expected Pm ~= 1/100)**

  (v. low overhead)

2. First response

The BCN traverses the network to the source of the data stream

- The edge device receives the BCN and installs a rate limiter

  Granularity is implementation dependant – assume DA/PRI

- Packets that match DA/PRI enter queue; others bypass

  All packets from queue are tagged with rate limiter id

- Queue drain rate goes down with each BCN received…

  rate' = rate * (1 – Gd * |Fb|)

  Gd = decrease gain multiplier

  Fb (feedback) = Qoff – W * Qdelta (W = derivative weight)

- Multiplicative decrease => rapid decrease of b/w

  Minimizes chance of queue overflow, even if many streams collide

3. Settling

Tagged frames from the source elicit responses from the congestion point

- **Packets are sampled with same probability**
  - **All sampled tagged packets generate a response**
- **As the source rate falls, the congestion point queue shrinks**
  - **Offset and delta counteract & rate settles to equilibrium**
- **Congestion point removes all RLT tags**
- **If the queue drops below the threshold, or is dropping rapidly**
  - **Rate increase: rate' = rate + Gi * Fb * Ru**
  - **Gi = increase gain multiplier**
  - **Ru = rate unit**
- **Additive increase => slower recovery of b/w**
  - **Avoids unfavorable oscillatory behavior**

4. Equilibrium

Depending on gain & weight, the stream will reach equilibrium sooner or later

- Equilibrium really means oscillation around equilibrium point
    - Queue depth rises & falls periodically
- Amplitude governed by gain, weight and RTT
    - Faster convergence related to larger oscillation
    - Larger oscillations also result in more rapid “fairness”
    - … but larger oscillations mean higher probability of packet loss
    - … or wasted bandwidth (queue goes empty)
- Control parameters may be optimized for specific network
    - Either by observing oscillation behavior
    - Or by using timestamps explicitly
- Eventually, multiple streams all settle to equal rates
    - Fairness optimization useful for very long flows

5. Recovery

Source must return to full b/w: either flow finishes or congestion dissipates (other flows finish)

- **Other flows subside, positive BCNs allow rate to increase**
  - **When rate reaches maximum, rate limiter is removed**
- **Otherwise, if flow at entry point ends, rate limiter dissipates**
  - **Slow recovery prevents problems with stop-start flows**
- **Restarting flow (with rate limiter still in place)…**
  - **… first frames are RLT tagged, generate positive responses**
  - **Rate limiter dissipates more rapidly**
- **Congestion might return – more BCNs & rate limiter increase**
- **In most cases, congestion point will move elsewhere**
  - **… especially for meshed networks & random flows**

6. Other considerations

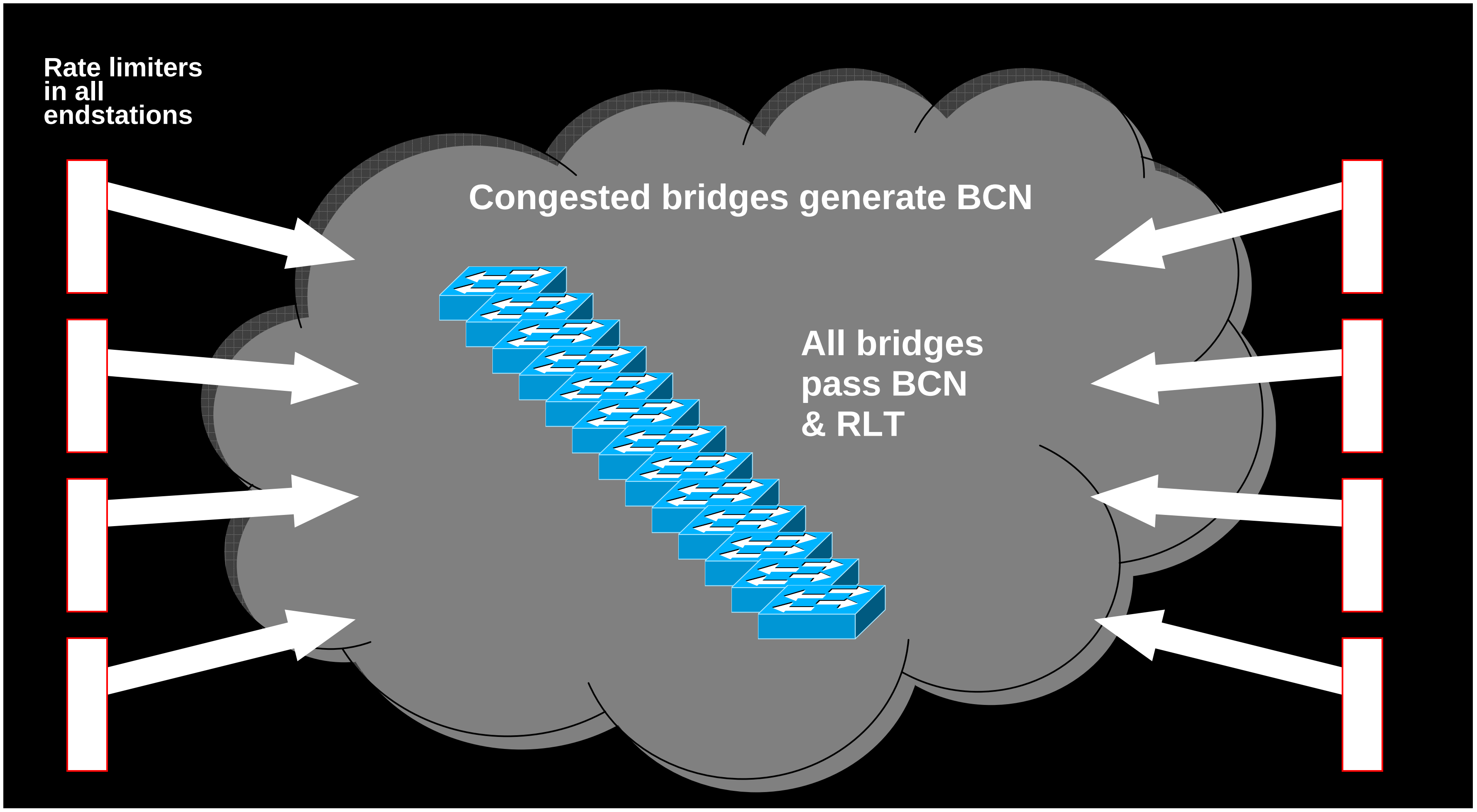
CM reduces network latency due to congestion

- Reducing network latency may improve performance
  - Some applications benefit – others don't!
- Reduction in latency due to reduced queue length
  - Could equate to reduced network device buffer size
  - And/or lower packet loss rate – depending on buffer size
  - For lossless behavior ~= sigma (input b/w) * control loop delay
- Mechanism beneficial if flow life >> network delay
  - e.g. 8 hops @ 2uS << 64kbyte @ 10Gbps
- Shorter flows do not benefit from BCN but fit in buffers
  - Flow response delay will throttle throughput

Deployments

Ideal installation

Compliant endstations & bridges



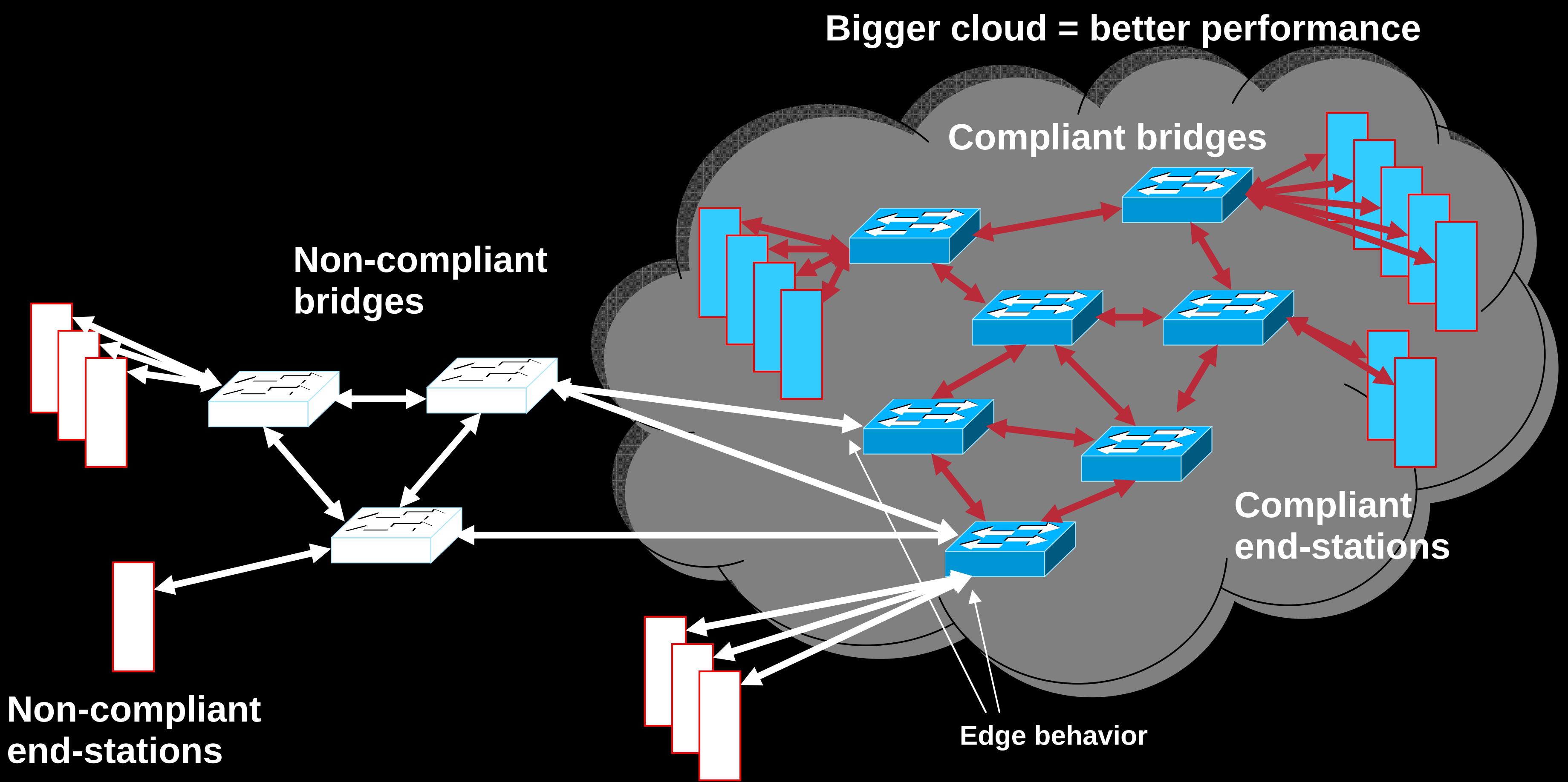
Optimal performance

With ideal network, analysis suggests >90% of maximum theoretical throughput with minimal increase in latency

- **Endstation rate limiter granularity optimized for application**
  - Single threaded or simple device = simple rate limiter
- **Timestamps may be used to optimize system parameters**
  - Balancing maximum performance vs latency or risk of packet loss
  - Further study required to weigh benefit vs simplicity
- **Scaleability supports network sizes >> 1000 nodes**
  - “workable” buffer sizes & near perfect throughput
- **Endstation optimization may ascend OSI stack**
  - Rate limiter backpressure feeds into transport or above…
  - … including application balancing based on congestion

CM cloud

Compliant devices in cloud, edge behavior



CM cloud, mixed old and new systems

Introduction of CM devices in key parts of network offers significant advantages

- CM cloud is formed, only compliant devices allowed inside
  - LLDP or other mechanism to define cloud
- If source, destination & path all use CM then optimal behavior
- At edge of cloud edge devices act as pseudo end stations
  - Rate limiters installed at cloud ingress
  - RLT tags stripped at egress (only occurs in corner cases)
- Rate limiters may require larger buffers or intelligent packet deletion
  - CM cloud edge devices similar complexity to legacy L2+ devices
- Network performance improves as cloud grows…
  - … best "bang for buck" = CM cloud in data core

Q and A

Cisco Systems