OverQoS: Enhancing End-to-End Path Quality using Overlays

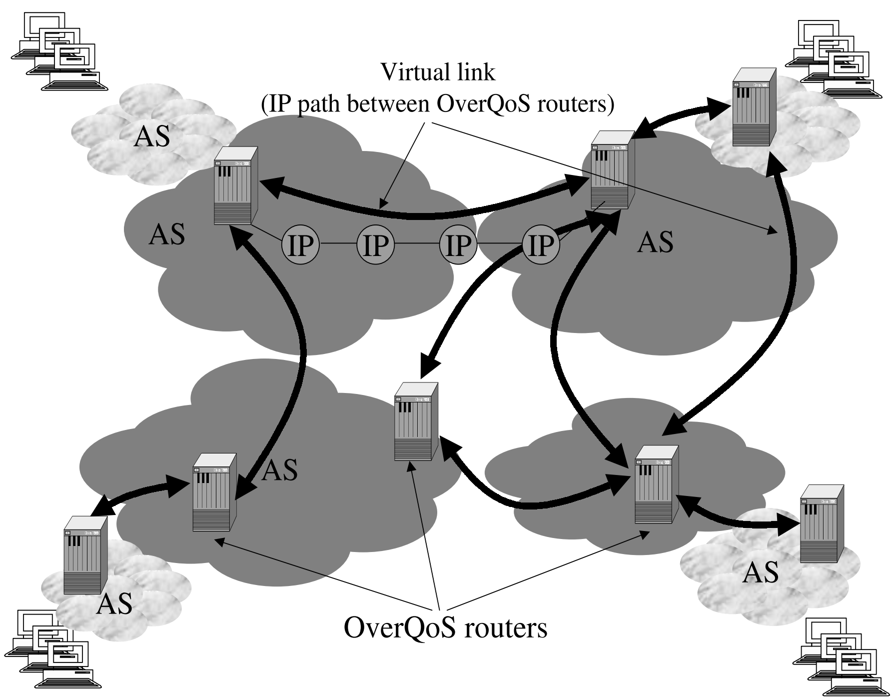
Lakshminarayanan Subramanian

Ion Stoica

Hari Balakrishnan

Randy H.Katz

Can Overlays better Best-Effort?

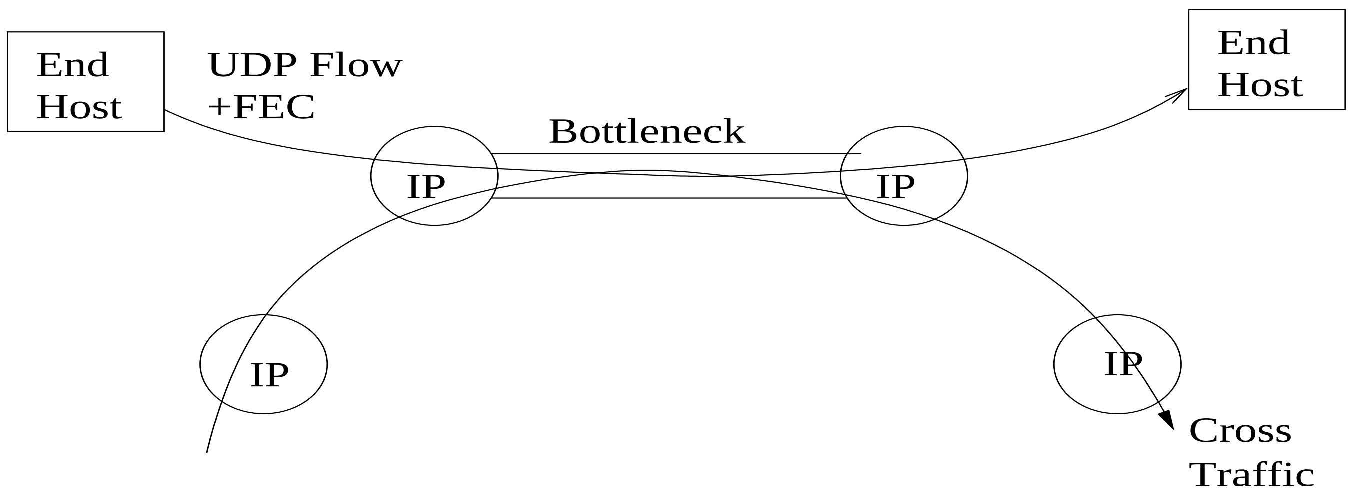


- **Given no QoS support in the routers, can we enhance the quality of paths end-to-end? If so, by how much?**

What Overlays provide?

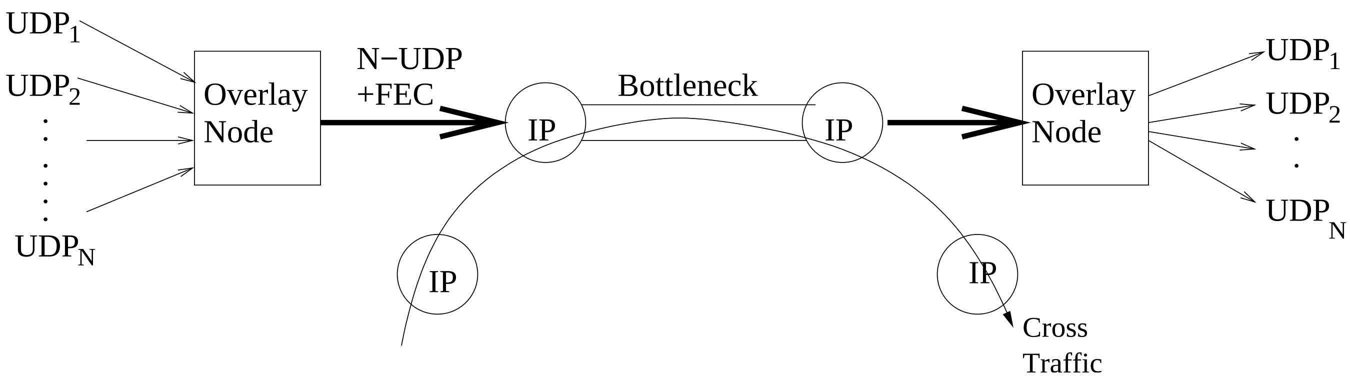
- **Overlays provide access to points within the network.**
  - **Exert control on the individual flows traversing an overlay node.**
  - **Can deploy any QoS scheduling discipline at these nodes.**
- **Overlay solutions are incrementally deployable and also fast to deploy.**
- **An overlay approach is supplementary/complementary to existing architectures.**

Scenario: One UDP flow



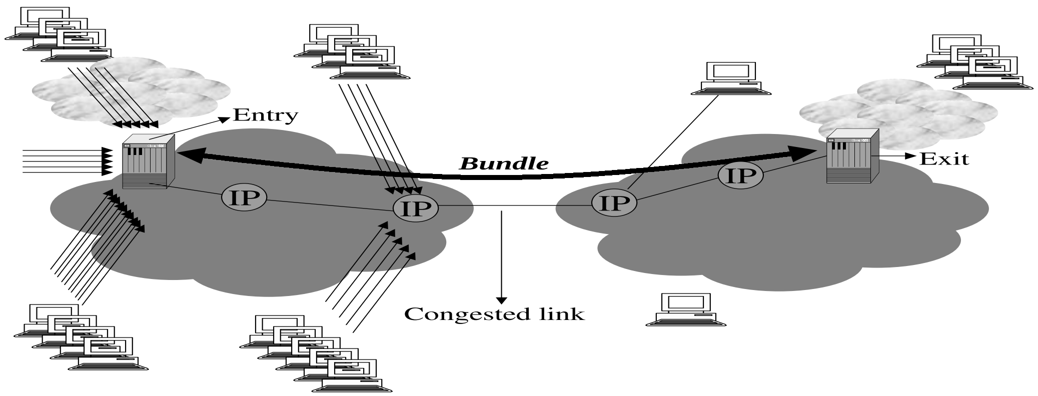
- ***Cannot control cross traffic and cannot avoid losses.***
- **Losses may be unpredictable.**
- **Proposal: End-host adds FEC to protect the UDP traffic from losses due to cross traffic.**

Applying FEC in the Aggregate



- Apply FEC to an aggregate of flows than on a per-flow basis.
- Benefits:
  - $FEC_{Aggregate} < FEC_{per-flow}$
  - Time to recovery from packet loss reduces.
- Routing using an Overlay automatically enables aggregation.

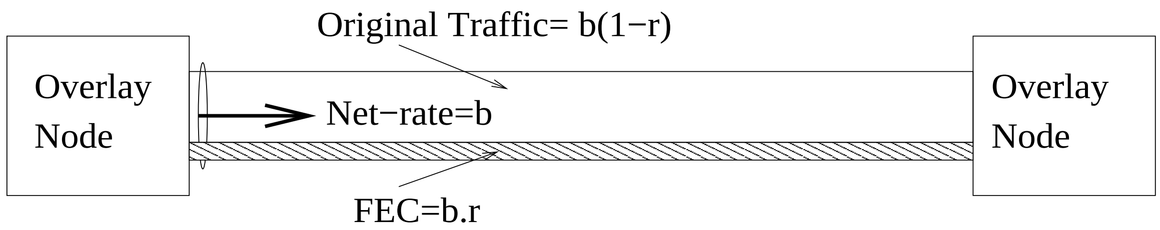
Aggregation provides Control



- The node can distribute the available bandwidth among the flows in a bundle.

- If the arrival rate is more than available bandwidth, the overlay node can redistribute the losses *unequally* between the flows.

Loss Control



- ***Since losses are unavoidable, lets control them using FEC.***
- **Goal of FEC: Reduce the net-loss rate observed by the original traffic from p to q.**
- **Note: q is a small constant independent of p.**
- **Why not $q = 0$?**
  - **Variation of p cannot be exactly predicted and we cannot achieve zero-loss abstraction.**

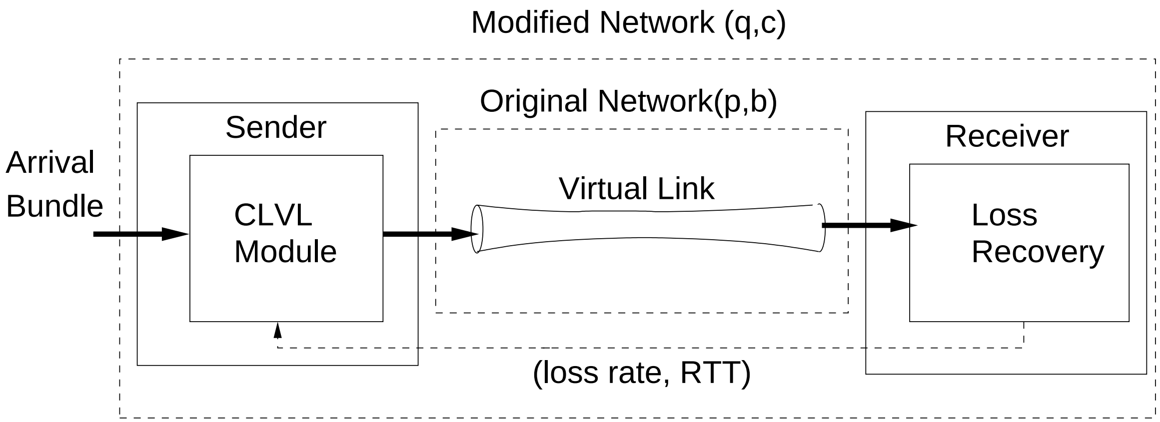
Being fair to cross traffic?

- **The overlay traffic should NOT blast out a lot of redundancy packets and kill cross traffic.**
  - **Unfair to cross traffic.**
  - **Causes instability in the system when multiple such overlays exist.**
- **Define an N-TCP pipe to have the throughput of N-equivalent TCPs:**

$$N \times \frac{K \times S}{RTT \times \sqrt{p}}$$

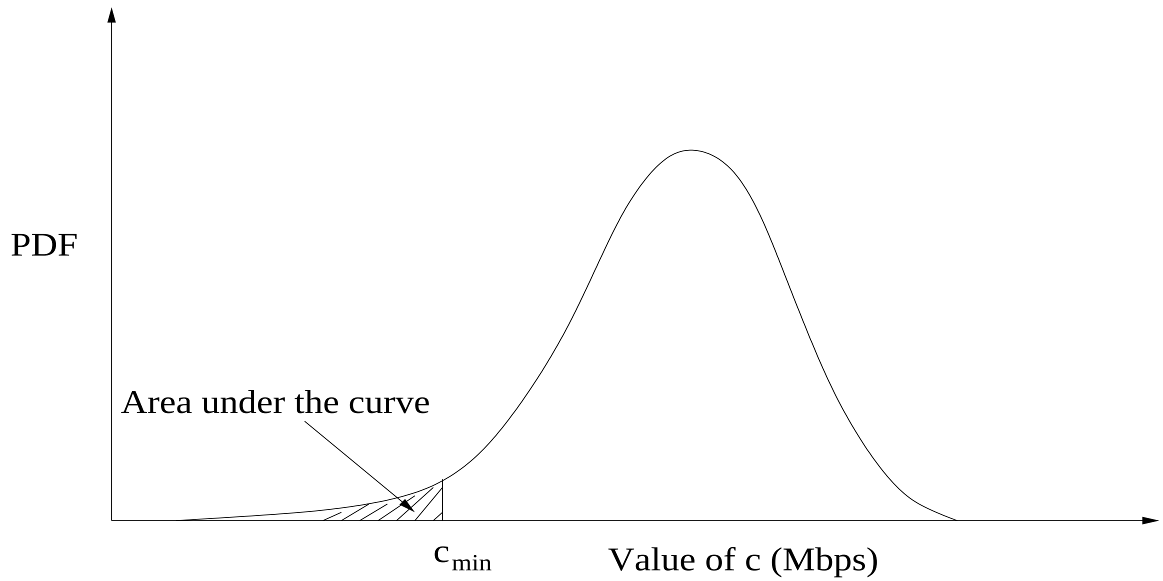
- **Calculate b using the $N-$TCP equation for a fixed and pre-specified value of N.**

Controlled Loss Virtual Link(CLVL)



- **Controlling $b \rightarrow$ controlling FEC or input-traffic**
  - **Achieving target loss rate q is more important**
  - **We control the input-traffic to be $\leq c = b(1 - r)$.**
- **If arrival bundle is less than c Mbps, the loss-rate is bounded by q.**
- ***Challenge: How to ensure input-traffic is less than c Mbps?***

Searching for a stable value: c_{min}



- **Divide time into $T_0 = O(RTT)$ slots and compute the distribution of c.**
- **Find a c_{min} such that $P(c < c_{min})$ is small.**
- **Admission Control: Admit flows with net bandwidth requirement $\leq c_{min}$.**
- **c_{min} is stable as long as distribution is stationary.**

Admitted flows and QoS

- **The flows admitted can be given:**
  - **Statistical loss guarantees**
  - **Statistical bandwidth guarantees (over periods where the distribution of c is stationary)**
  - **No delay guarantees (depends on how CLVLs are built)**
- **Admission control is useful only for the period over which c_{min} is stable.**
- **Is this good for streaming media?**

What about $c - c_{min}$?

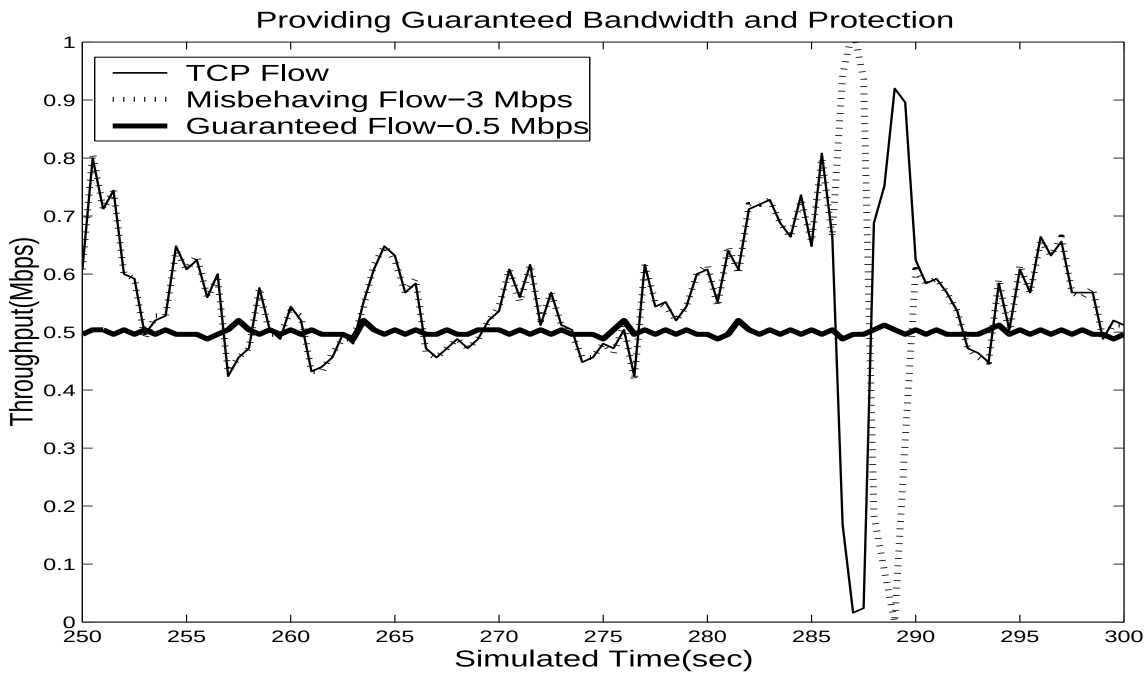
- In practice, c_{min} is only a fraction of c (say 30% of the average value of c)
- Allocate this bandwidth to ***best-effort*** overlay traffic.
- Benefits for this class of flows:
  - Add FEC on the fly to protect from losses due to cross traffic.
  - Protect one overlay flow from another by employing Fair-Queuing at the overlay node.
  - Overlay node can also do ***DiffServ*** among these flows.
- We need these flows in the aggregate:

$$FEC_{aggregate} < FEC_{per-flow}$$

What enhancements do we provide?

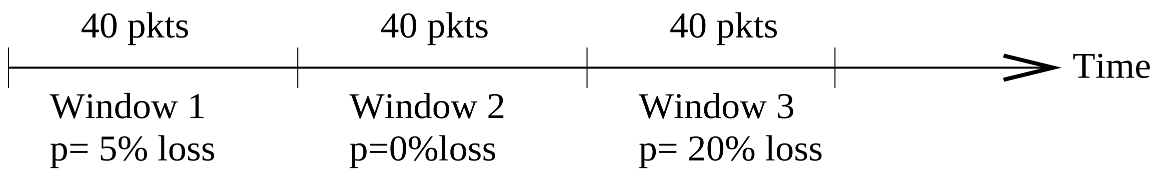
- **Introduce two classes of *overlay* flows:**
  - **QoS flows**
  - **Best-effort flows**
- **QoS flows get:**
  - **Statistical loss and b/w guarantees over stationary time periods.**
- **Best-effort flows get protection from other flows and experience reduced loss.**

Variety of services using CLVLs



- **Setting:** 3 **flows in a bundle,** 50 **background TCP flows.**
- **Providing bandwidth guarantees to one flow and protection to TCP flow.**

Building CLVLs using FEC



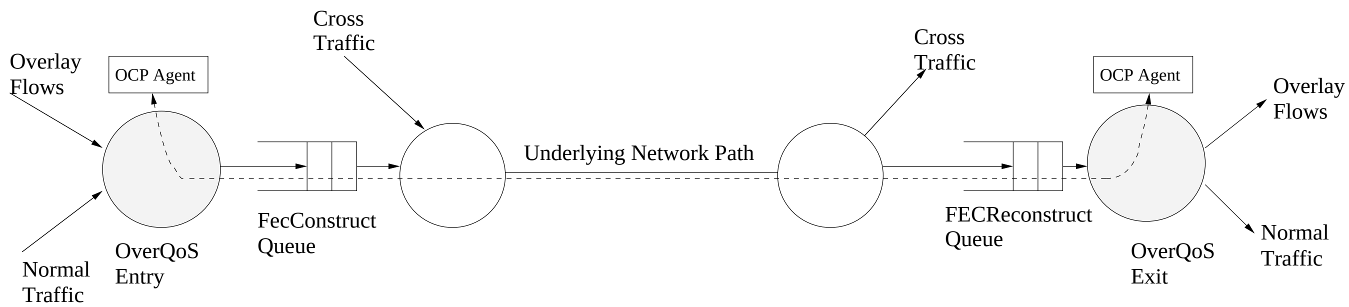
- **Consider a Reed Solomon code (n, k) consisting of k original packets and $n-k$ redundant packets.**
  - **Let p be fraction of packets lost in this window of n packets.**
  - **If $p > \frac{n-k}{n}$, then the effective loss rate of window is p.**
- **Simplistic Assumption: Let p be drawn from a distribution $f(p)$.**
- **Determine minimum *redundancy factor* $r = \frac{n-k}{n}$ such that the expected loss rate is q.**

$$\int_r^1 p f(p) dp = q$$

FEC-based CLVL algorithm

- **Basic Algorithm:**
  - **Given a target loss rate q, consider $2/q$ window loss samples.**
  - **Compute $\widehat{f()}$ as an approximation to the original distribution $f()$.**
  - **Compute $\hat{r}$ using $\widehat{f()}$ for a given value q.**
- **Addressing the problem of correlated loss samples:**
  - **Take a random subset of the loss samples and compute $\widehat{f}$ over the subset of samples.**
  - **Repeat for different subsets of loss samples.**

Simulation Setup

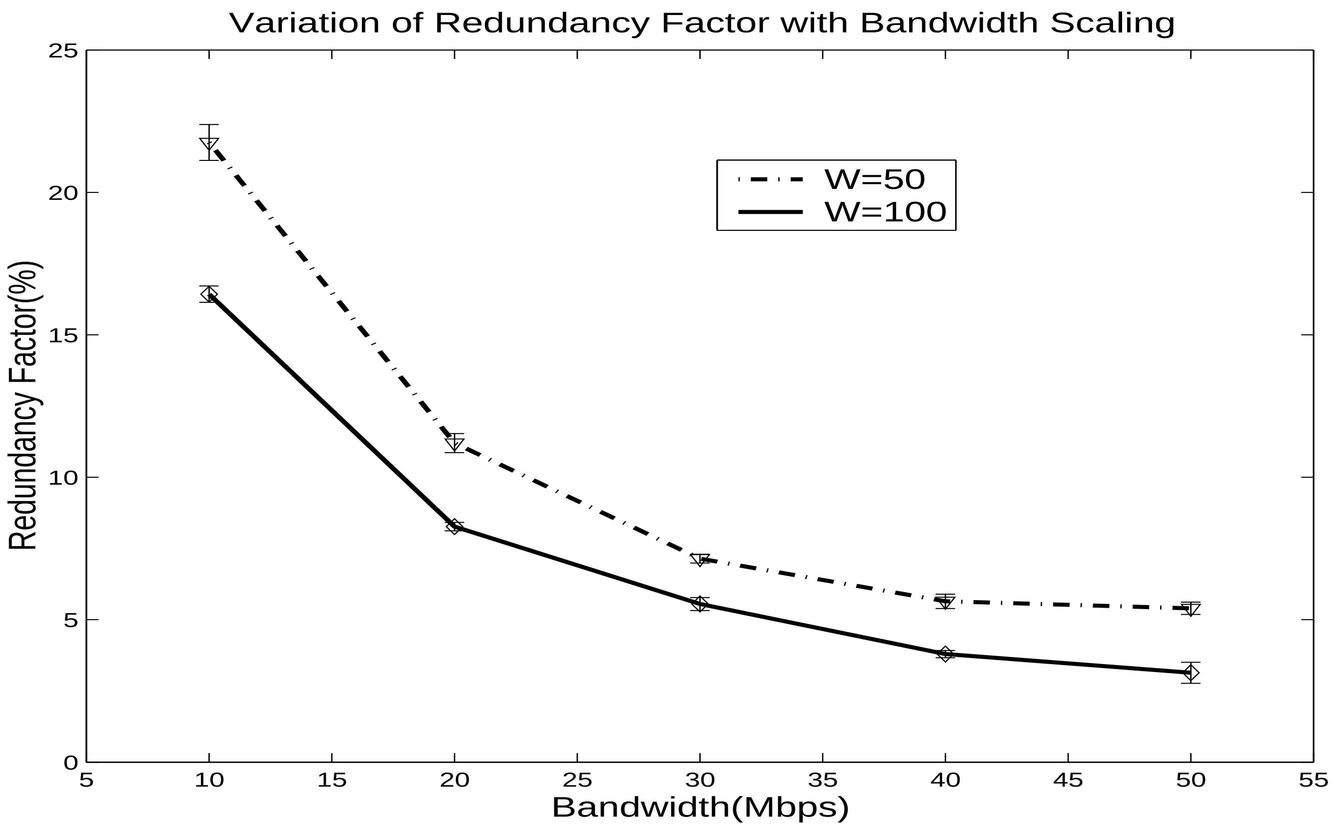


- **Implemented in *ns-2*.**
- **Test our algorithm against variety of background traffic models**
  - **Long lived TCPs**
  - **Web Traffic**
  - **Self Similar Traffic**
  - **Long lived TCPs + Impulses**
  - **Access Router Traces**

Loss Characteristics: Self Similar background

- **Achieves target loss rate=** 0.1% **even when background=**9 **Mbps self-similar and** $b = 2$ **Mbps.**

Scaling Characteristics: Proportional



- **Proportional scaling phenomenally decreases the overhead.**
- **Background traffic is self-similar.**

Loss/Overhead: Other Scenarios

- **Loss-rate is also met in following scenarios:**
  - **Access router traces**
  - **Web Traffic and Self-Similar Traffic**
  - **Bottleneck gateway is FIFO.**
  - **Across SACK, Reno, New reno variants of TCP.**
  - **When virtual link includes multiple bottlenecks.**
  - **Multiple CLVLs compete.**
- **Except in very heavy tailed distributions, the overhead is very close to calculated optimum.**
- **Overhead decreases when q increases.**

Take-aways

- **Overlays can be used to provide two types of *quality enhancements* to end-end paths:**
  - **Statistical loss and bandwidth guarantees to some flows.**
  - **Best-effort traffic with reduced loss and increased protection.**
- **Aggregation is the key.**
  - **Aggregation provides control in distributing b/w c among the flows.**
  - $FEC_{aggregate} < FEC_{per-flow}$
- **We rely on stationarity of the loss distribution over short time-periods to provide these services.**