Aggregate Congestion Control for Distributed Multimedia Applications

David E. Ott, Travis Sparks, and Ketan Mayer-Patel
Department of Computer Science
University of North Carolina at Chapel Hill
{ott,sparkst,kmp}@cs.unc.edu
IEEE INFOCOM 2004

Outline

- INTRODUCTION
- COORDINATION PROTOCOL (CP)
- SINGLE FLOWSHARES
- MULTIPLE FLOWSHARES
- IMPLEMENTATION AND EVALUATION
- SUMMARY AND FUTURE WORK

INTRODUCTION

- A class of distributed multimedia applications that we call *Cluster-to-Cluster (C-to-C) applications*.

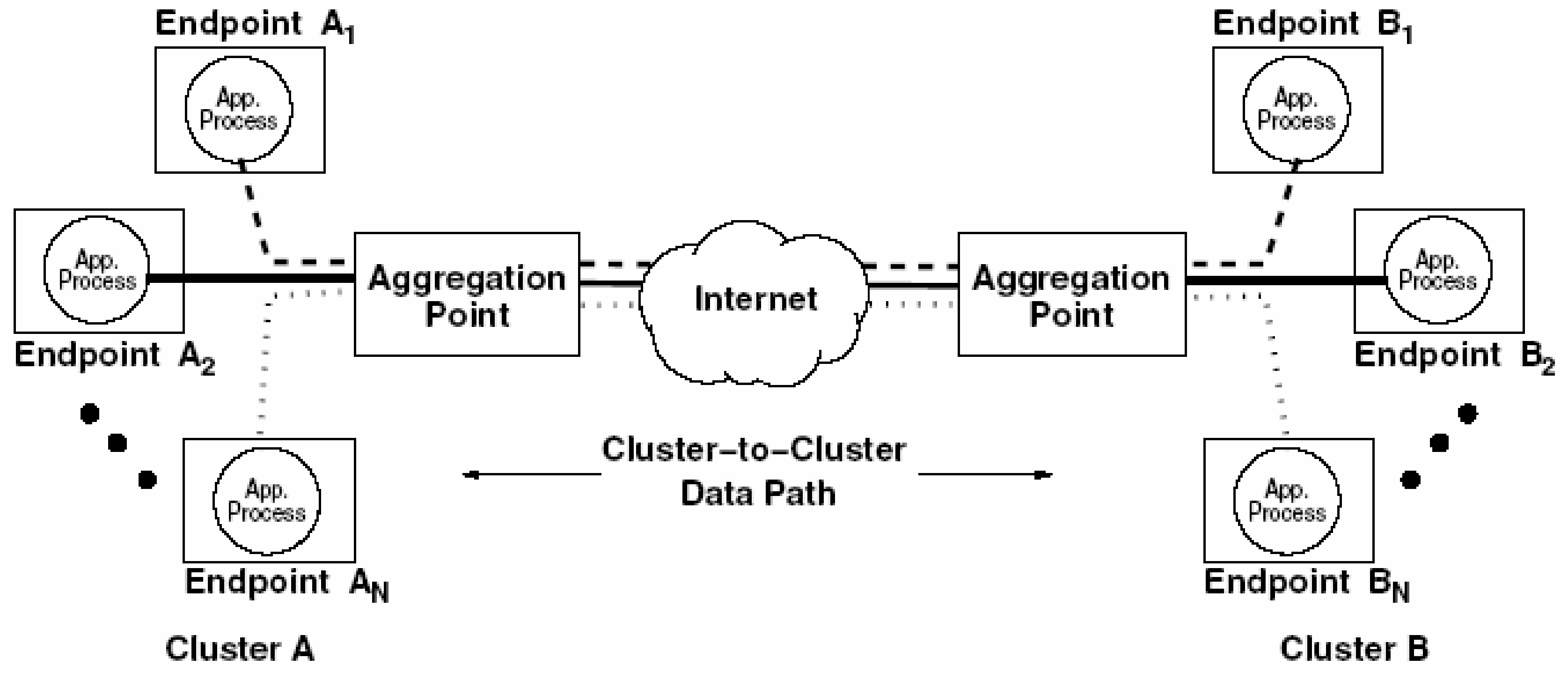

Fig. 1. C-to-C application model.

INTRODUCTION

- An important issue is *congestion control*.
  - individual flows use a variety of transport-level protocols, including those without congestion control.
  - it is essential that *aggregate application traffic* is congestion responsive

INTRODUCTION

- Applying congestion control to aggregate C-to-C application traffic.
- Leveraging existing single-flow congestion control schemes for C-to-C aggregate flows such that：
  - *Cluster endpoints are informed of bandwidth available.*
  - *Endpoints may respond to this information.*
  - *End-to-end semantics are preserved for each individual flow.*
  - *Aggregate application traffic is congestion responsive.*

INTRODUCTION

- An aggregate congestion control scheme should support *multiple flowshares*.
- A C-to-C application that involves multiple flows should receive multiple flowshares.
- An application with *m* flows may receive the *equivalent of m flowshares*.
- For example, some application flows may take more than a single flowshare,while others take less.

INTRODUCTION

- The main contributions of this paper are:
  - *Coordination Protocol (CP)*
  - *TCP Friendly Rate Control (TFRC)*
  - *Bandwidth filtered loss detection (BFLD)*

COORDINATION PROTOCOL (CP)

- CP is implemented between the network layer (IP) and the transport layer (TCP, UDP, etc.).

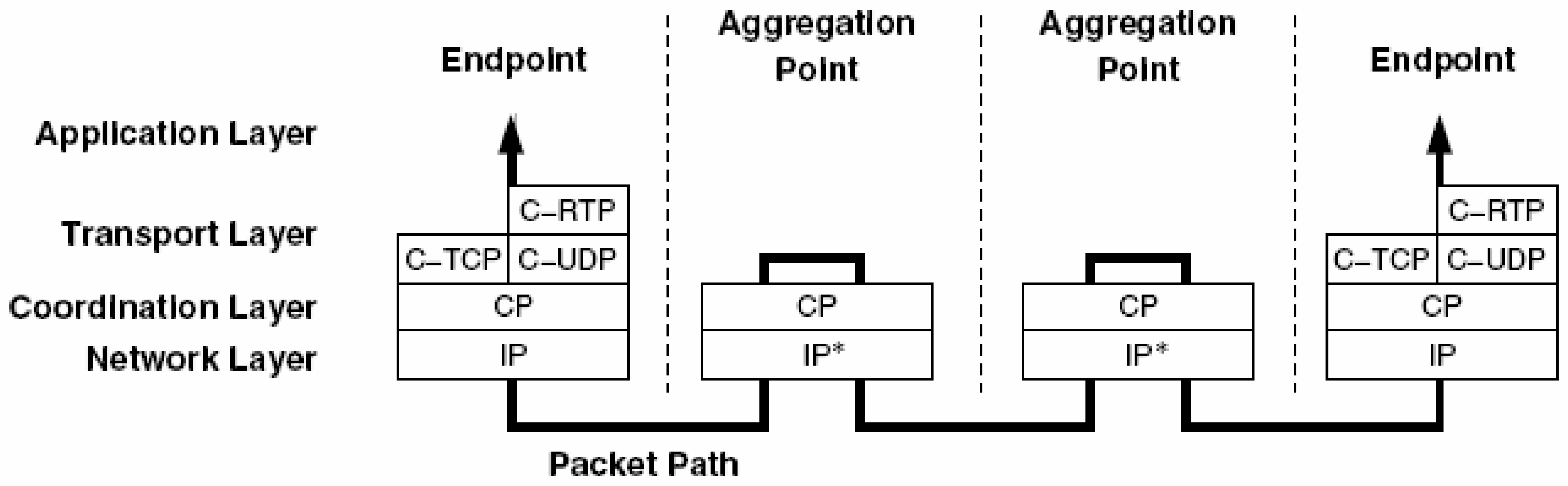

Fig. 2. CP network architecture.

COORDINATION PROTOCOL (CP)

- Using the CP header :
  - a cluster AP identifies C-to-C application packets and
  - Attaches network probe information to each.

- An AP uses aggregate measurements of RTT and loss to drive a rate-based congestion control algorithm (e.g., TFRC or RAP).

COORDINATION PROTOCOL (CP)

- When C-to-C endpoints receive this estimate, they respond by modifying their sending rate.

- The benefits of this approach include:
  - A fast forwarding path
  - Aggregate bandwidth availability
  - Complete application control over the manner in which an aggregate congestion response is realized.
  - Support for multiple flowshares.

COORDINATION PROTOCOL (CP)

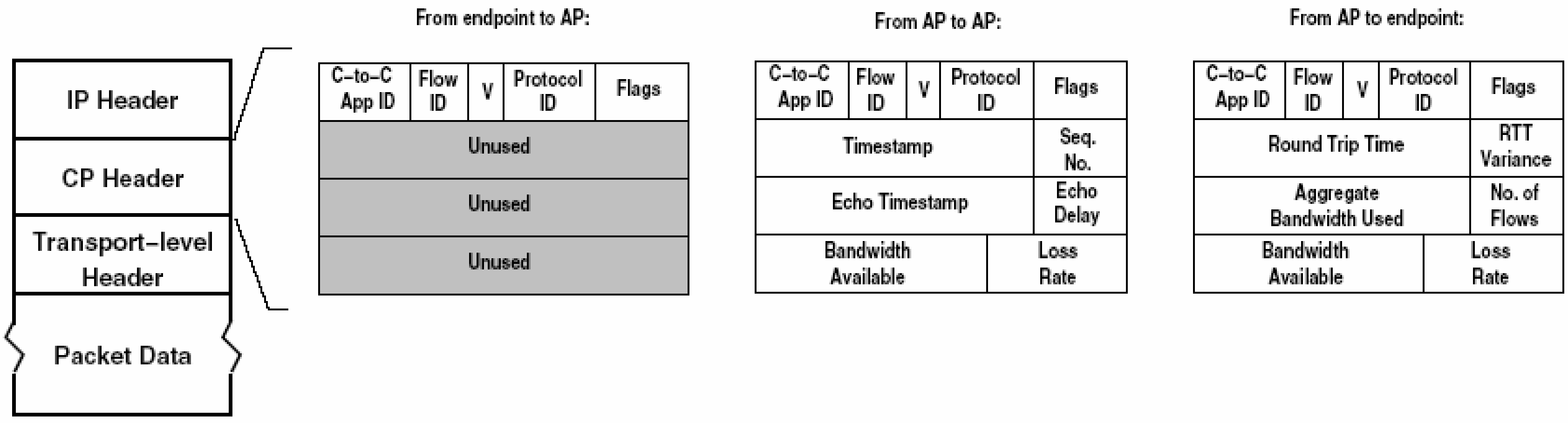

Fig. 3. CP packet structure.

- The basic operation of CP is as follows:
  - **As packets originate from source endpoints**
  - **As packets arrive at the local AP**
  - **As packets arrive at the remote AP**
  - **As packets arrive at the destination endpoint**

COORDINATION PROTOCOL (CP)

- The APs use fields in the CP header to measure RTT and detect loss：
  - To measure RTT：
    - Inserts a timestamp which is echoed along with the delay since that timestamp was received.
    - *RTT = current time - timestamp echo -echo delay*.
  - To detect loss：
    - inserts a monotonically increasing sequence number.

COORDINATION PROTOCOL (CP)

- TCP (C-TCP) and UDP (C-UDP) implemented using a modified socket API.

- UDP(C-UDP)：provide an interface to set ：
  - the C-to-C application id and flow id,
  - and get the latest estimated RTT, aggregate loss rate, and estimated available bandwidth.

- TCP (C-TCP)：provides the same end-to-end semantics as TCP (i.e., a reliable byte stream), but relies on the underlying CP protocol to detect congestion and suggest an appropriate sending rate.

SINGLE FLOWSHARES

- We refer to our *ns2* implementation of the TFRC congestion control algorithm in CP as *CP-TFRC*.

$$X = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}(3\sqrt{\frac{3bp}{8}})p(1+32p^2)}$$

- transmission rate *X* (bytes/sec)：
  - *s* is the packet size (bytes),
  - *R* is the round trip time (sec),
  - *p* is the loss event rate,
  - t_{RTO} is the TCP retransmission timeout (sec)
  - *b* is the number of packets acknowledged by a single TCP acknowledgement.

SINGLE FLOWSHARES

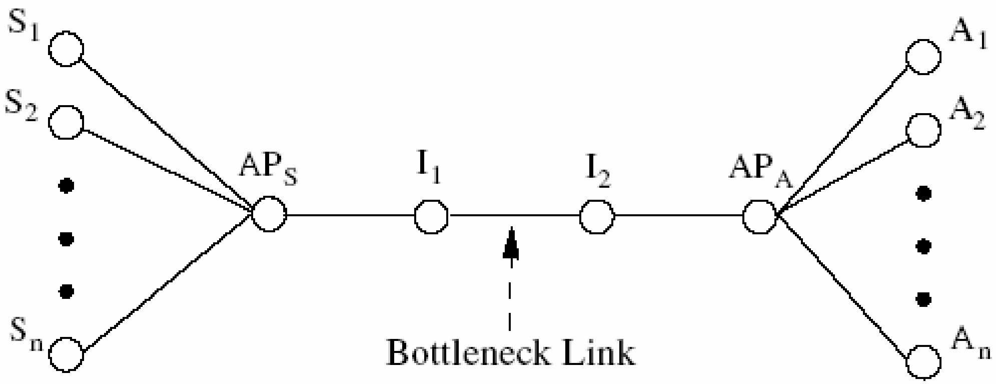

Fig. 4. Simulation testbed in ns2.

Parameter	Value
Packet size	1 K
ACK size	40 B
Bottleneck delay	50 ms
Bottleneck bandwidth	15 Mb/sec
Bottleneck queue length	300
Bottleneck queue type	RED
Simulation duration	180 sec

TABLE I

CONFIGURATION PARAMETERS.

SINGLE FLOWSHARES

- Compare aggregate CP-TFRC traffic using a single flowshare with competing TFRC flows.

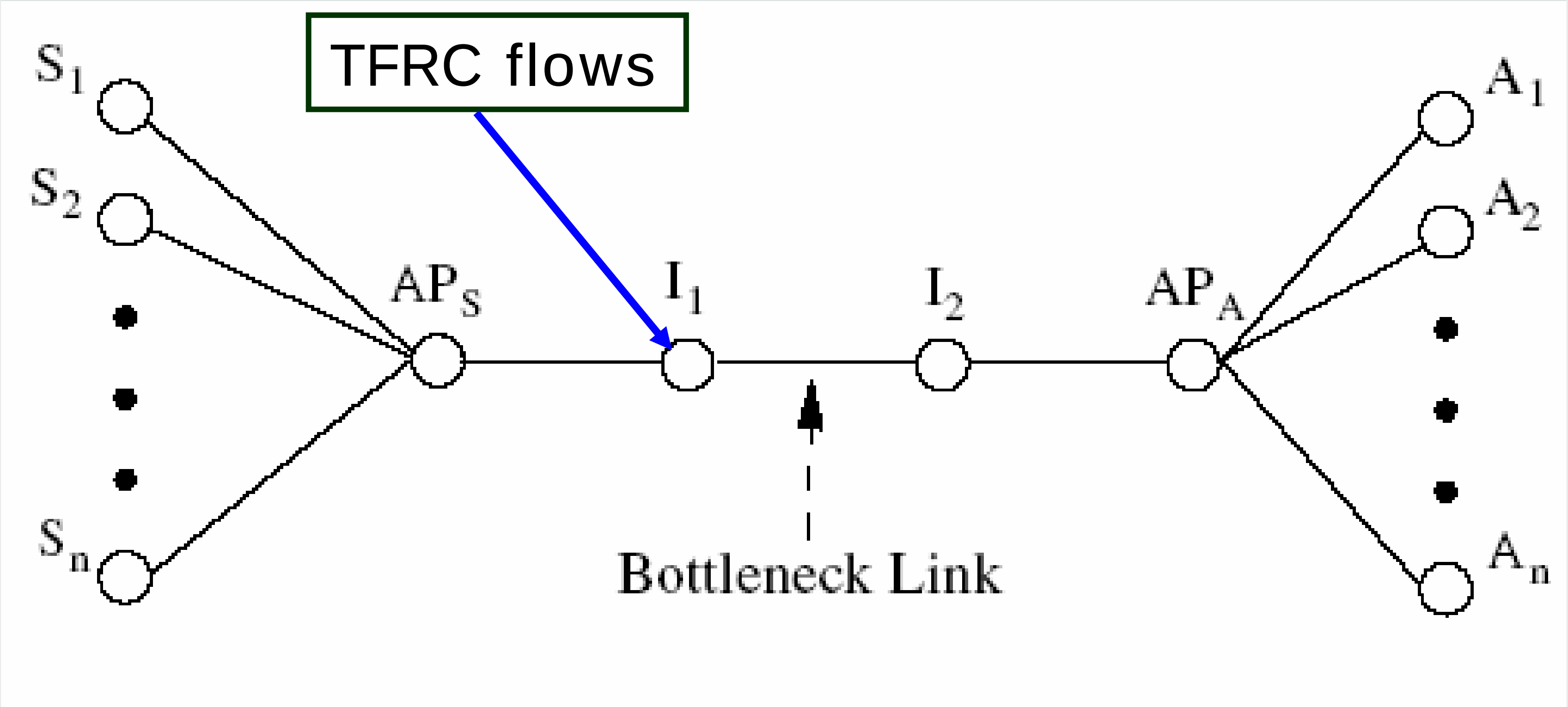

Fig. 4. Simulation testbed in ns2.

SINGLE FLOWSHARES

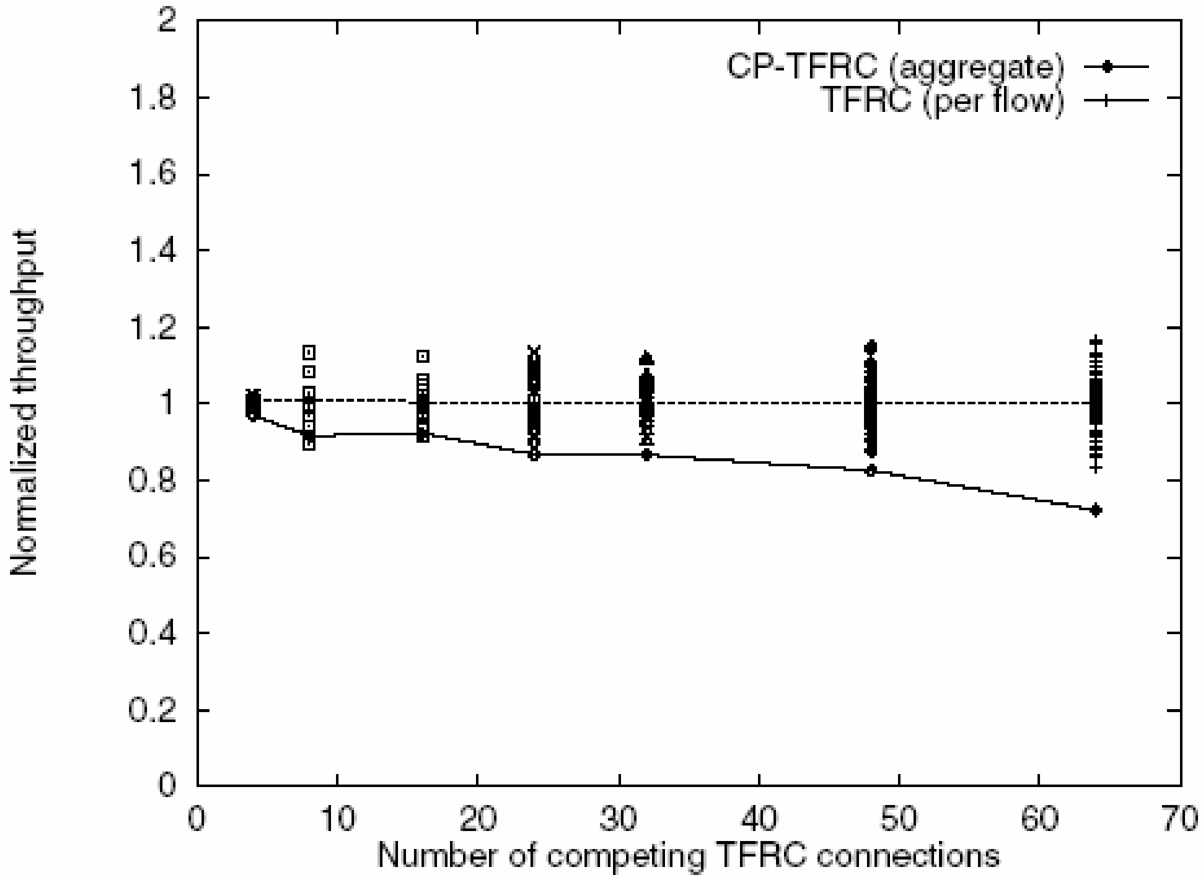

Fig. 5. TFRC versus CP-TFRC normalized throughput as the number of competing TFRC flows is varied.

SINGLE FLOWSHARES

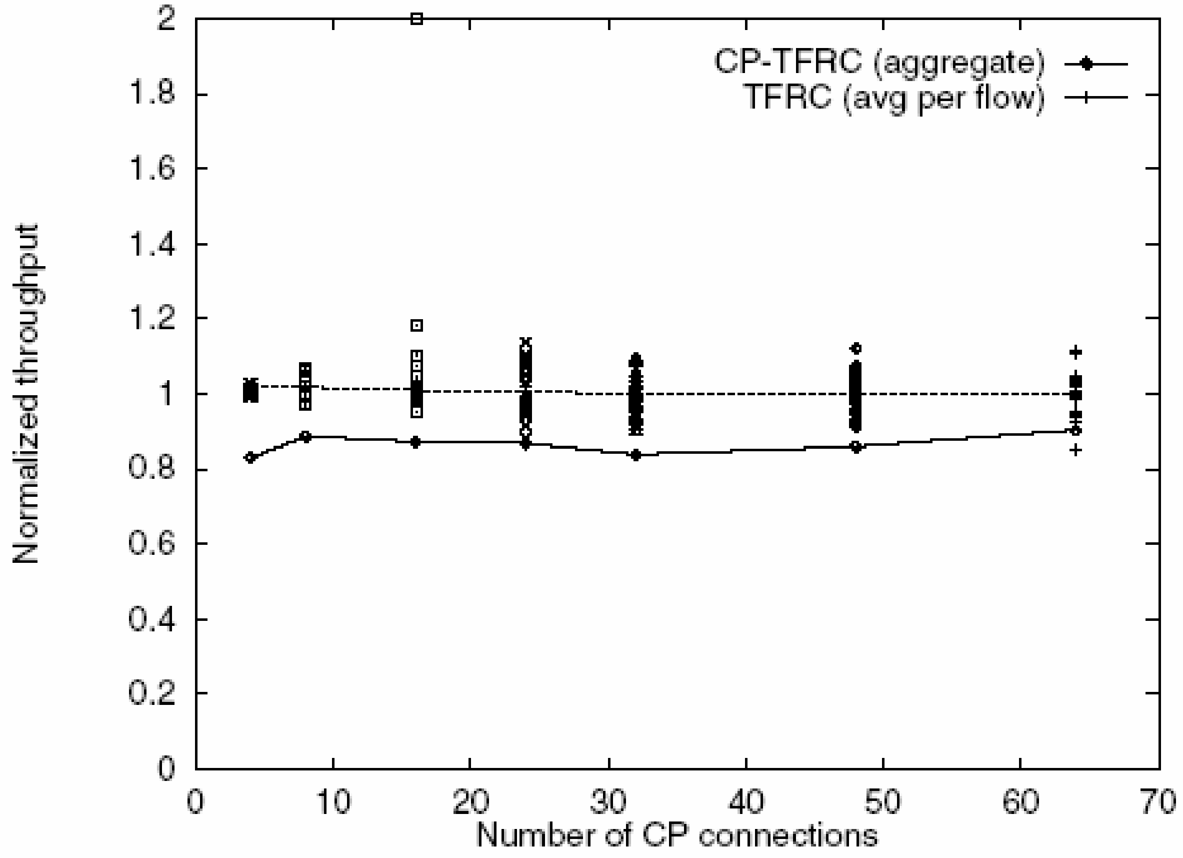

Fig. 6. TFRC versus CP-TFRC normalized throughput as the number of flows in the C-to-C aggregate is varied.

MULTIPLE FLOWSHARES

- That single-flow congestion control algorithms break when a sender fails to limit their sending rate to the rate calculated by the algorithm.
- After discussing the problem, we present a new technique, *bandwidth filtered loss detection (BFLD)* in enabling multiple flowshares.

MULTIPLE FLOWSHARES

- Allow C-to-C applications to *m* flowshares in aggregate traffic,where *m* is equal to the number of flows in the application.

MULTIPLE FLOWSHARES

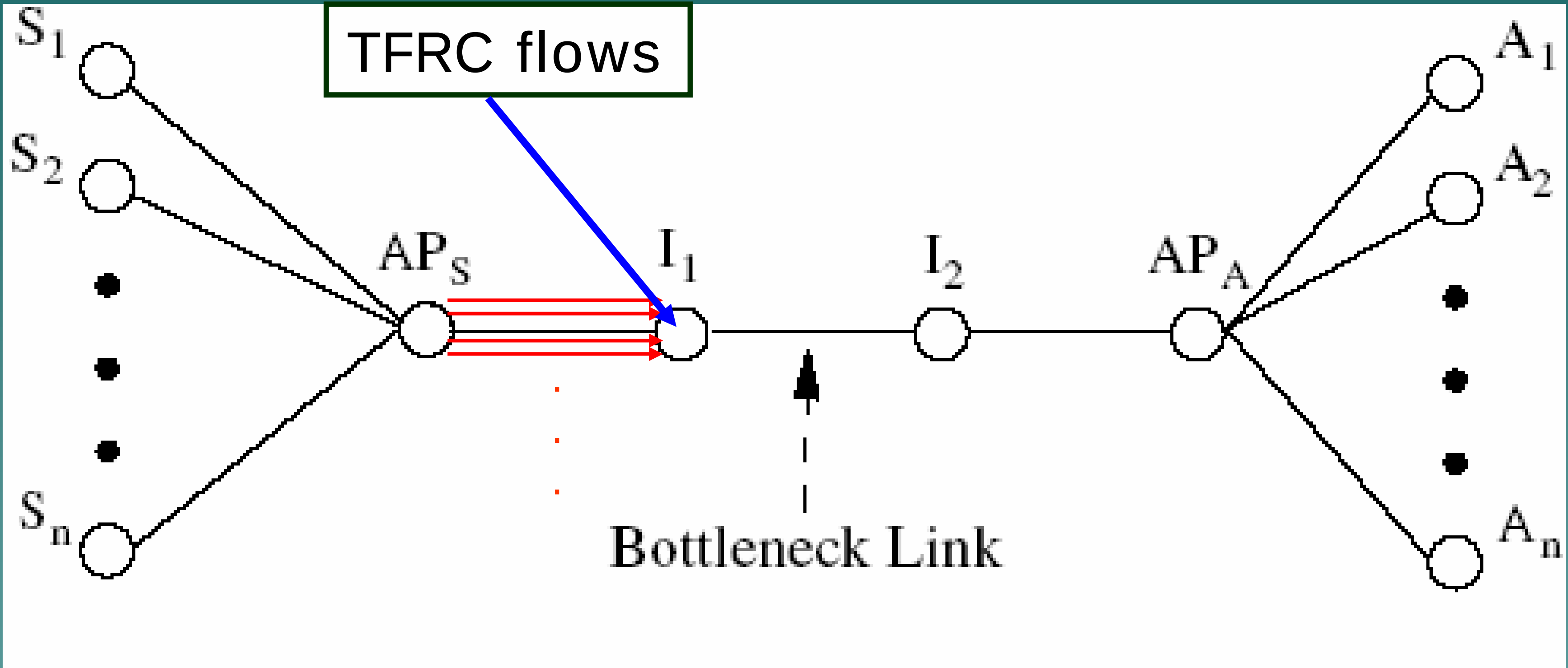

Fig. 4. Simulation testbed in ns2.

MULTIPLE FLOWSHARES

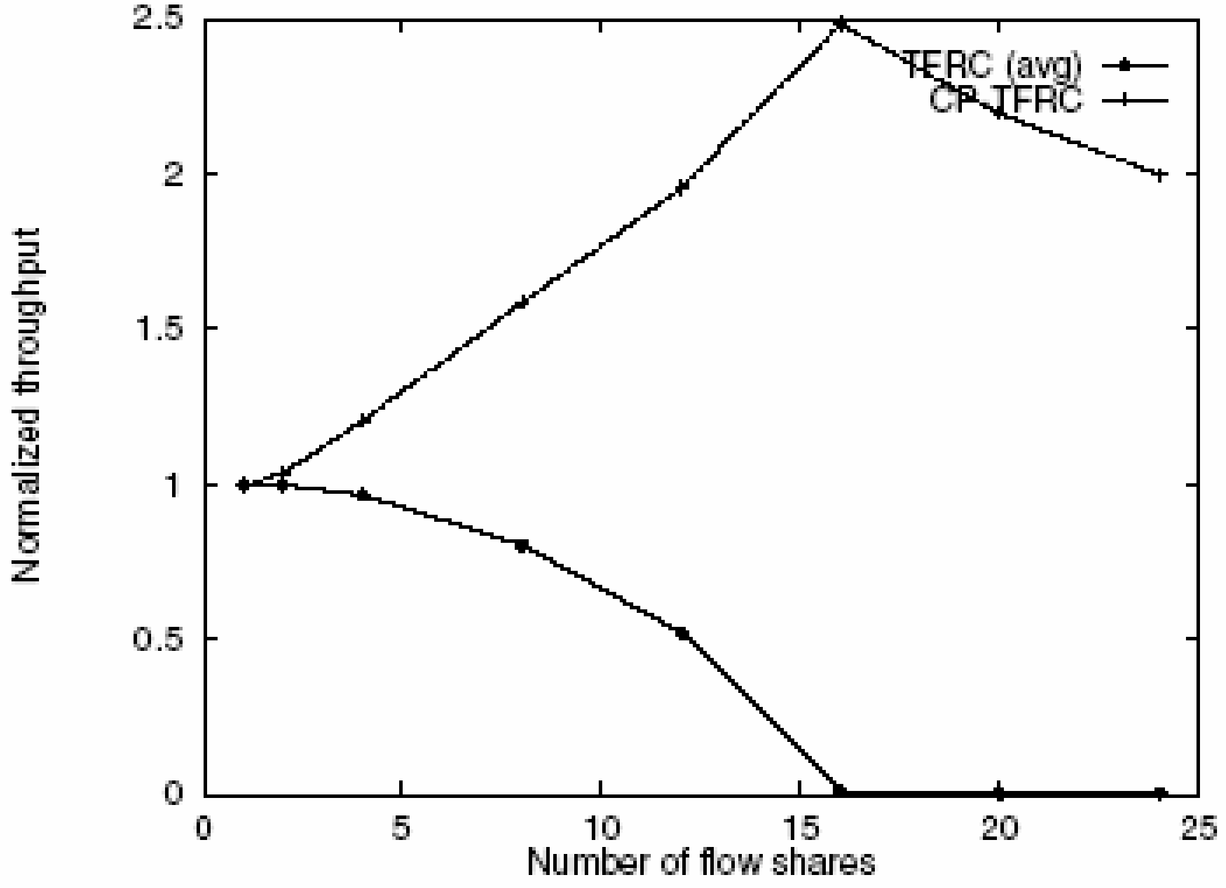

Fig. 7. Throughput for multiple flowshares (naive approach).

MULTIPLE FLOWSHARES

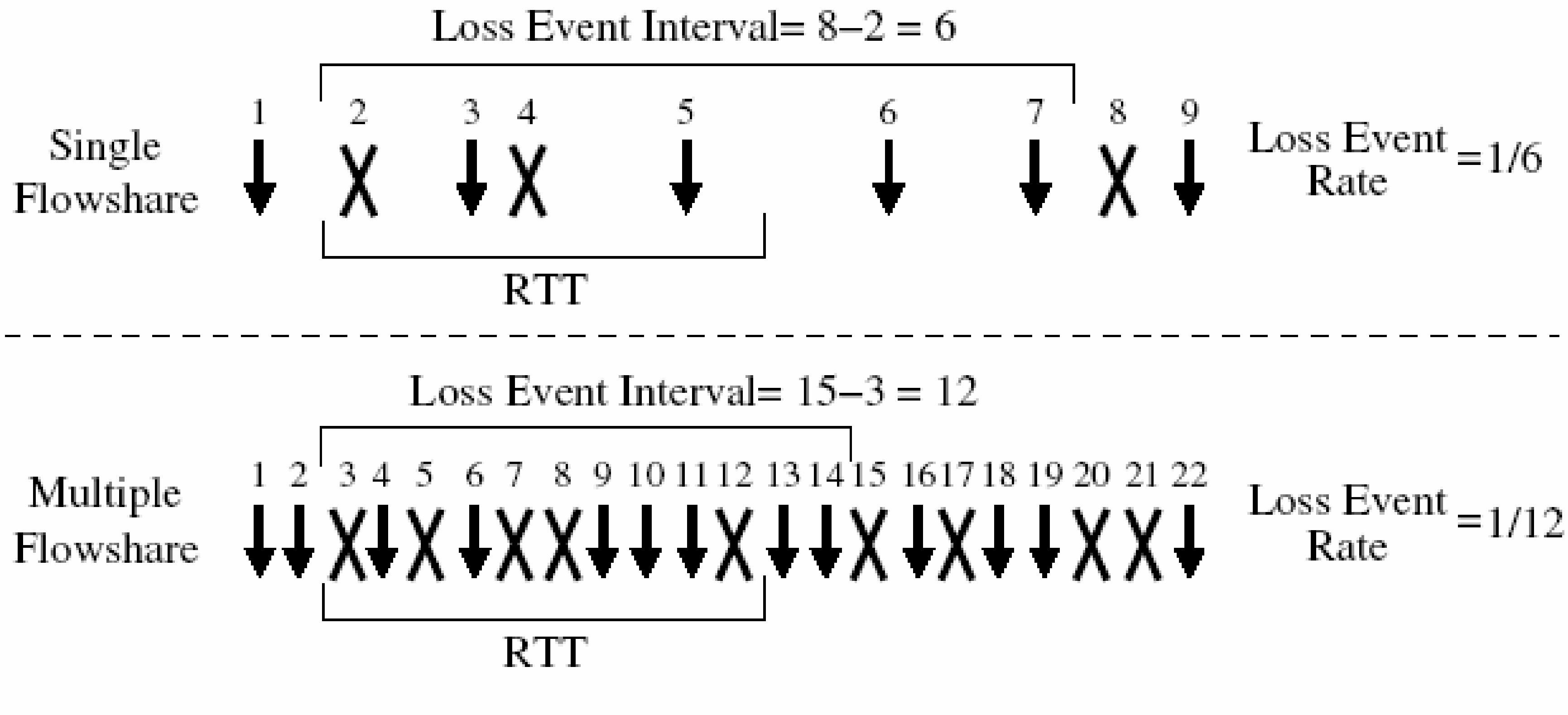

Fig. 8. Loss event rate calculation for TFRC.

MULTIPLE FLOWSHARES

- Our solution to the problem of loss detection in a multiple flowshare context is called *bandwidth filtered loss detection(BFLD)*.
- A *sampling fraction F* is calculated as：
  - $F = B_{avail}/B_{arriv}$. If $B_{avail} > B_{arriv}$, then F is set to 1.0.
  - *available bandwidth*(B_{avail}) calculated by the congestion control algorithm employed at the AP.
  - *arrival bandwidth*(B_{arriv}) is an estimate of the bandwidth currently being generated by the C-to-C application.

MULTIPLE FLOWSHARES

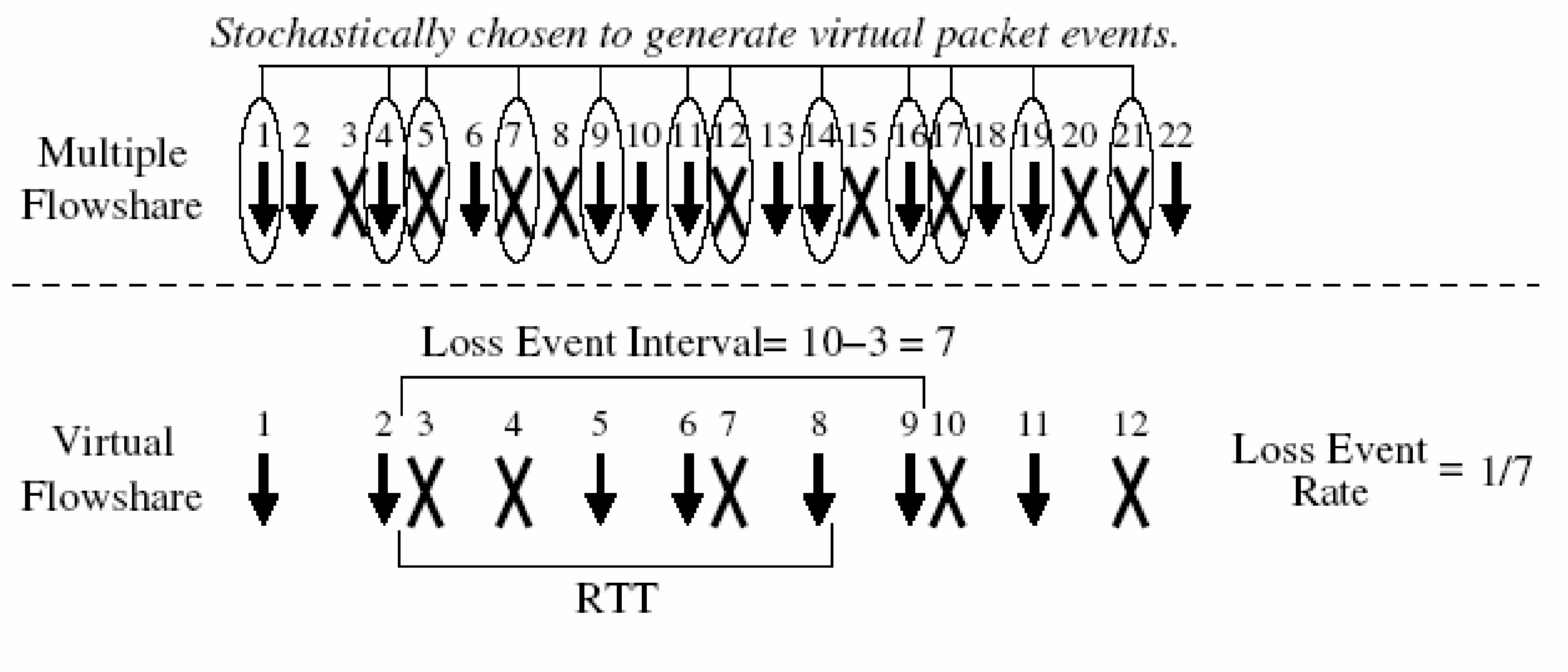

Fig. 9. Virtual packet event stream construction by BFLD.

- A random number *r* is generated in the interval $0 \le r \le 1.0$. If *r* is in the interval $0 \le r \le F$

MULTIPLE FLOWSHARES

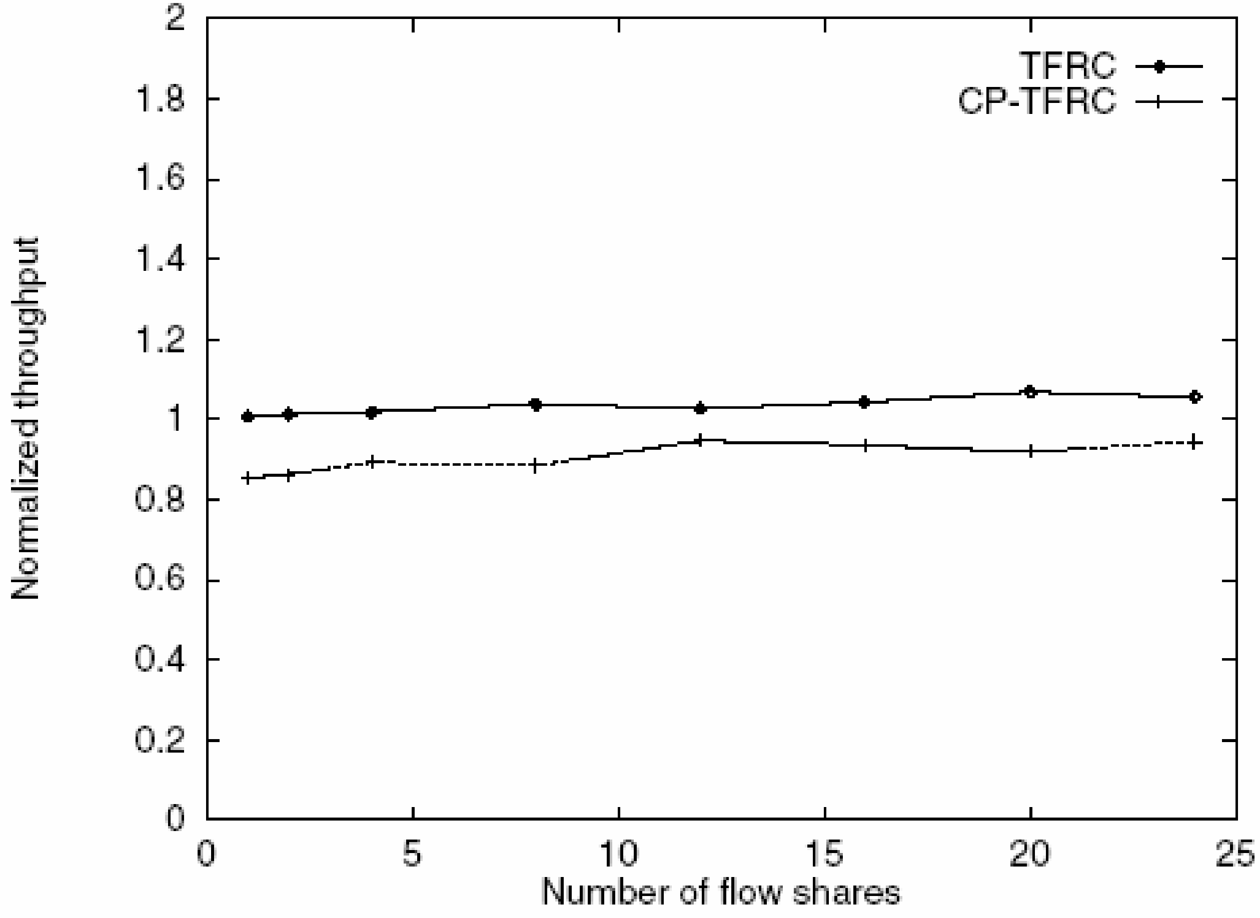

Fig. 10. Throughput for multiple flowshares using BFLD.

IMPLEMENTATION AND EVALUATION

- The Coordination Protocol using FreeBSD and Linux.
- Go on to present results showing how BFLD performs in an experimental network.
- Using UDP packets with CP packet headers nested within the first 20 bytes of application data.

IMPLEMENTATION AND EVALUATION

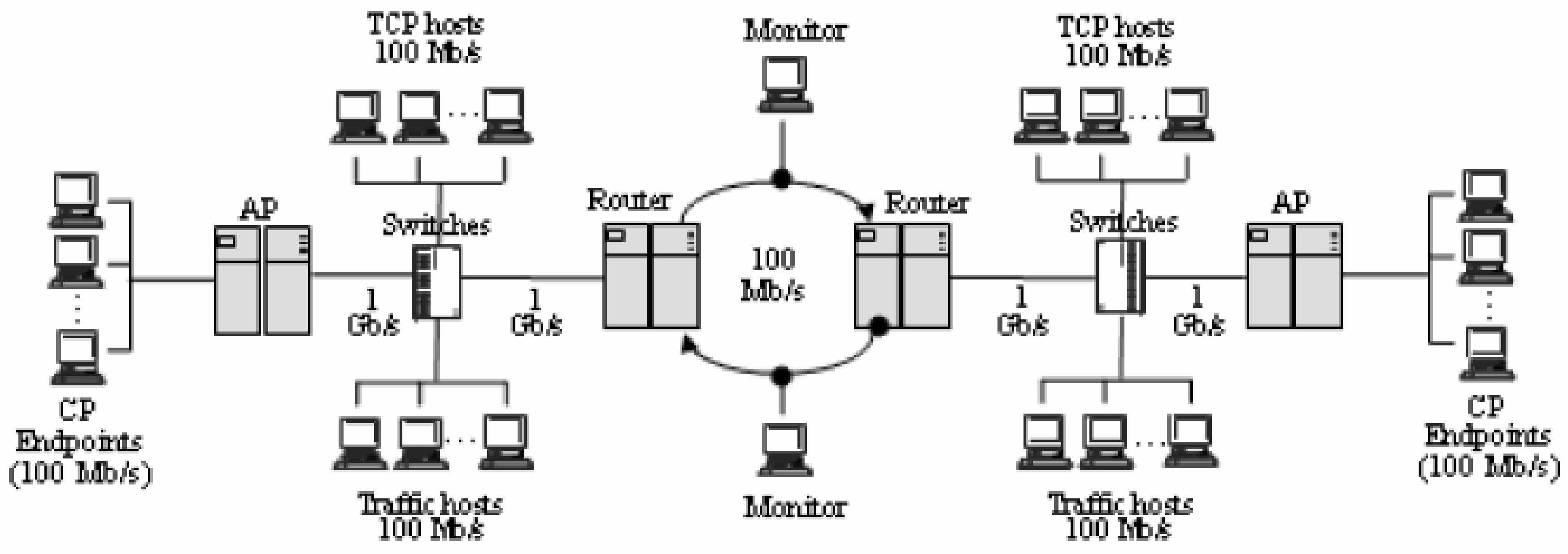

Fig. 11. Experimental network setup.

- Network monitoring：
  - First, used to capture TCP/IP headers from packets traversing the bottleneck.
  - Second, monitor queue size, packet forwarding events, and packet drop events.

IMPLEMENTATION AND EVALUATION

- *Normalized throughput ratio* ：
  – normalized average throughput for a single TCP flow to the normalized average throughput for a single CP flowshare.

- *coefficient of variance (C.O.V.)*：
  – the degree of throughput variation seen in aggregate TCP and CP traffic:

IMPLEMENTATION AND EVALUATION

- *Delay experiments*

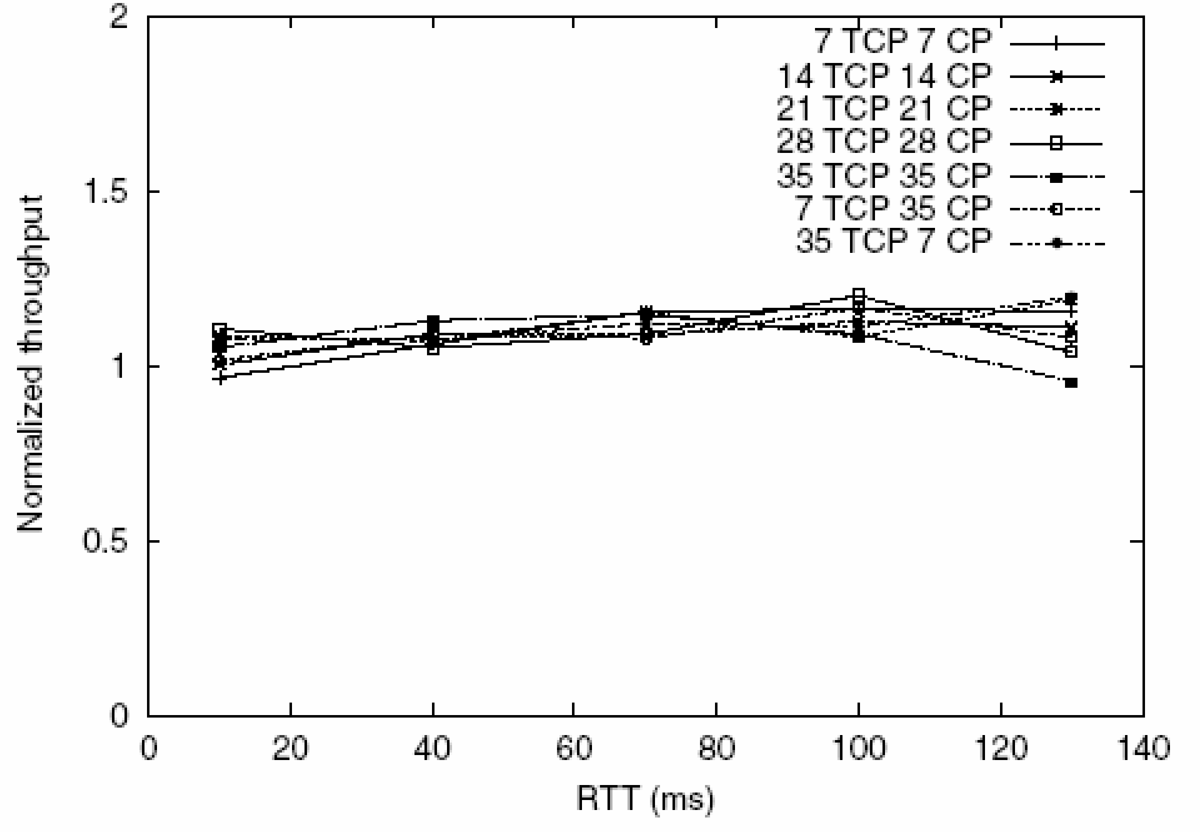

Fig. 12. Normalized throughput ratio as delay varies.

IMPLEMENTATION AND EVALUATION

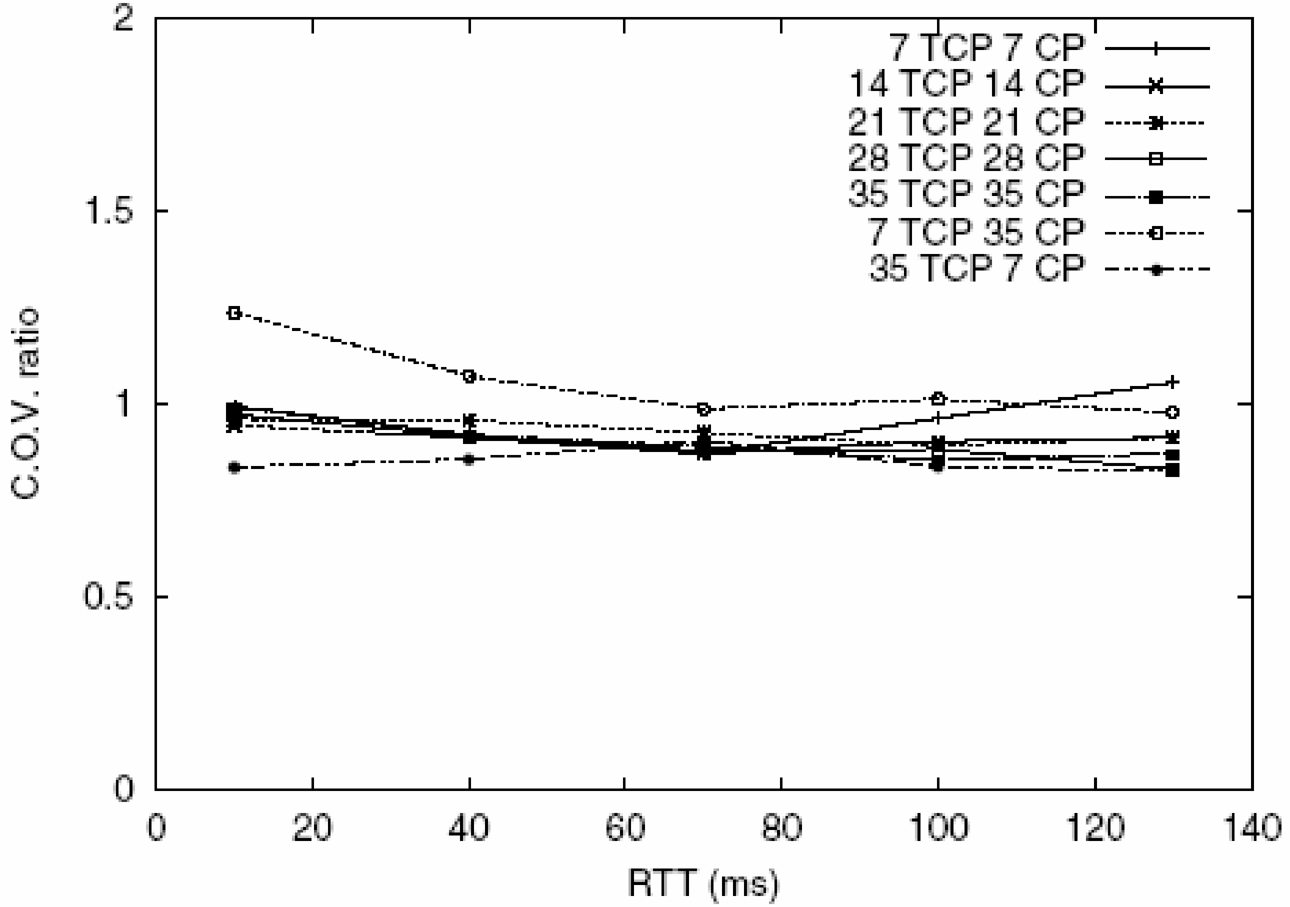

Fig. 13. C.O.V. ratio as delay varies.

IMPLEMENTATION AND EVALUATION

- *Bottleneck bandwidth experiments*

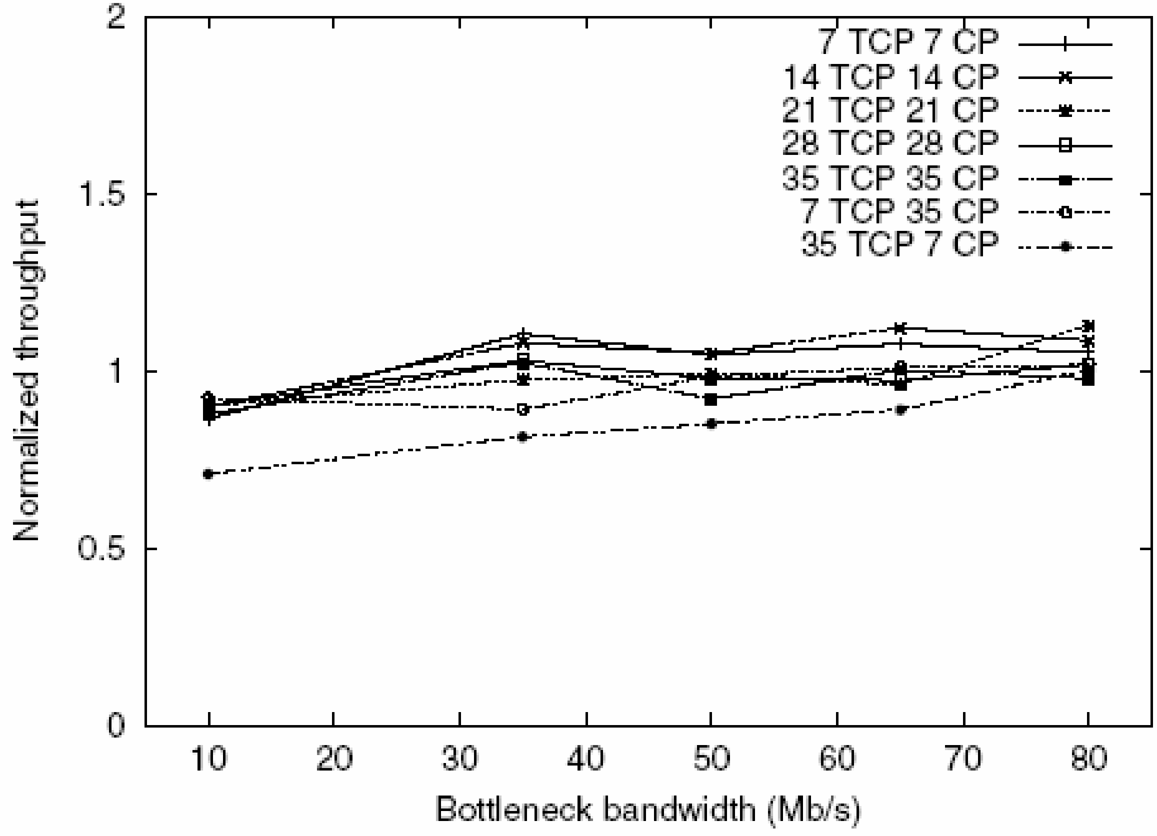

Fig. 14. Normalized throughput ratio as bottleneck bandwidth varies.

IMPLEMENTATION AND EVALUATION

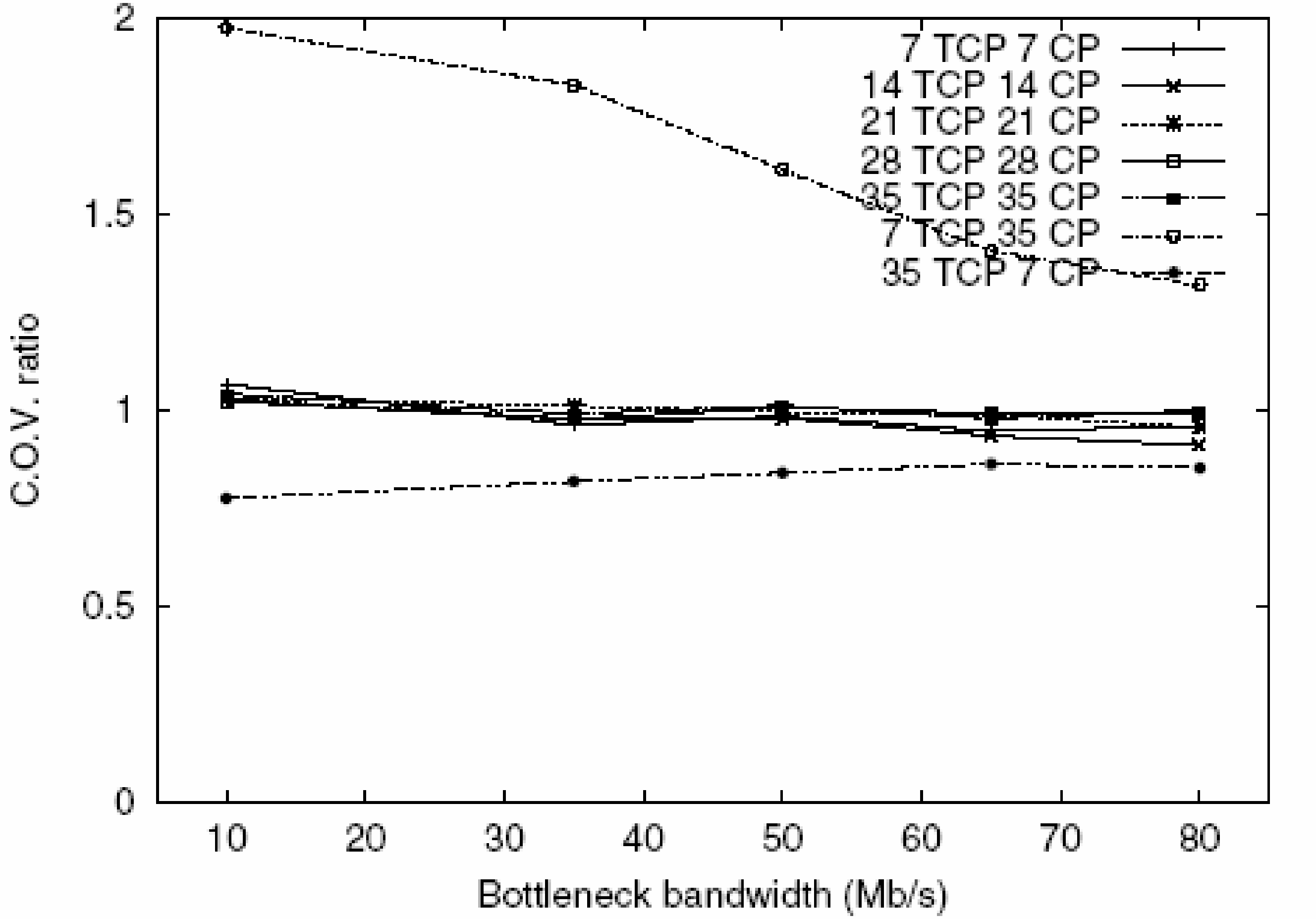

Fig. 15. C.O.V. ratio as bottleneck bandwidth varies.

IMPLEMENTATION AND EVALUATION

- *Random loss experiments*

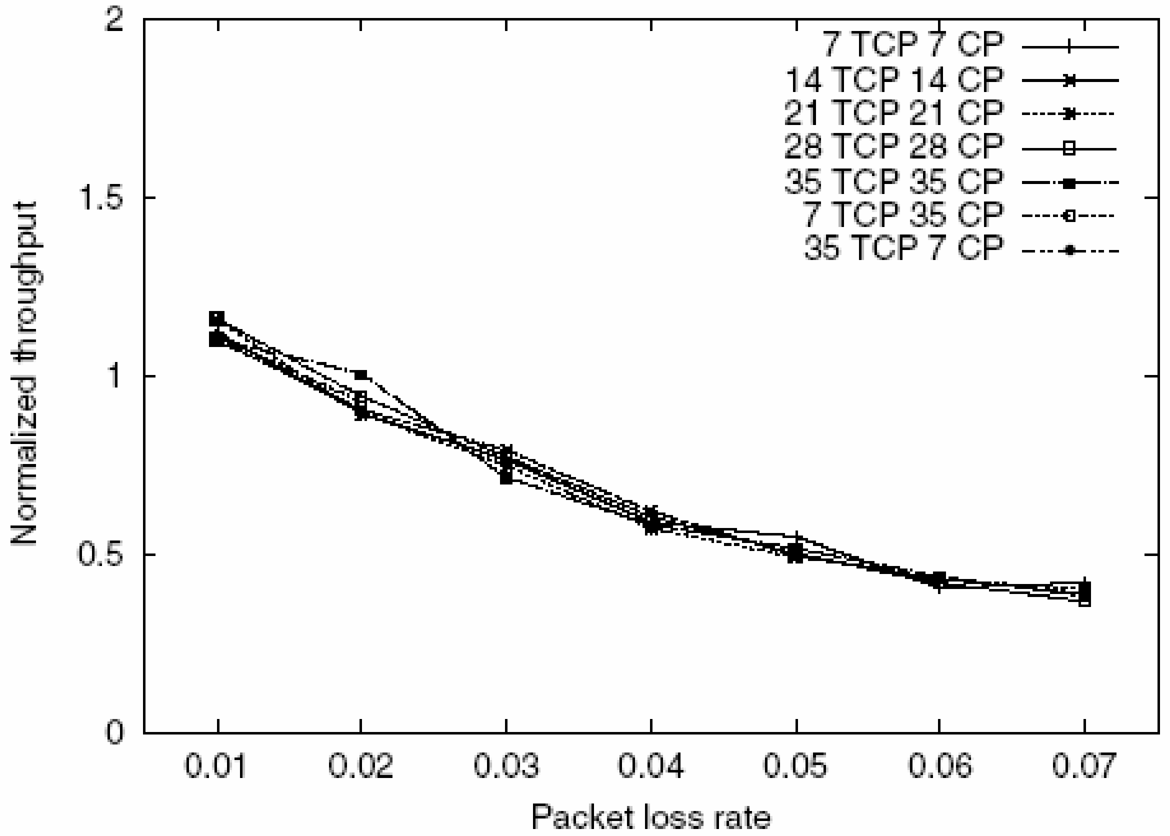

Fig. 16. Normalized throughput ratio as random loss varies.

IMPLEMENTATION AND EVALUATION

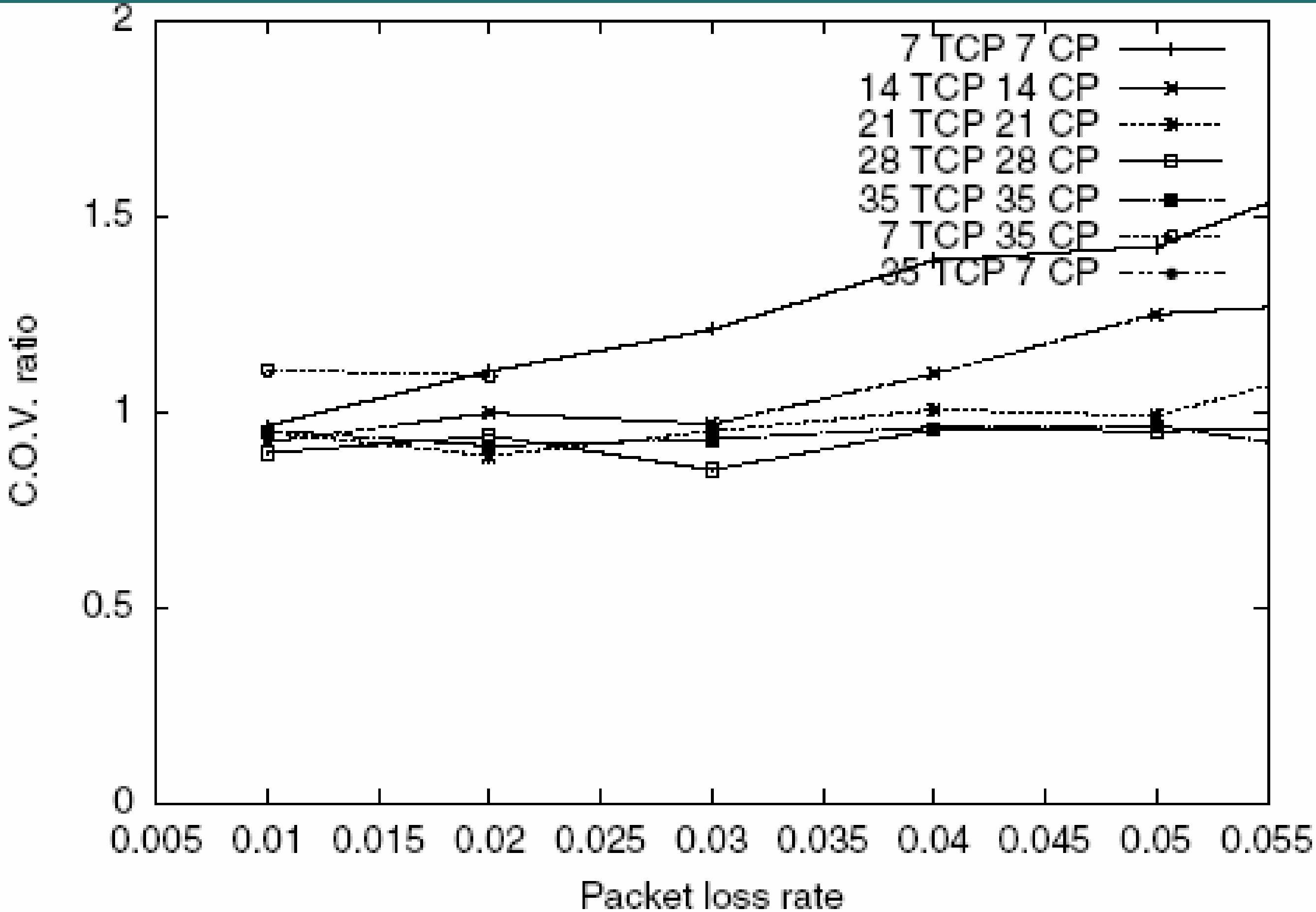

Fig. 17. C.O.V. ratio as random loss varies.

IMPLEMENTATION AND EVALUATION

- *Traffic load experiments*

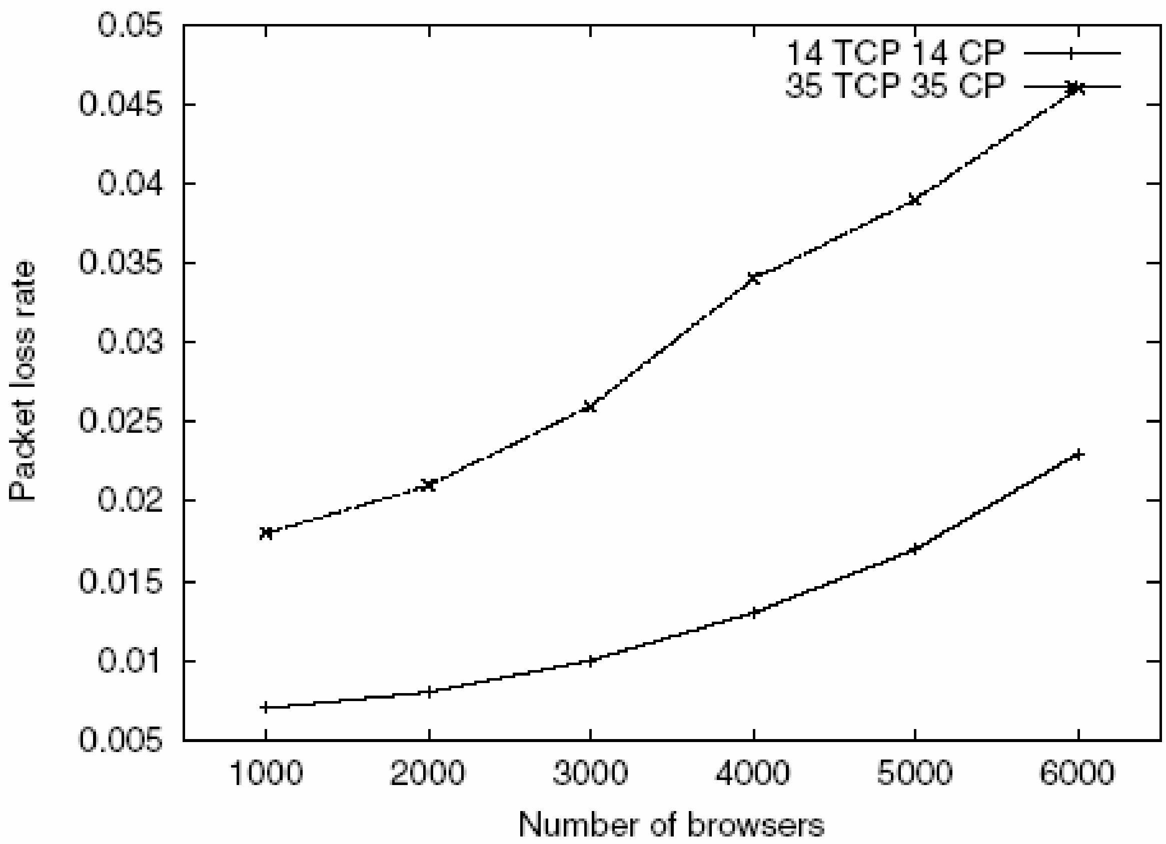

Fig. 18. Loss rates generated by background web traffic.

IMPLEMENTATION AND EVALUATION

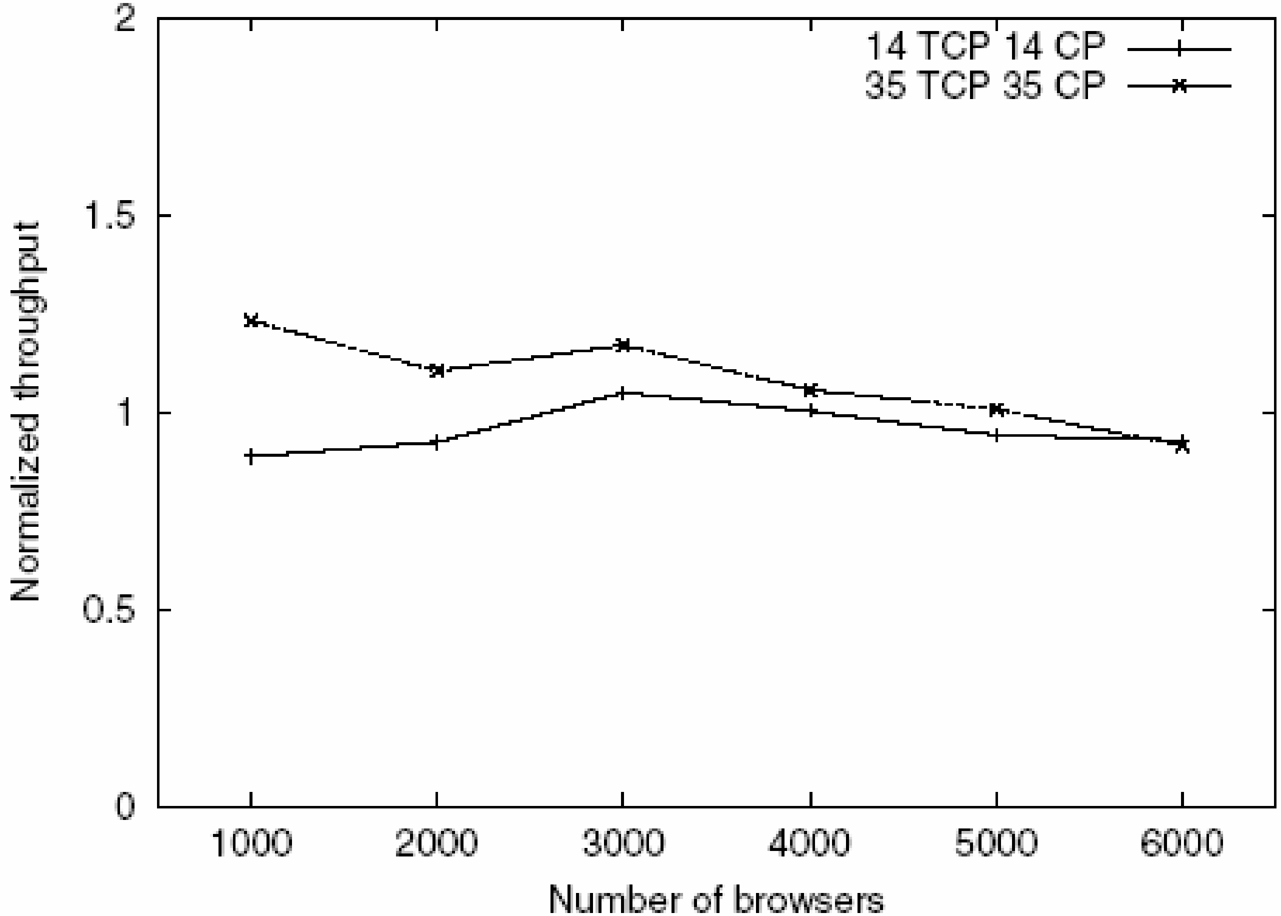

Fig. 19. Normalized throughput ratio as competing load varies.

IMPLEMENTATION AND EVALUATION

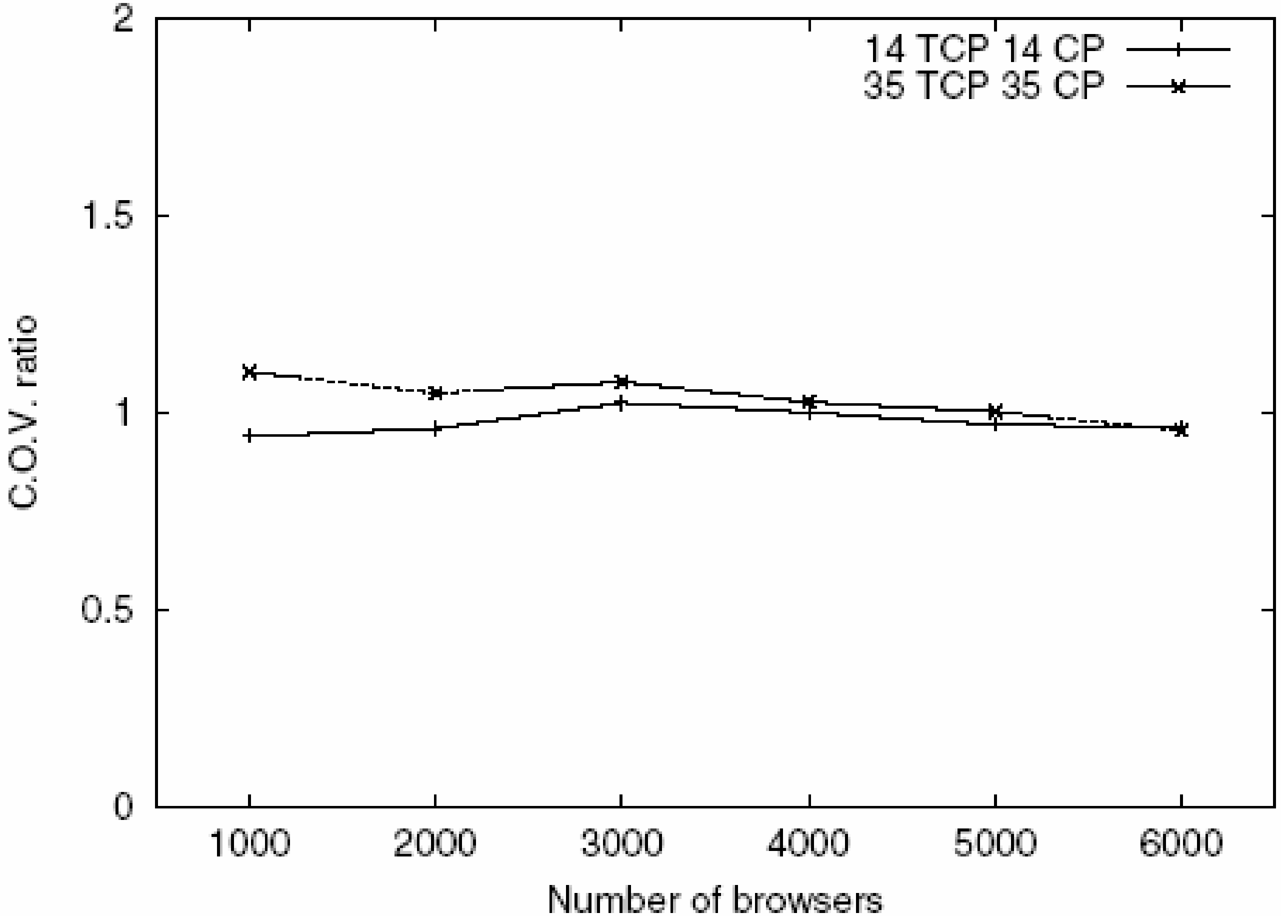

Fig. 20. C.O.V. ratio as competing load varies.

SUMMARY AND FUTURE WORK

- The *Coordination Protocol (CP)* works by providing network probe mechanisms that measure round trip time and packet loss for aggregate application traffic.
- Using BFLD, aggregate C-to-C traffic can effectively realize multiple flowshares.
- Finally, an issue we have considered for future work is the use of wireless endpoints within a C-to-C application cluster.