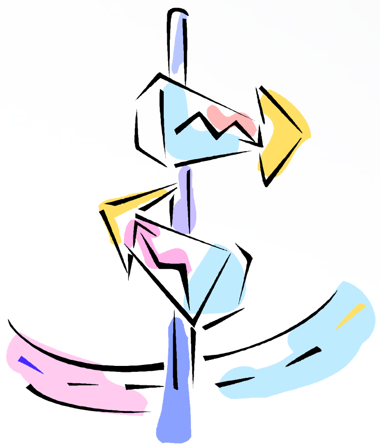
CMPT 885: SPECIAL TOPICS: HIGH-PERFORMANCE NETWORKS
FINAL PROJECT PRESENTATIONS
Fall 2003

TCP-Friendly Rate Control : An analysis

Charu Jain
www.sfu.ca/~cjain
cjain@cs.sfu.ca

Roadmap



- Introduction
- Ongoing Work
- Problem Statement
- Goal
- My work
- Results and Conclusion
- References

Introduction

- Network Congestion and growth in the number of Multimedia Applications
- UDP is unresponsive to congestion
  - Starvation of TCP traffic
  - Congestion collapse

Ongoing work : An Update

- End-to-End Vs. Active core debate
- TCP congestion ctrl. , TFRC, TFRC-PS (still an abstract concept)
- DCCP : UDP plus Congestion Control
- RED-PD (under development) is a flow-based mechanism that keeps state for just the high-bandwidth flows. RED-PD uses the packet drop history at the router to detect high-bandwidth flows in times of congestion and preferentially drop packets from these flows.
- ECN

Problem Statement

- For some applications such as Internet Telephony, it is more natural to adjust packet size while keeping the packet rate as constant as possible.
- Many VoIP applications use 20-30 ms packets despite low payload efficiency, in order to keep end-to-end delay lower than 150 ms.
- Rate varying congestion control mechanisms are NOT suitable for Internet Telephony.

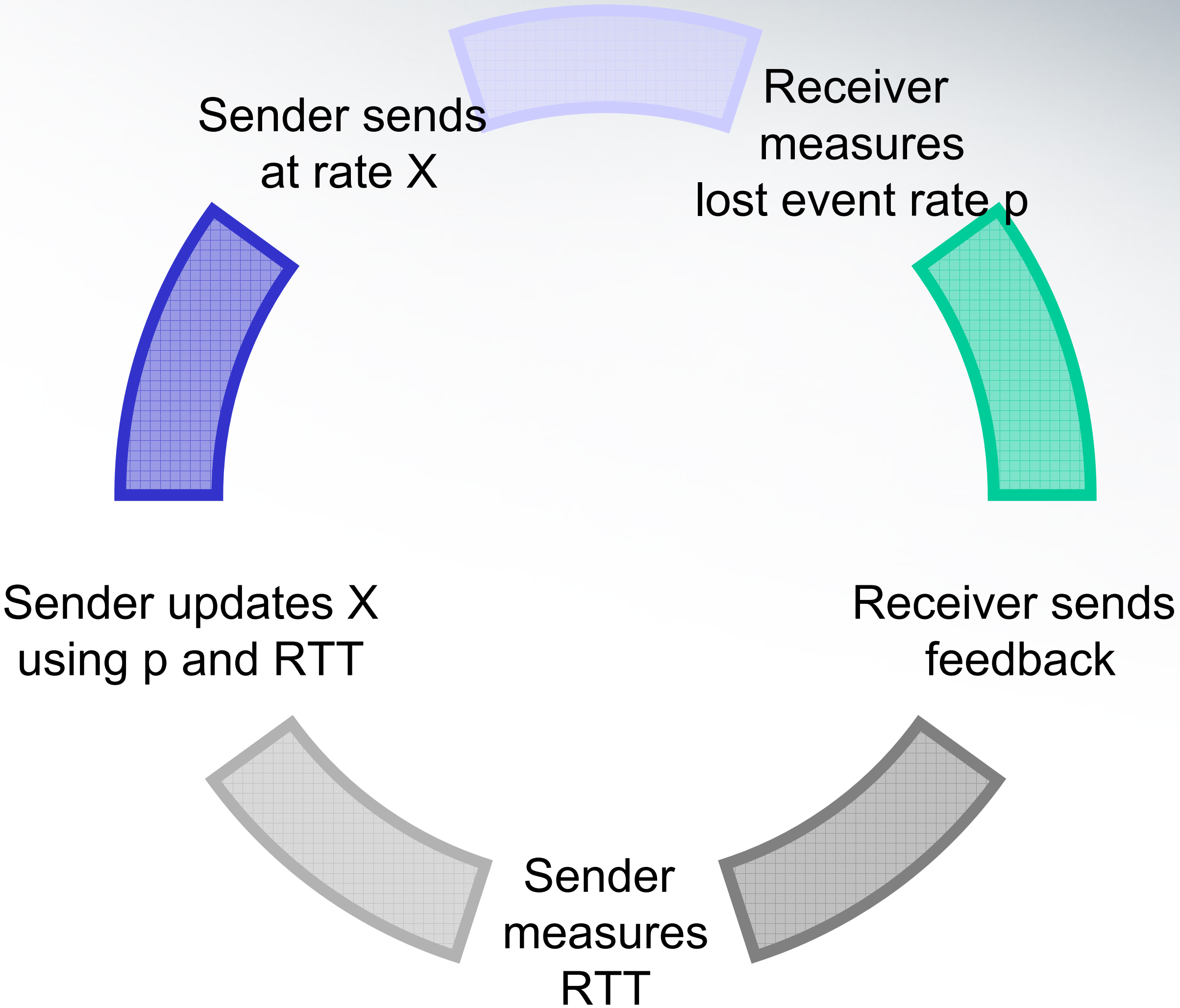
Goal

- To devise and implement a simplified protocol that is suitable for VOIP-like applications and at the same time responds constructively to Congestion.

Why can’t TFRC work for VoIP?

- TFRC is a congestion control mechanism designed for unicast flows operating in an Internet environment and competing with TCP traffic
- TFRC is a receiver-based mechanism, with the calculation of the congestion control information in the data receiver rather in the data sender.
- Documented in RFC 3448, by M. Handley, S. Floyd, J. Padhye, J. Widmer
- Category: Standards Track, Released January 2003

Protocol Mechanism



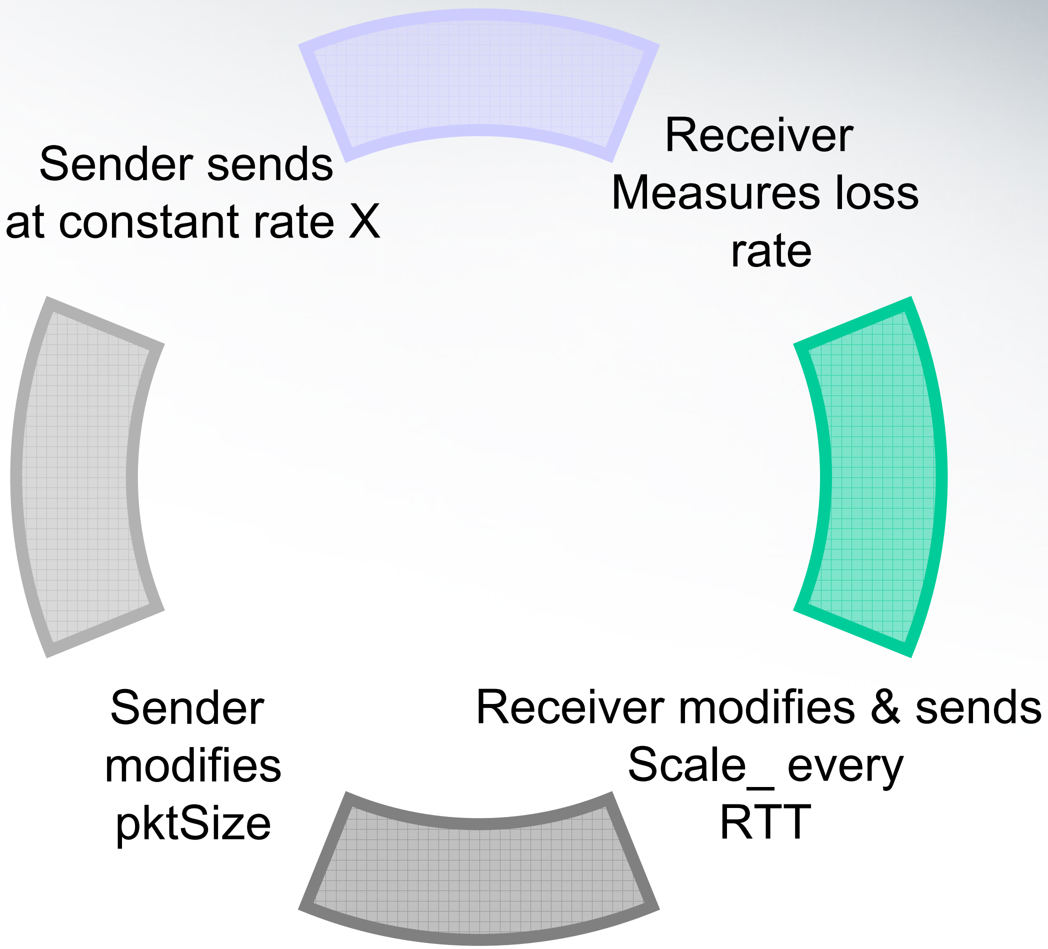
- There is a clear tradeoff between payload efficiency (payload/total packet size) and packetization delay (time to fill a packet), and one way to increase bandwidth efficiency would be to accumulate many audio frames within the same packet. VoIP apps prefer to use small packets
- Thus, audio sources typically generate packets at a constant rate and perform congestion control by switching codecs, which has the effect of varying their packet size
- Other applications may be driven to adjust the packet size independently of congestion control (for example: a high bit error rate in a wireless environment induces a small packet size).
- Such applications have a variable packet size and simply adjusting their packet rate as though packets were path-MTU-sized is clearly not fair.

- What happens when a rate-control mechanism like TFRC is used for an application that uses variable pktSizes?
- Similar to TCP, where commonly only one window reduction per congestion window is possible, a loss event is defined as one or more packets lost during one RTT (i.e., packet loss during an RTT is aggregated to a single loss event). Using the reference packet size in the equation results in a higher packet rate.
- The higher the number of packets per RTT, the more likely it is that multiple lost packets will be aggregated to a single loss event and the average number of loss events per packet will decrease, resulting in a strong bias *in favor* of sending small packets at a high rate.

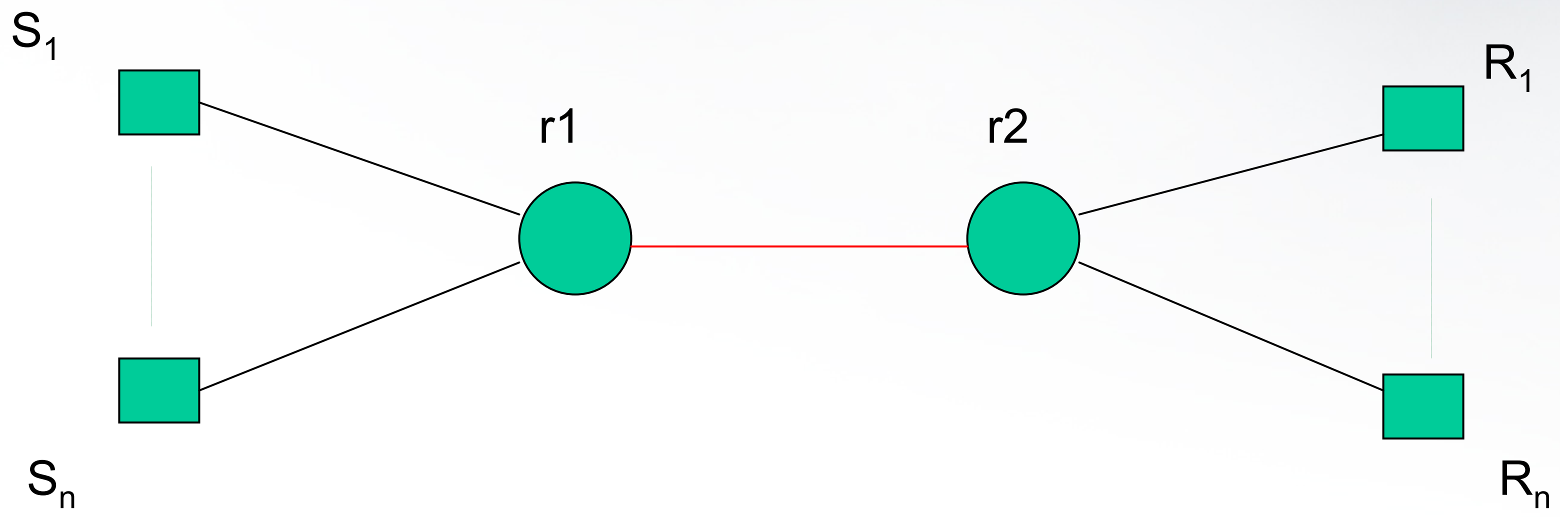
Packet-Size Scaling Protocol (PSP)

- VoIP data packets ↔ RTP ↔ UDP ↔ IP I,II layers
- Modifications at the transport layer
- Simple N-level packet size scaling mechanism

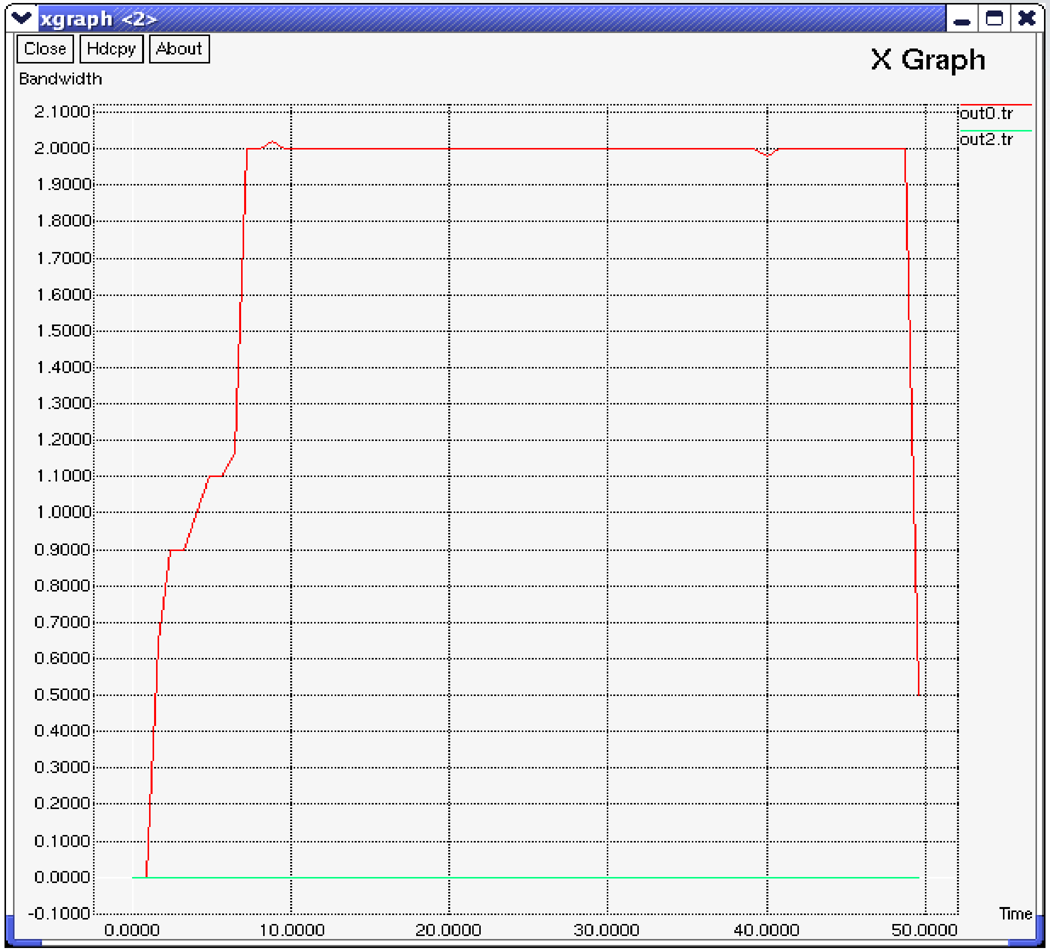
Protocol Mechanism



Simulation Topology



PSP

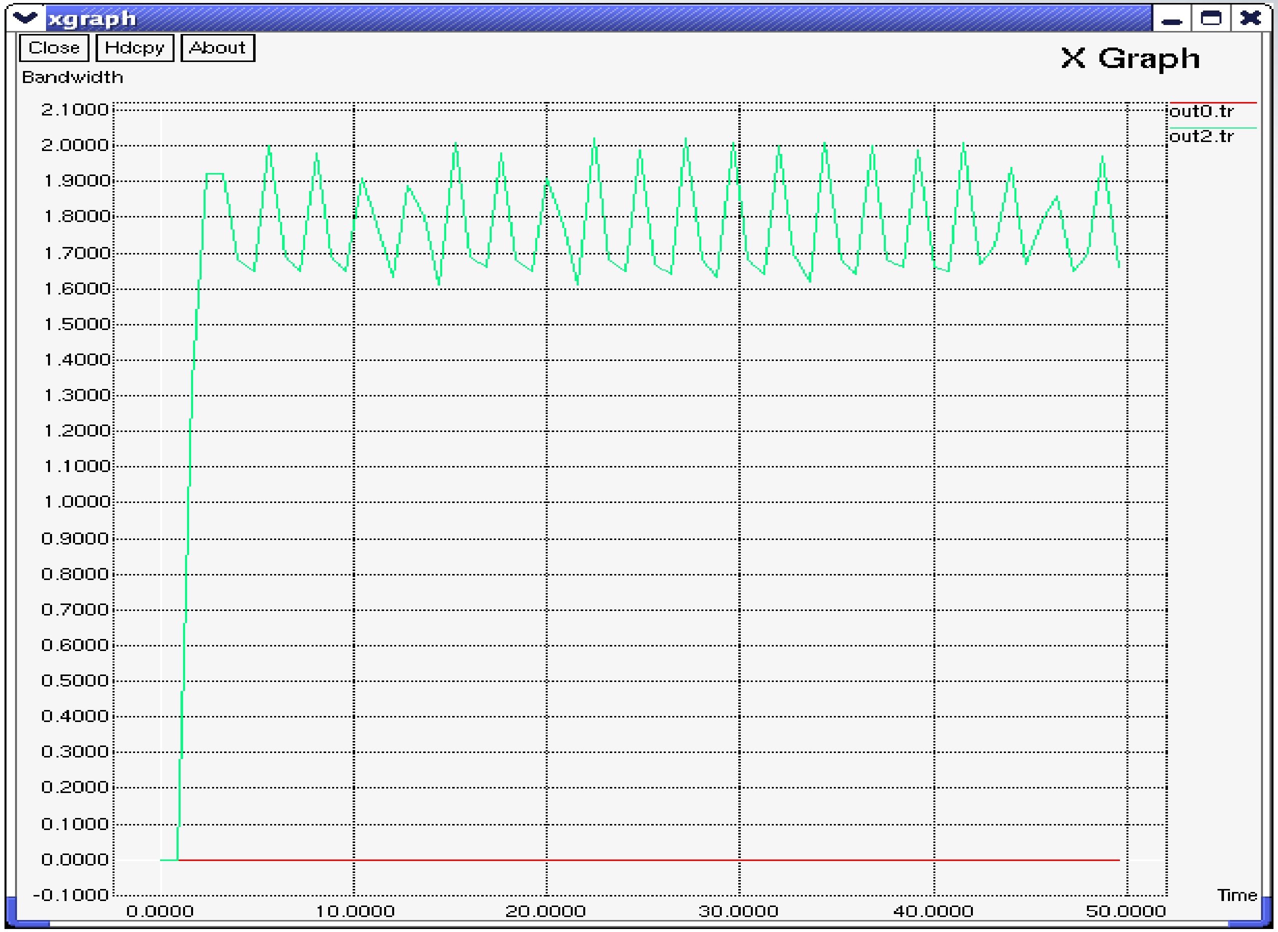


xgraph <2>
Close
Hdcpy
About
X Graph
Bandwidth
2.1000
2.0000
1.9000
1.8000
1.7000
1.6000
1.5000
1.4000
1.3000
1.2000
1.1000
1.0000
0.9000
0.8000
0.7000
0.6000
0.5000
0.4000
0.3000
0.2000
0.1000
0.0000
-0.1000
out0.tr
out2.tr
Time
0.0000
10.0000
20.0000
30.0000
40.0000
50.0000

PSP

TFRC

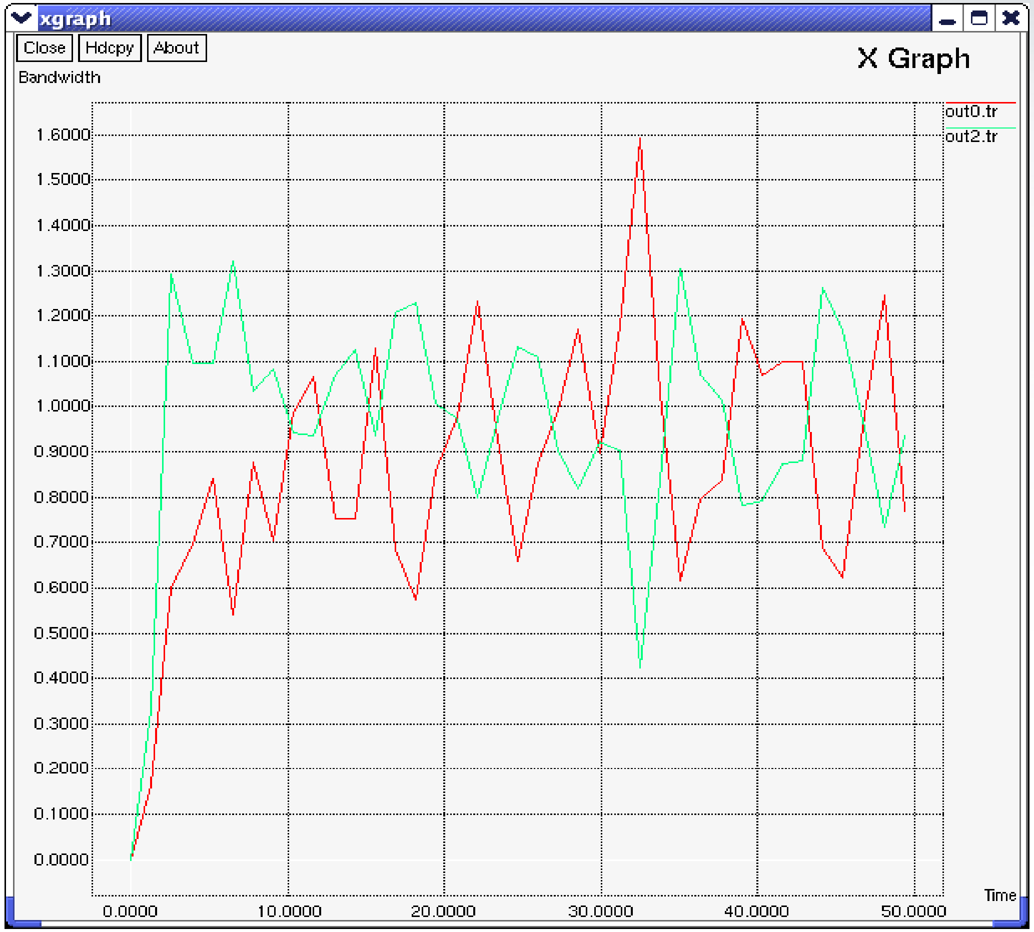
TCP alone



PSP

TCP

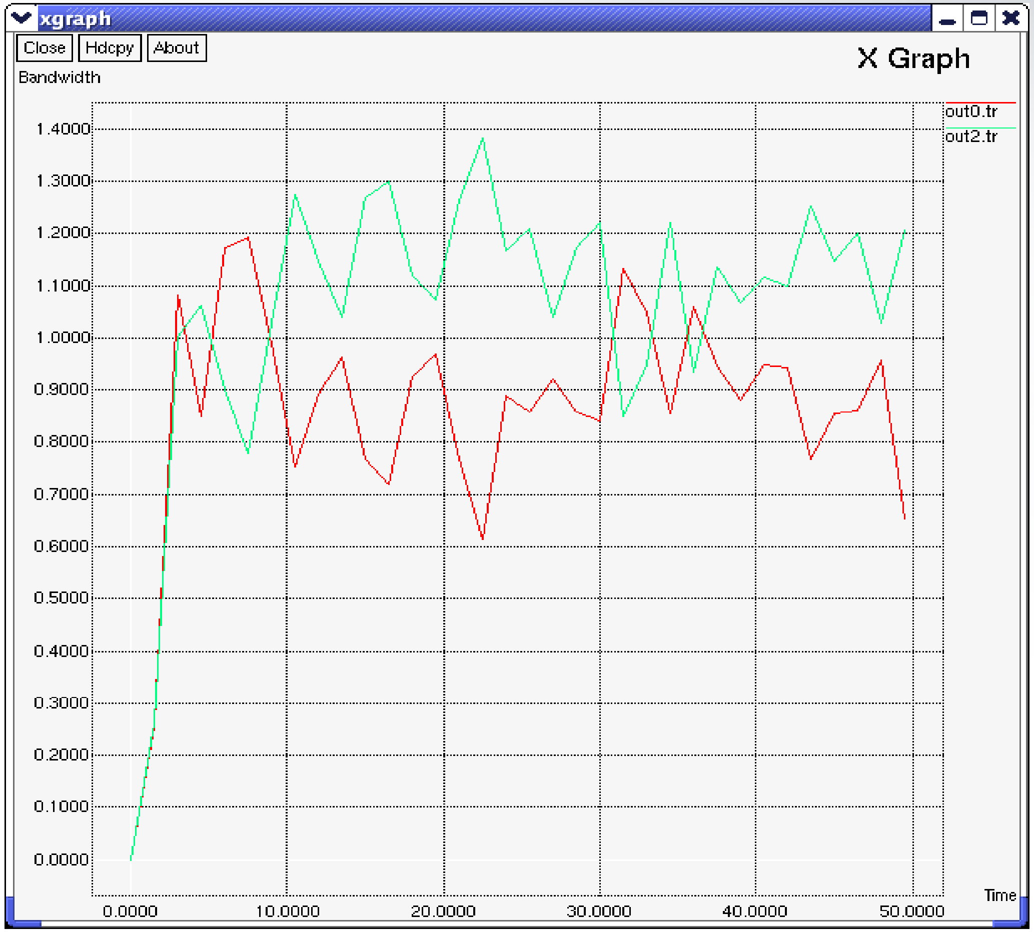
TCP Vs PSP



PSP

TCP

TFRC Vs PSP



PSP

TFRC

Conclusions

- PSP is TCP-Friendly
- It quickly achieves it’s fair share of Bandwidth.
- No demands of high processing capacity at the receiver-end
- It is suitable for applications that choose to maintain a high rate at the expense of reduced packet size.

- Optimum results were obtained when
  - PS_MAX ≤ MTU
  - Constant Rate, X ≤ (Bottleneck Bandwidth / PS_MAX)
  - PS_MIN ≥ PS_MAX / 4
- 1% packet loss due to congested queue at the router (with TCP)

Future work

- A connection establishment phase, in order to automatically populate the “Preset packet-sizes” table and settle upon a an optimum constant transmission rate. These are important parameters in a Bandwidth limited environment to ensure a fair share of resources.
- Study PSP behavior with RED-PD.
- What compromises does a simplified mechanism like PSP entail? What applications can do away with it?

References

- [1] S. Floyd and K. Fall, “Promoting the Use of End-to-end Congestion Control in the Internet,” *IEEE/ACM Trans. Net.*, vol. 7, no. 4, Aug. 1999, pp. 458–72.
- [2] *Widmer, J.; Denda, R.; Mauve, M.;* “ A survey on TCP-friendly congestion control “ Network, IEEE , Volume:15 Issue:3, May-June 2001 Page(s): 28 -37
- [3] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [4] Balakrishnan, H., Rahul, H., and S. Seshan, "An Integrated Congestion Management Architecture for Internet Hosts," Proc. ACM SIGCOMM, Cambridge, MA, September 1999.
- [5] Handley, M.; Floyd, S.; Padhye, J.; Widmer, J.; “TCP Friendly Protocol Specification (TFRC): Protocol Specification”; RFC 3448, January 2003.
- [6] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, NJ, June 1999. 20 http://citeseer.nj.nec.com/padhye99model.html

- Interrogation
- Cross-examination
- Third degree