The Streaming Media Congestion Control (SMCC) Protocol

N, Aboobaker, D. Chanady, M. Gerla, M. Sanadidi

UCLA Computer Science Department

www.cs.ucla.edu/NRL/HPI

Streaming Media Transport Requirements:

- Minimal fluctuations in packet reception rate
- Looser error control (and in-order delivery) requirement

Congestion Control Schemes Addressing Streaming Media Requirements:

- TCP Friendly Rate Control (TFRC)
- Time-lined TCP (TLTCP)
- RAP
- SQRT
- SCTP
- Multimedia Streaming TCP-Friendly Protocol (MSTFP)
- RTC

Limitations of Previous Schemes

- Adhere to blind *multiplicative rate decrease*, and in some cases *Slow Start Phase* (TLTCP, SCTP, RAP, SQRT) => unwarranted reduction and higher fluctuation in sending rate
- Provide TCP-like *error control* functionality (TLTCP, RAP, SQRT) => overhead and lower throughput
- Mimicking *TCP connection throughput* (TFRC): possible lower bandwidth utilization (in error loss case)
- Active specific probing to determine path bandwidth (RTC): overhead in sending additional packets specifically for bandwidth estimation function

SMCC: Streaming Media Congestion Control

- Goals:
  - Avoid delivery rate fluctuations and unnecessary reduction in sending rate
  - Efficient utilization of path bandwidth, while remaining friendly to TCP
  - Limited error control functionality

- Features:
  - Rate based congestion control
  - Sending rate adjusted according to share estimation as in *TCPW* methods, BUT: *estimation done at receiver* and fed back to the sender
  - Continuously in "Congestion Avoidance" mode
  - Friendliness enhanced by filtering of "compressed" bandwidth samples
  - Limited error control via NACKs, Occasional ACKs only for RTT monitoring

SMCC: Message Formats

Client Initiation: Start three-way handshake

C	GET	[filename]

Server Response: Contains the number of layers in the video, each layer's data rate

C	[# of layers]	[1^{st} layer rate…	last rate]	[starting sequence #]

Client Information: Reports the size of the client's buffer to the server, in bytes

C	BUFFERSIZE	[buffer size]

Message Formats (Continued)

ACK: Sent when the client receives a *Video Data* packet that the server requests to be acknowledged. Used for updating the server's round trip time estimate.

C	ACK	[SEQ #]			

NACK: Contains the sequence number of the lost packet and the BWE, at the time the lost packet is detected.

C	NACK	[SEQ #]	[BSE]		

Video Data: Video time is time data should be played out, frame number within the video, size of the video data in the packet, the packet number within the frame, and the video data. The last field in this packet specifies whether the sender is requesting an acknowledgement for this packet.

D	[SEQ #]	[layer number]	[sender time]	[video time]
[frame]	[data size]	[packet number]	[video data]	[request ack]

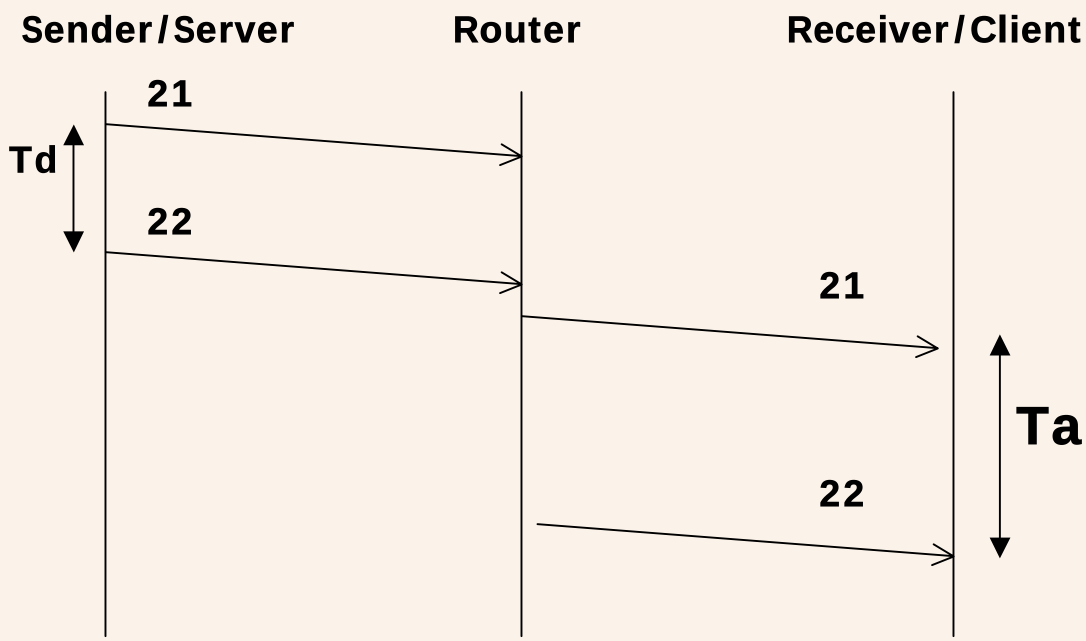
Bandwidth Estimation in SMCC

At Receiver (Client) Side:

- Ignore
  - (1) “compressed samples”,
  - (2) sample where packet loss is detected, i.e. nonconsecutive sequence numbers
- Otherwise, Bandwidth sample is:

  Data bytes delivered in last packet / Time between last two packets receptions
- Bandwidth samples are *exponentially filtered* to determine current share estimate
- If packet loss is detected, Receiver sends a NACK to Sender including the latest share estimate

Filtering Bandwidth Sample for Friendliness



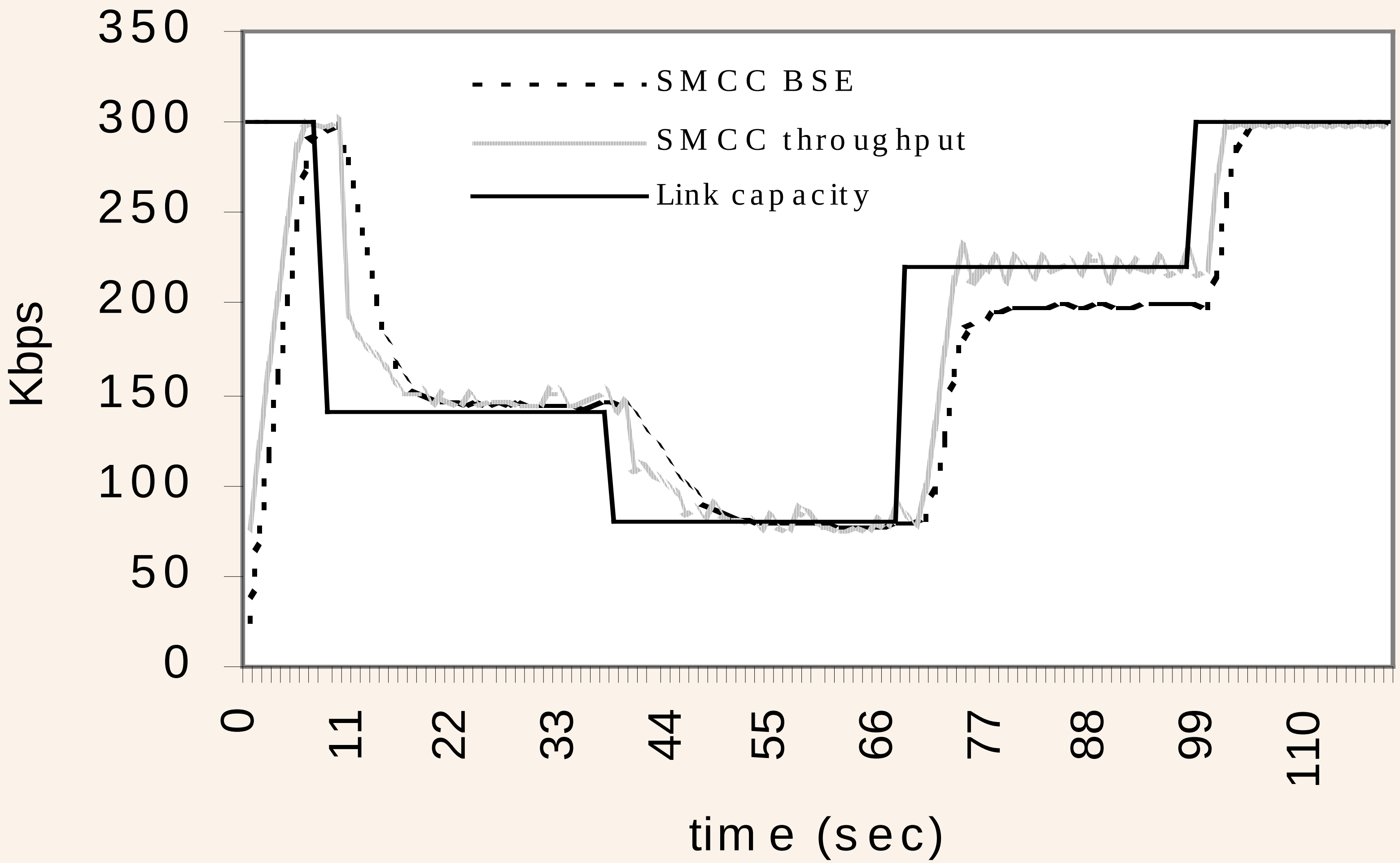
Good bandwidth sample: Ta > Td, and no missing sequence number

Sending Rate Adaptation

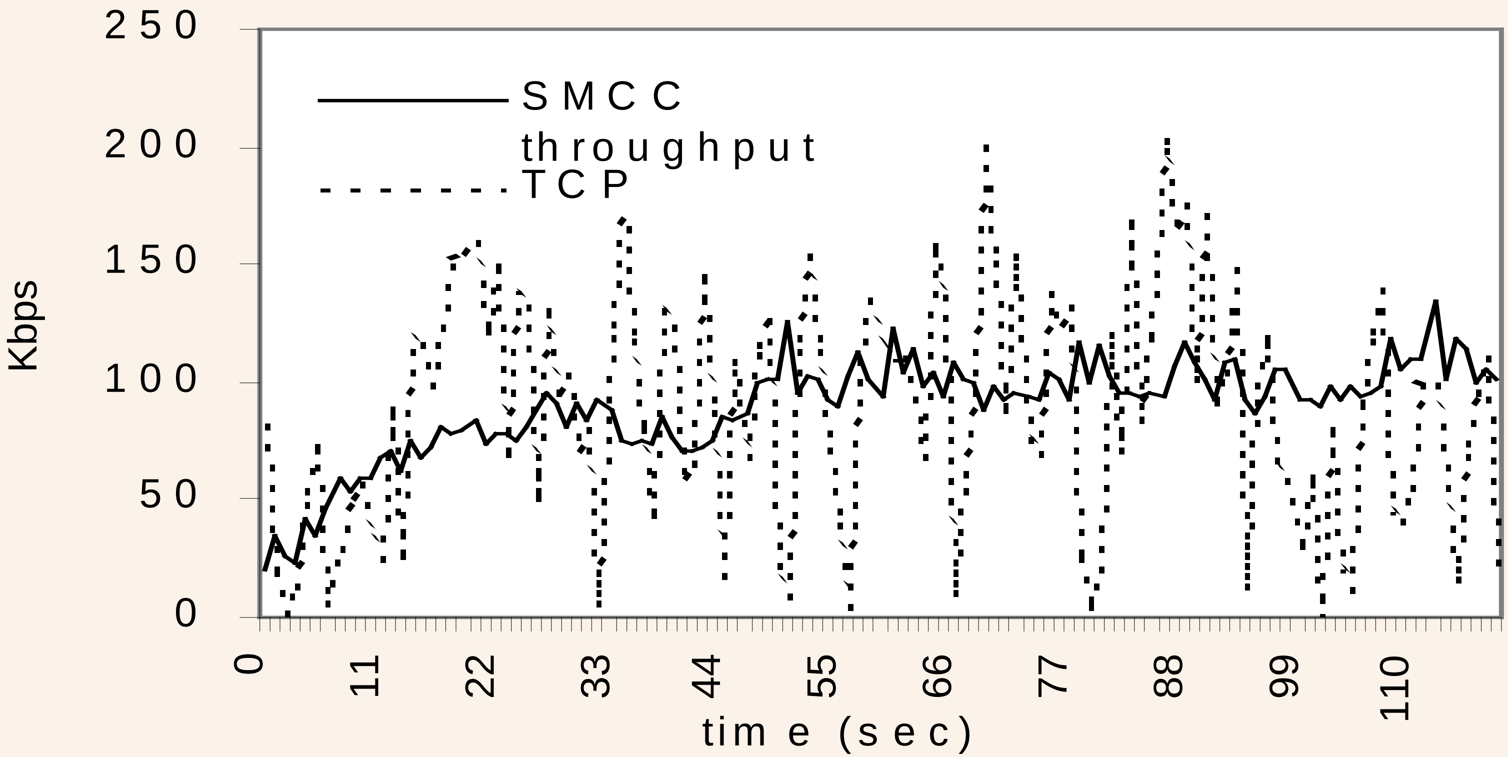
At Sender (Server) Side:

- Sender is continuously in "*congestion avoidance*" mode; probing: sending rate is *increased* by 1 packet every RTT
- Upon reception of NACK, Sending Rate is *adjusted* according to BWE received in the NACK
- Sender *retransmits* a packet upon receiving a NACK, only if application requires it

Bandwidth Share Estimation Accuracy

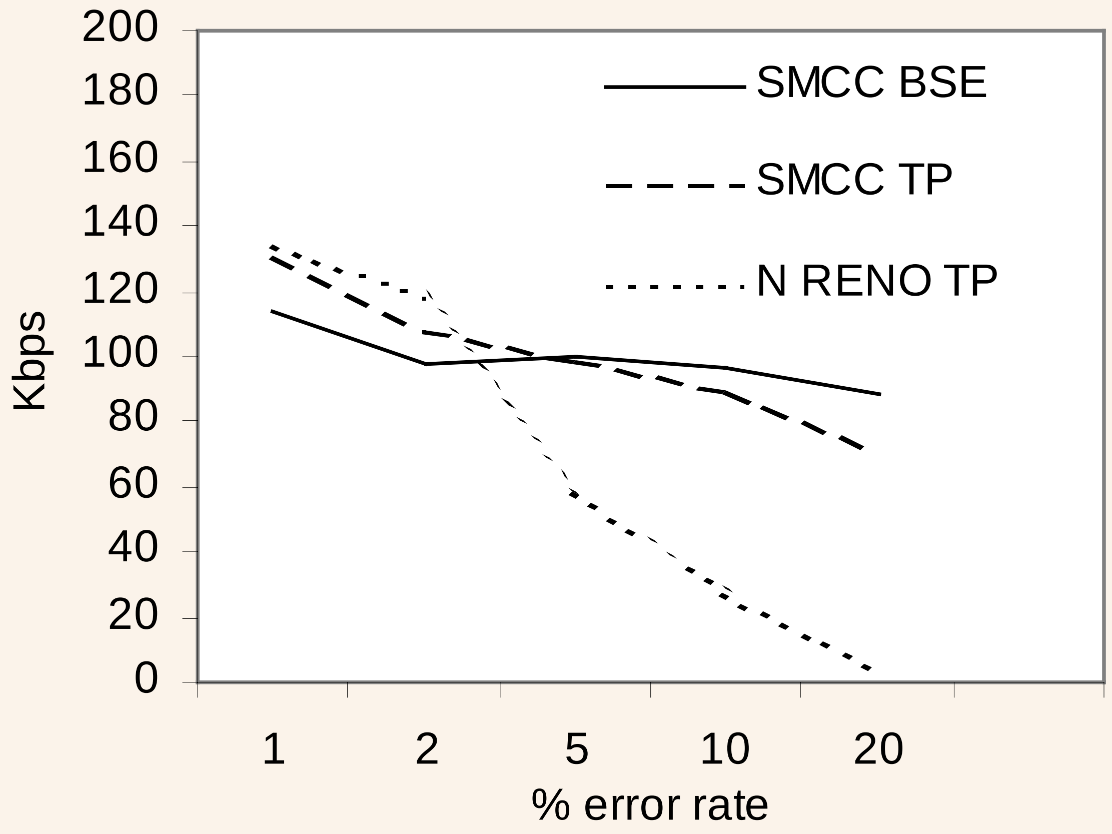


Delivery Rate Fluctuations

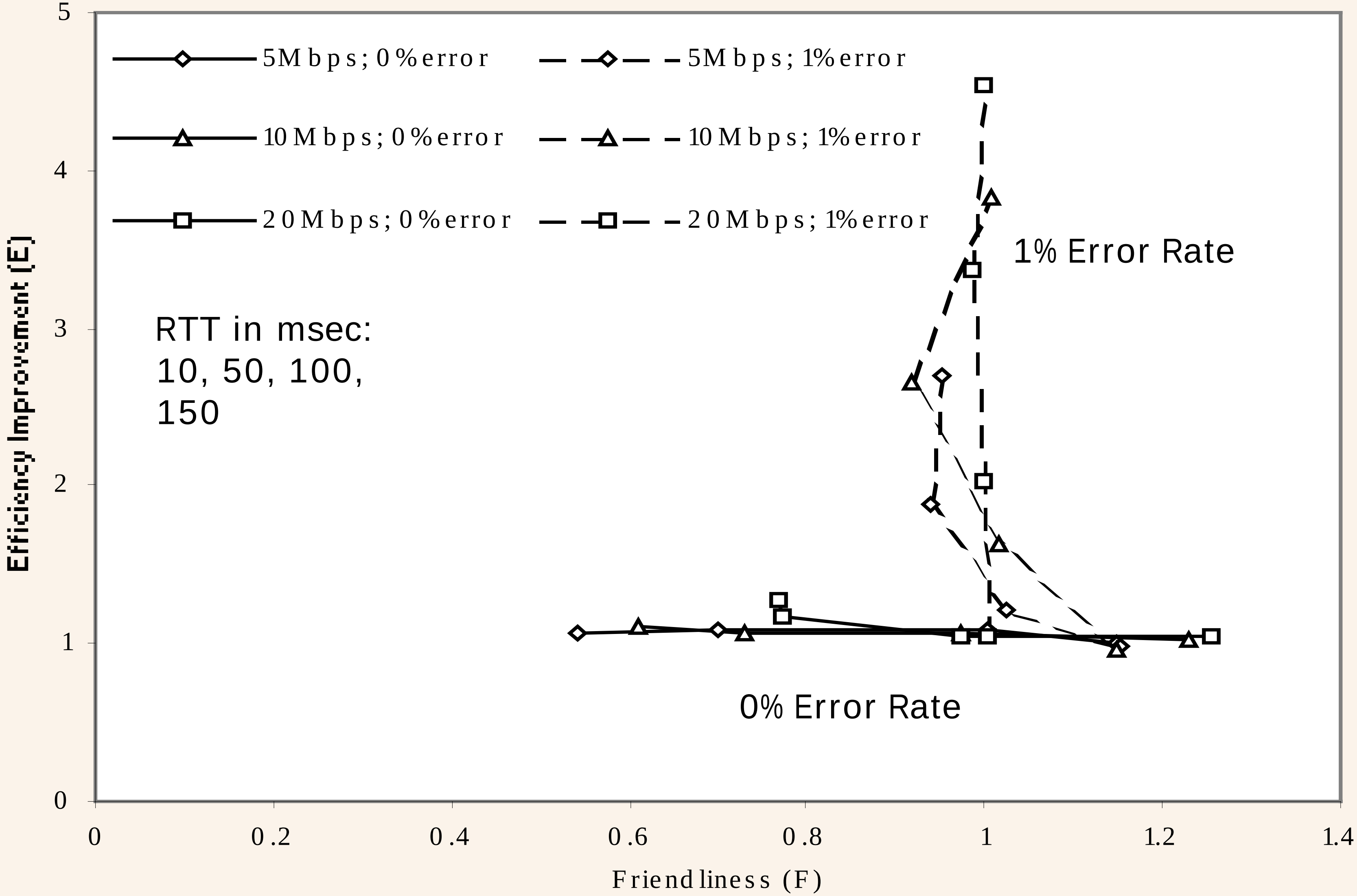


One SMCC and One TCP Reno Connection Sharing a Bottleneck

Throughput Under Random Error Losses



SMCC Efficiency/Friendliness Graph



SMCC Summary

- SMCC is efficient and smooth transport protocol for streaming media traffic, with congestion control and limited error control functions
- Friendliness in congestion loss cases to be enhanced by techniques similar to TCPW ASF
- Future work:
  - Compare to performance of other media congestion control schemes
  - Adapt SMCC to effectively interface to existing codec schemes, and evaluate video quality delivery; e.g. with MPEG-4
  - Include adaptive sampling and filtering estimation as in TCPW ASF