Enhancing TCP for Networks with Guaranteed Bandwidth Services

Yujie Zhu, Oyebamiji Oladeji, Kyu-han Kim and Raghupathy Sivakumar

GNAN Research Group, Georgia Tech

http://www.ece.gatech.edu/research/GNAN/

Introduction

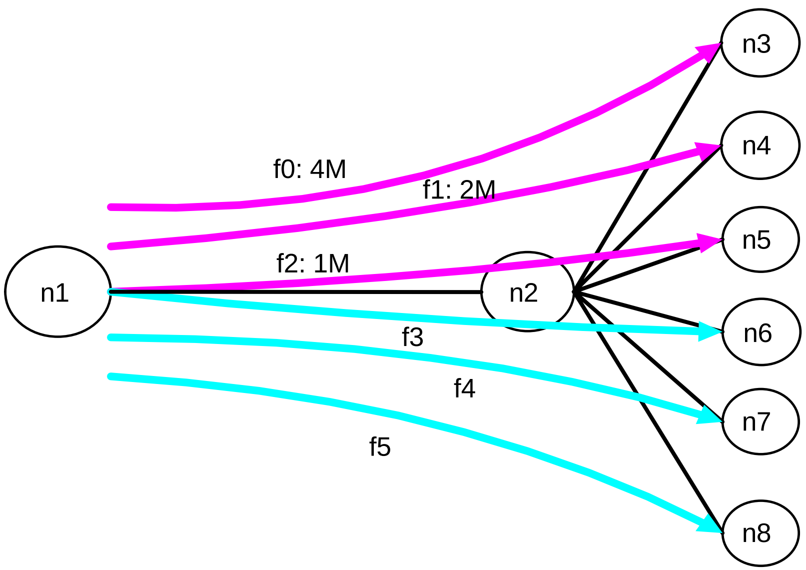
- The Internet is envisioned to be able to provide QoS services in the near future (*diffserv, intserv*)
- Consider applications that can enjoy bandwidth provided by both the *intserv guaranteed service* and the *intserv best effort service*
- Problem Statement: How can a transport layer protocol deliver to such applications the ideal aggregate (reserved + best-effort) bandwidth, while providing TCP's end-to-end semantics?
- We propose GTCP, an enhanced version of TCP tailored for bandwidth guaranteed environments

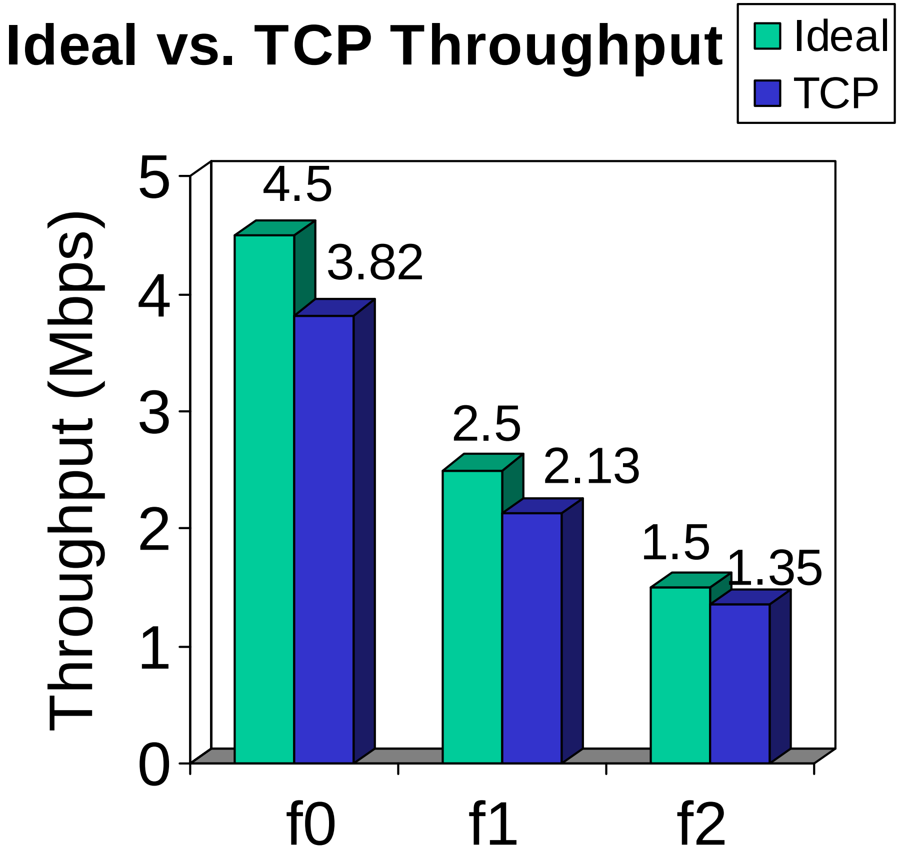
TCP over Bandwidth Guaranteed Networks

Bottleneck link 10Mbps, 10ms

f0 – f2: 4Mbps, 2Mbps and 1Mbps reservation

f3 – f5: best effort





Georgia Institute of Technology

GNAN Research Group

TCP Congestion Window Adaptation

- Slow start: *cwnd = 1; cwnd++ for every ACK;* exit slow start when *cwnd > ssthresh*
- Loss indicated by 3 DUPACKs: *cwnd = cwnd/2*
- Retransmission Timeout: *cwnd = 1,* re-enter slow-start
- Illustration: reserved bandwidth = 50, cwnd = 80

	TCP Behavior		Ideal Behavior	
Orignal cwnd size = 80	$cwnd_{BE}$		$cwnd_G$ | $cwnd_{BE}$	
cwnd size after 3 DUPACKs	$cwnd_{BE}/2$	=40	$cwnd_G$ | $cwnd_{BE}/2$	=65

TCP's *cwnd* adaptation is unaware of the reserved bandwidth!

Georgia Institute of Technology

GNAN Research Group

TCP Self Clocking

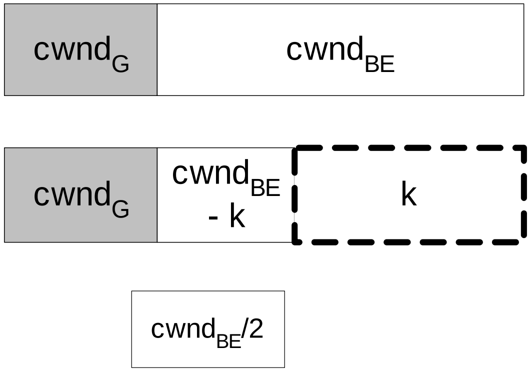
- The receipt of an ACK for a packet triggers expansion of congestion window and transmission of a new packet
- However, when there are packet losses, self-clocking is stalled
- Problems exist even when TCP-NewReno is used, as long as the number of packet losses exceed a threshold

$k > cwnd_{BE}/2$

original cwnd size

outstanding packets after k losses

best effort part of cwnd after cut down



n ... d_G!

TCP's *self-clocking* is unaware of the reserved bandwidth!

Georgia Institute of Technology

GNAN Research Group

Ideal Transport Protocol Design Goals

- Reserved Bandwidth Awareness
  Recognize and reliably deliver guaranteed network bandwidth to applications
- Service Aggregation
  Achieve aggregation of the best effort and reserved bandwidth available in the network
- TCP - friendliness
  Best effort part of the throughput should conform to fairness criteria
- No Additional Implementation Overhead

GTCP Design Overview

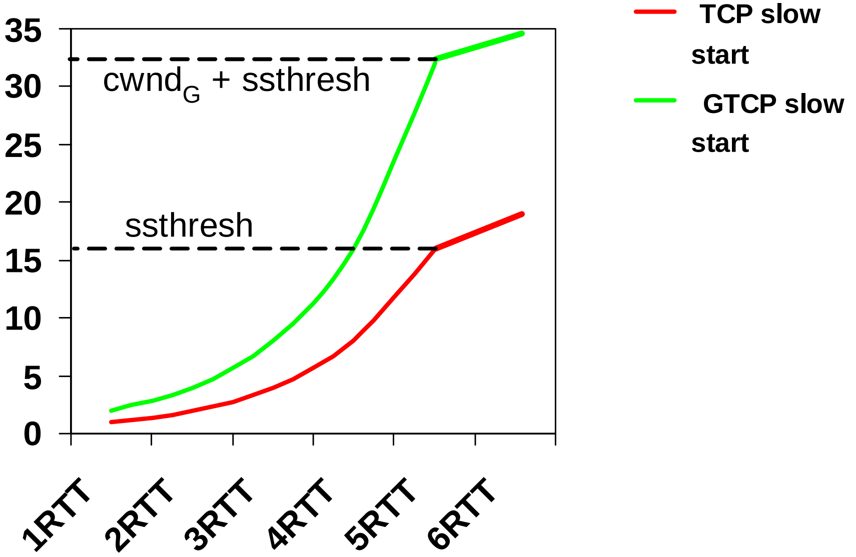
GTCP is a TCP-friendly transport layer protocol that is reserved bandwidth aware, and delivers to applications the effective aggregate of the reserved and best effort bandwidths

- GTCP uses enhanced congestion window adaptation and self-clocking achieved through tailored mechanisms for *RTT estimation, cwnd calculation, start-up behavior, and congestion control*
- GTCP re-uses TCP's mechanisms for *flow-control, reliability, sequencing, and connection management*

GTCP Window Adaptation

- *$cwnd_G$* estimation
  - $cwnd_G = rate_G * rtt_{base}$
  - rtt_{base} : min round trip time recorded in incoming packets
  - $cwnd = cwnd_G + cwnd_{BE}$

- Start-up behavior
  - Initial *$cwnd = 2$*
  - $cwnd > ssthresh + cwnd_G$
    - ☞ exit slow start



Georgia Institute of Technology

GNAN Research Group

GTCP Self Clocking

- Transient Congestion
  - Receive 3 DUPACKs:
    - $cwnd_{update} = cwnd_G + cwnd_{BE}/2$
    - *cwnd* not reduced immediately
  - For the first $cwnd_G$ DUPACKs
    - Forced data transmission
    - $cwnd = cwnd + 1$
  - Ignore later $cwnd_{BE}/2$ (or $cnwd_{BE}/2 – k$ when $k > cwnd_{BE}/2$) DUPACKs
  - Transmit new packets for further DUPACKs, if any
  - Full ACK arrival: $cwnd = cwnd_{update}$
- Severe congestion
  - Timeout recovery has similar design: at least $cwnd_G$ packets are transmitted per RTT during timeout

GTCP Self Clocking Illustration

$$k > cwnd_{BE}/2$$

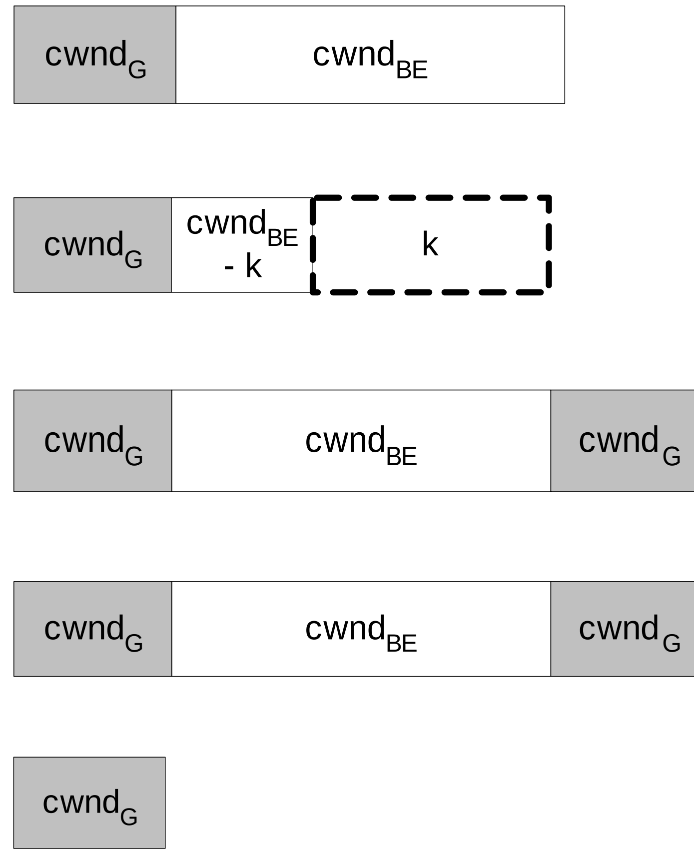
(1) original cwnd size

(2) outstanding packets after k losses

(3) cwnd after the first cwndG DUPACKS

(4) cwnd after the later $cwnd_{BE}$ - k DUPACKS

(5) new packets sent during fast recovery

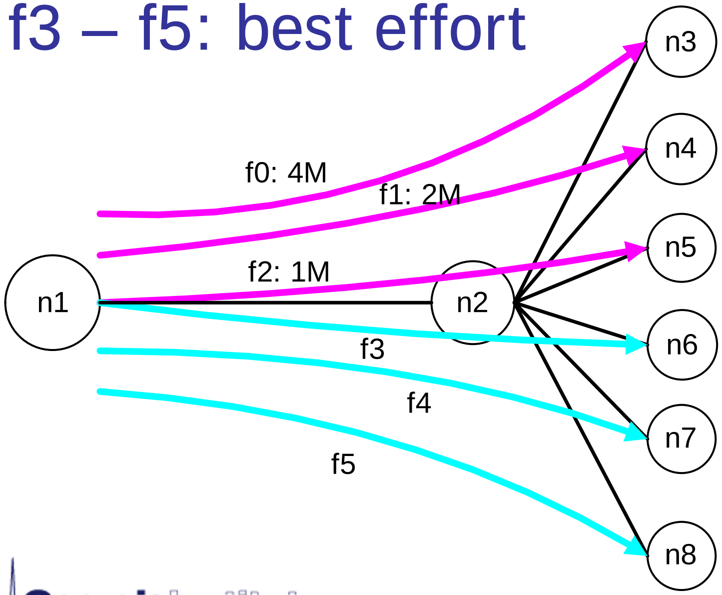


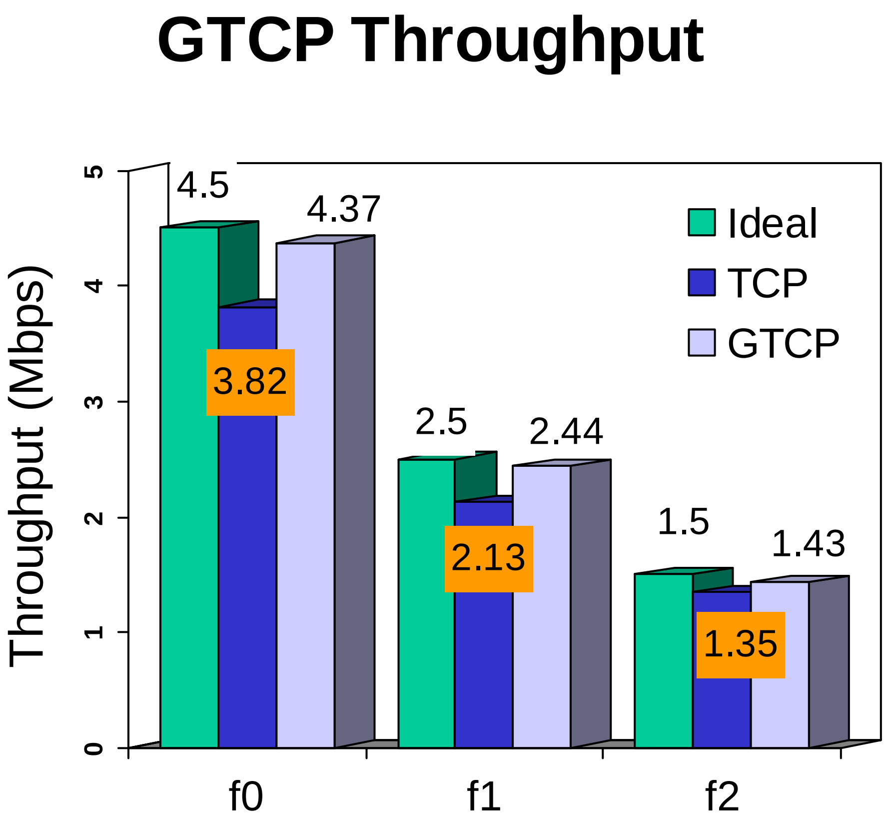
Simulation Results

- Same topology
- 6 flows (f0 – f5)

f0 – f2: 4Mbps, 2Mbps and 1Mbps reservation

f3 – f5: best effort

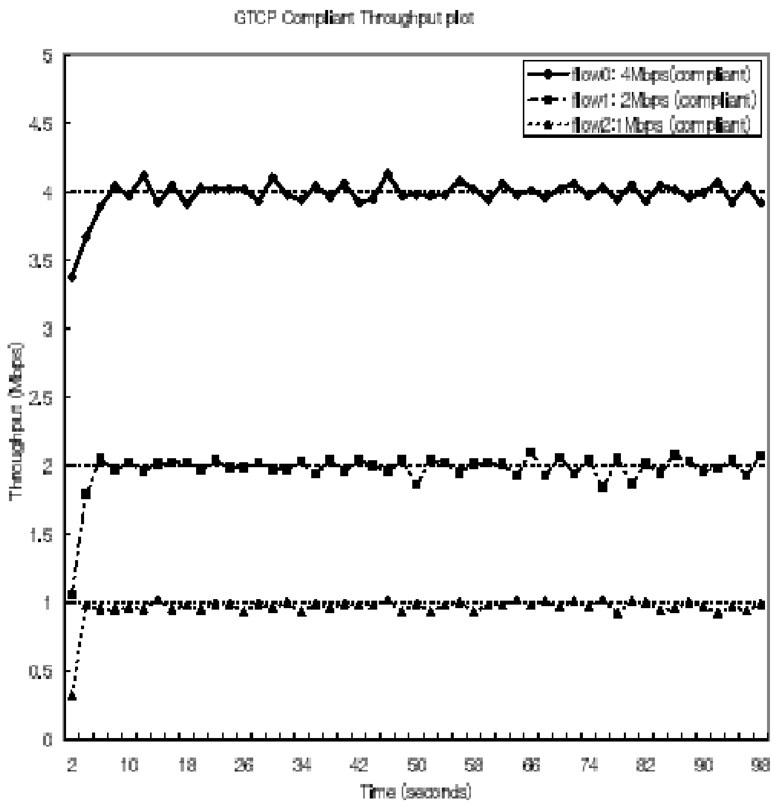


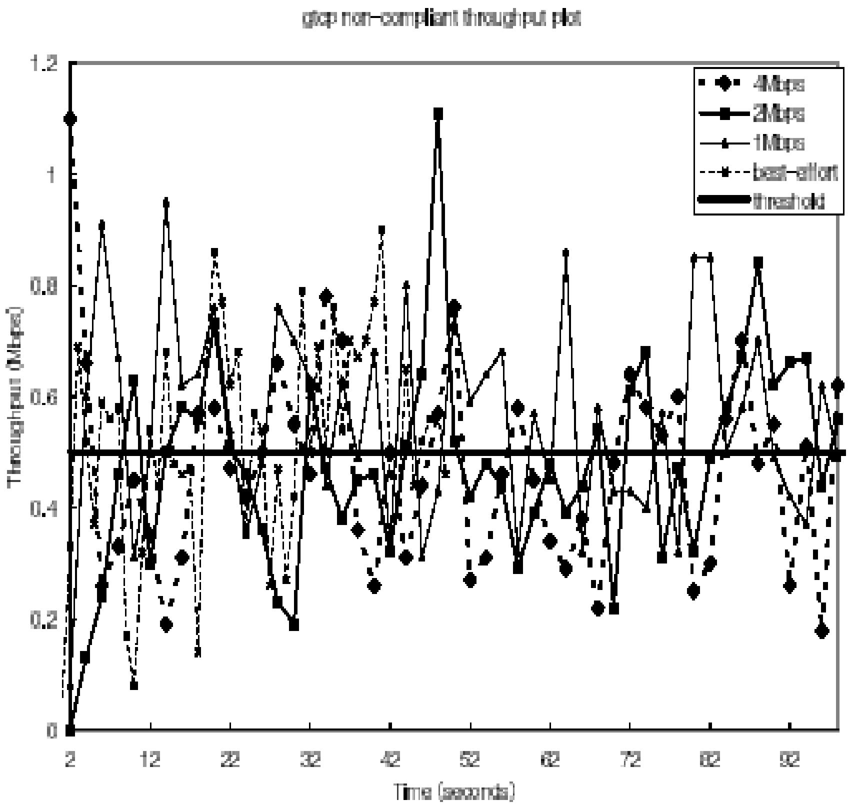


Georgia Institute of Technology

GNAN Research Group

Simulation Results (Contd.)





Other Simulation Results

- Scalability with Link Capacity
- Scalability with Number of Total Flows
- Scalability with Number Flows with reservations
- Scalability with Reserved Bandwidth
- Impact of RTT

- GTCP is able to achieve close to ideal throughput in all the above scenarios while maintain TCP-friendliness

Related Work

- Saha D. Shin et. al.[Transnet'99]:
  - Improve TCP performance with delayed and timed transmission
  - Timer overhead (at granularity of 20ms)
- Lars Wolf et. al.[KiVS'01]:
  - Remove slow start, scale up TCP's flow control window, rate based transmission
  - No performance comparison with TCP
- Ikjun Yeom et. al.[ICMCS'99]:
  - Inverse rate drop mechanism
  - Require network support

Conclusions & Future work

- Default TCP does not perform well in a bandwidth guaranteed environment
- Reasons for TCP's non-ideal performance
- GTCP, an transport protocol for achieving ideal throughput in the target environment
- GTCP's performance verified throug simulations
- GTCP implemented in Linux Kernel
- Future work:
  - Non guaranteed QoS: controlled load, diffserv

Questions & Comments ?

For more information:

http://www.ece.gatech.edu/research/GNAN/