TCP-based background data transfer using inline network measurement

Tomoaki Tsugawa
Osaka University, Japan
t-tugawa@ist.osaka-u.ac.jp
Outline

- Introduction
- Objective
- Related work in background TCP data transfer
- Proposed mechanism
- Performance evaluation
- Conclusion & future work
Introduction

- Prioritized data transfer for "better" Internet services
  - e.g. Contents Delivery Network (CDN)
    - If backup data transfer is set to lower priority, they can be transferred without affecting user-requested data transfer
  - Achieving that by TCP Reno is very difficult
    - TCP Reno cannot avoid affecting the foreground traffic
      - TCP Reno continues to increase its congestion window size until a packet loss occurs
Objective

- Achieve TCP-based background data transfer
  ImTCP-bg (ImTCP background mode)
- Satisfying the following two objectives is important:
  - No adverse effect on foreground traffic
  - Full utilization of the network link bandwidth
- Proposed scheme utilizes results of inline network measurement
Previous studies about TCP-based background data transfer

- Main objective is unaffecting foreground traffic
  - e.g. TCP Nice [9], TCP-LP [10]
    - Achieving lower-prioritized data transfer (rather than TCP Reno) by using RTT as an indication of network congestion
- These protocols cannot efficiently utilize the available bandwidth
  - Degree to which the congestion window size can decrease is fixed and too large
  - No way for obtaining the network bandwidth information

Inline measurement TCP (ImTCP)

- One of inline network measurement techniques
  - Use only data/ACK packets transmitted in TCP
  - Measure available bandwidth of the network path from arrival intervals of ACK packets

- Features
  - Small number of packets used for measurement
  - Continuously and quickly yielding measurement results
  - Only sender TCP modification is enough for measurement
ImTCP’s Problems for background data transfer

- ImTCP does not always provide reliable measurement results for available bandwidth
  - ImTCP cannot measure the available bandwidth when the congestion window size is small
  - Measurement accuracy depends on network environment e.g. RTT, number of active connections, etc

- Background data transfer based on the unreliable result may affect the foreground traffic
ImTCP-bg mechanisms

- Judge whether or not a measurement result is reliable by using the observed RTT value

\[
\frac{RTT_{\text{cur}}}{RTT_{\text{min}}} > \delta
\]

\( \delta \) : threshold \((1 \leq \delta)\)

\( RTT_{\text{cur}}, RTT_{\text{min}} \) : current/minimum RTT value

- Control the congestion window size according to these two mechanisms
  - Bandwidth-based mechanism
  - Enhanced RTT-based mechanism
Bandwidth-based mechanism

Case when the measurement result is reliable

- Control the congestion window size by using the measurement result of available bandwidth
  - Smooth the measurement result
    \[
    \bar{A} \leftarrow (1 - \gamma) \times \bar{A} + \gamma \times A_{\text{cur}}
    \]
    \(\gamma\): smoothing parameter (0 \(\leq\) \(\gamma\) \(\leq\) 1)
    \(A_{\text{cur}}\): the current available bandwidth
  - Determine the upper limit of congestion window size
    \[
    \text{maxcwnd} = \bar{A} \times RTT_{\text{min}}
    \]
    \(RTT_{\text{min}}\): minimum RTT value
- The other congestion controls are the same as TCP Reno
Enhanced RTT-based mechanism

Case when the measurement result is unreliable

- Decrease the congestion window size according to the observed RTT value
  - Determine the value by using the current/minimum RTT
    \[ cwnd \leftarrow cwnd \times \frac{RTT_{min}}{RTT_{cur}} \]
    - \( RTT_{min} \): minimum RTT value
    - \( RTT_{cur} \): current RTT value
  - Preserve the upper limit of congestion window size
Performance evaluation

- Simulation experiments by using ns-2
  - Case of one connection
  - Case of multiple connections

- Performance comparison of ImTCP-bg
  - TCP Reno
  - TCP Nice
  - TCP-LP
Case of one connection

Performance metric:
- Degree of affection to the foreground traffic
- Throughput of background traffic

Router:
- Drop-tail buffer size is 1000 packet

Parameter settings:
- $\gamma = 1/8$, $\delta = 1.2$
Average throughput of the ImTCP-bg connection is the closest to the available bandwidth.

ImTCP-bg has the most ideal characteristics for background data transfer.

For TCP Nice, TCP-LP, and ImTCP-bg, the average download time is almost identical to the case of no background traffic.
Case of multiple connections

Performance metric:
- Degree of affection to the foreground traffic
- Throughput of background traffic
- Fairness among connections

Router:
Drop-tail
buffer size is 1000 packet

Parameter settings:
\( \gamma = 1/8, \ \delta = 1.2 \)

Network environment
Change of throughput
ImTCP-bg can utilize the available bandwidth even when the number of connections is one

Just as is the case of one connection, TCP Reno shows the worst behavior for the background data transfer, in terms of over-utilization of the available bandwidth

TCP Nice cannot utilize the available bandwidth effectively especially when the number of connections is small

For TCP-LP when a new connection joins, packet losses occur and the throughput becomes low
Conclusion & Future work

Conclusion
- We introduced a new background TCP data transfer
  - It uses an inline network measurement technique
- We investigated the effectiveness of ImTCP-bg through simulation experiments
  - No bad effect on foreground traffic
  - Full utilization of the network available bandwidth

Future work
- Consideration about parameter settings
- Performance evaluation in an actual network
Thank you for your attention