Application-Level Congestion Control (ALCC)

Yasir Zaki*, Thomas Pötsch*, Talal Ahmad+, Tai Liu*, Arjuna Sathiaseelan#, Jay Chen*, Lakshmi Subramanian+

NYU Abu Dhabi*, Gaius Networks#, New York University+

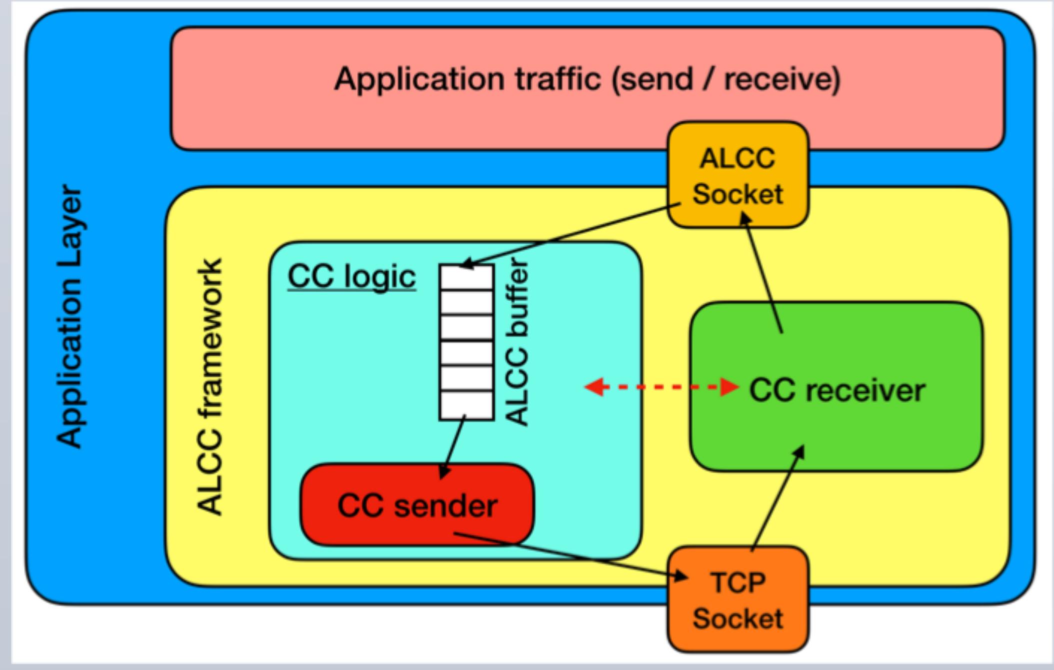
Introduction

Despite the significant performance gains of recent congestion control protocols, TCP remains the de-facto standard, due to:

- Most mobile and middlebox applications adopt the standard TCP stack and there are difficult to change
- The tight coupling between the TLS/SSL layer and the underlying TCP layer

Application-level Congestion Control

- Perform congestion control at the application layer
- Replace TCP's congestion window with the congestion window computed at the application
- Constrains the application data flow to the transport layer based on this computed window

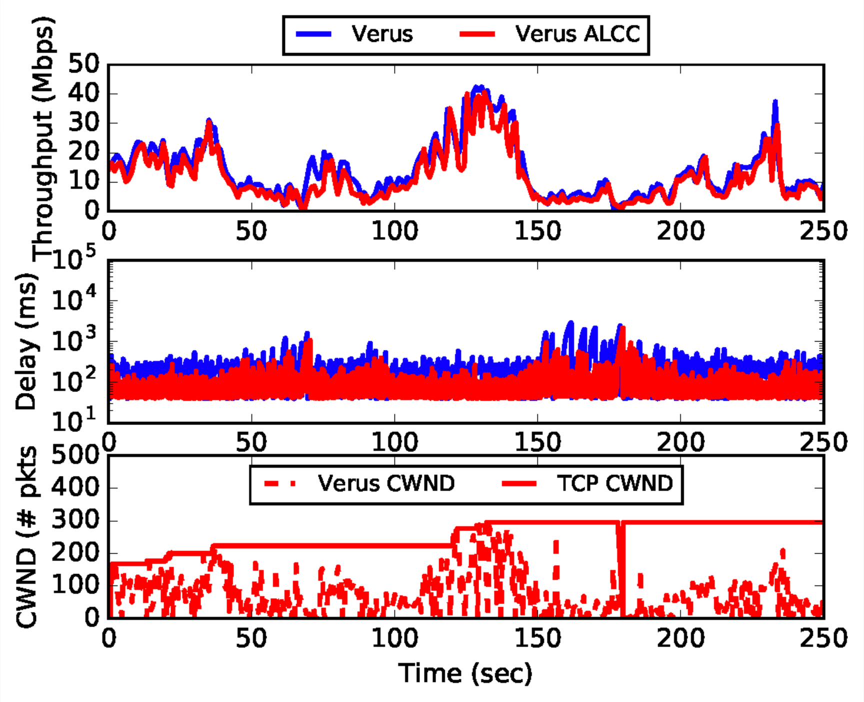


Implementation

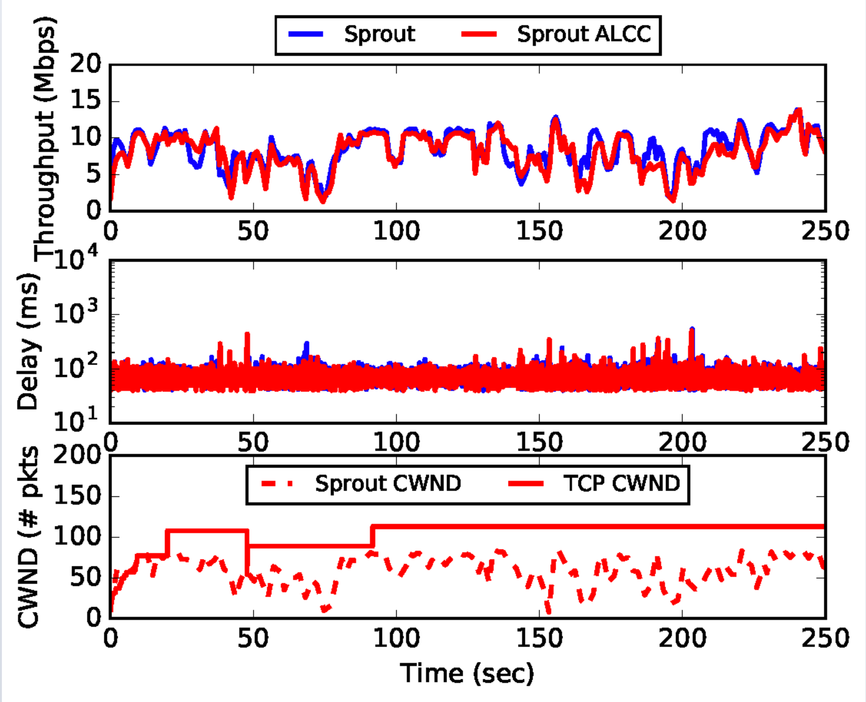
- Implemented ALCC as a C++ application library:
  - wrapper around the default BSD TCP library
  - expose the same socket APIs
- ALCC provides:
  - place-holder functions to easily integrate any congestion control implementation

ALCC Results

- ALCC achieve nearly identical throughput and packet delays to native protocol
- Both ALCC implementations of Sprout and Verus maintain a smaller congestion window compared to the underlying TCP window
- The packet delay is significantly reduced and the throughput performance is preserved



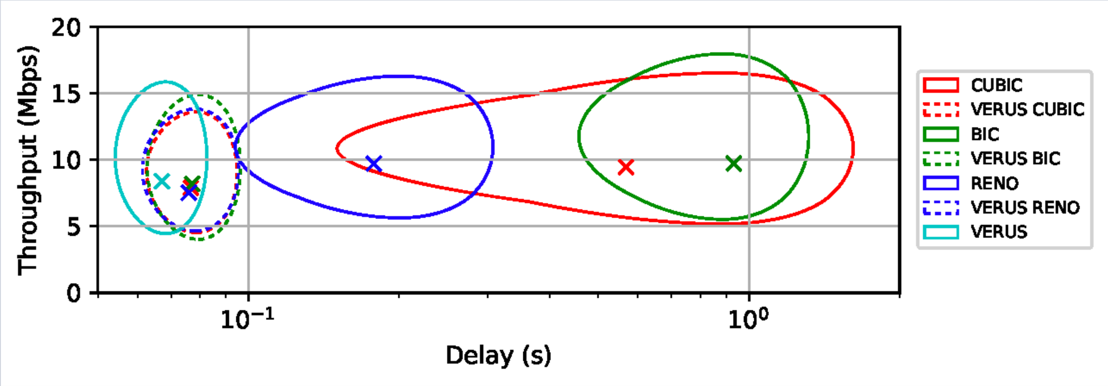
Native Verus vs. Verus ALCC throughput, packet delay, and congestion window over time



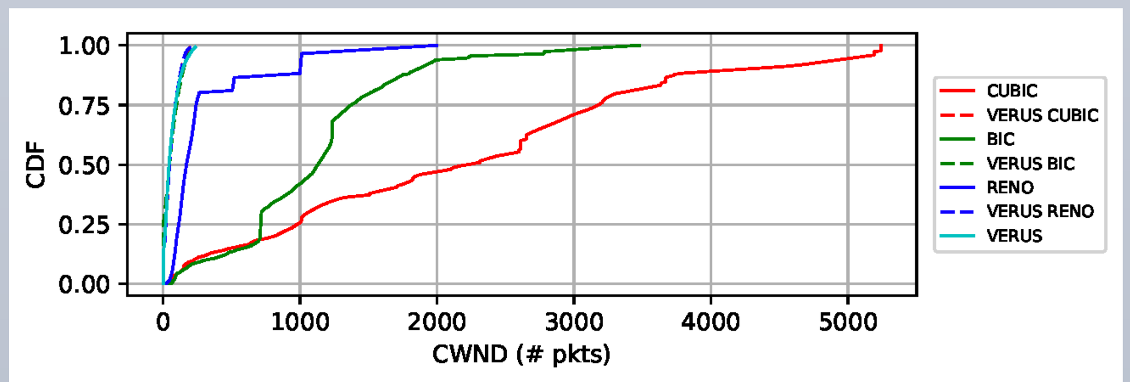
Native Sprput vs. Sprout ALCC throughput, packet delay, and congestion window over time

ALCC also works over other TCP flavors such as TCP Reno and TCP BIC.

- All three TCP flavors (solid lines) suffer from large packet delays
- Verus exhibits the lowest packet delay while still achieving similar throughput to all TCP flavors
- Verus ALCC achieves a very similar performance to native Verus
- Native Verus and Verus ALCC show very similar CDFs (< 250 packets)
- All three TCP flavors exhibit fairly large CWNDs, ranging up to 5000 packets in the case of TCP Cubic



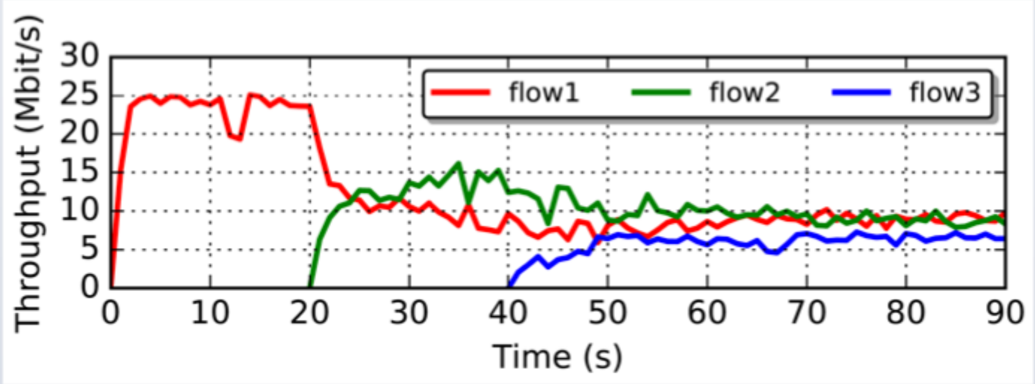
Throughput vs. packet delay



Congestion window cumulative distribution function

ALCC Fairness

- Verus ALCC shows similar fairness characteristics to the native protocol implementation
- We made similar observation for the Sprout ALCC implementation



Conclusion

- Legacy congestion control protocols like TCP do not work efficiently on cellular
- ALCC facilitates the deployment of new congestion control protocols without modifying the native TCP stack
- ALCC tightly controls the underlying congestion window of TCP
- ALCC achieves comparable performance to the native protocols.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0048. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

جامعة نيويورك ابوظبي NYU ABU DHABI

NEW YORK UNIVERSITY