Stateless TCP
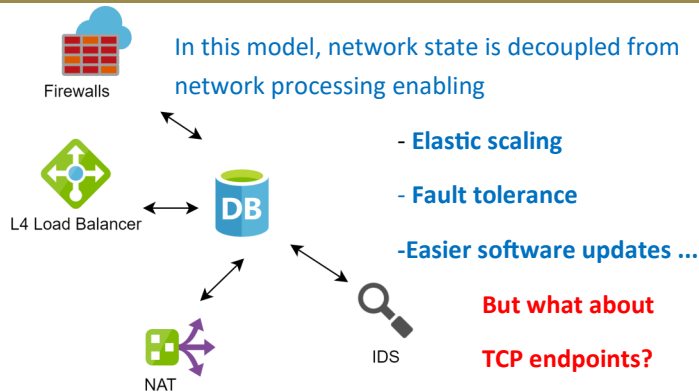
Marcelo Abranches, Eric Keller

University of Colorado, Boulder
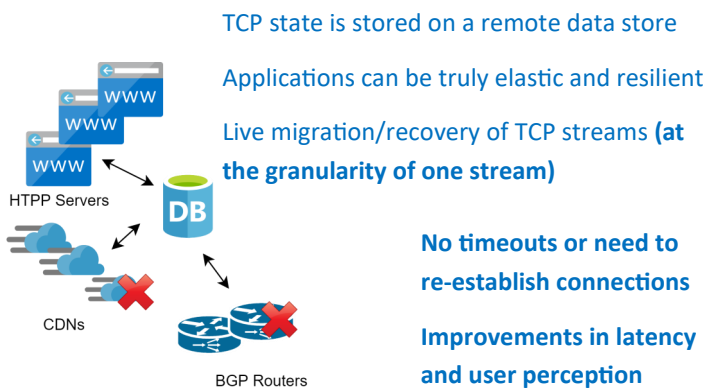
University of Colorado **Boulder**

Stateless Networking


In this model, network state is decoupled from network processing enabling

- **Elastic scaling**

- **Fault tolerance**

-**Easier software updates ...**

But what about

TCP endpoints?

Stateless TCP

TCP state is stored on a remote data store

Applications can be truly elastic and resilient

Live migration/recovery of TCP streams **(at the granularity of one stream)**

No timeouts or need to re-establish connections

Improvements in latency and user perception
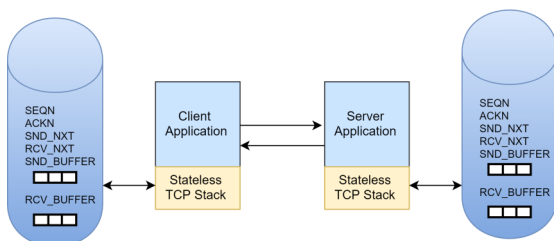
Use cases

CDNs, HTTP servers, BGP routers

Load Balancing, live malware recovery, offloaded firewall/IDS and more
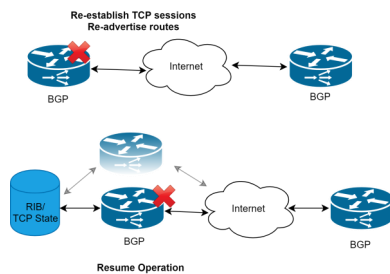
Challenges

New state to deal with (e.g., **TCP send and receive buffers**)

Tighter coupling to the end application

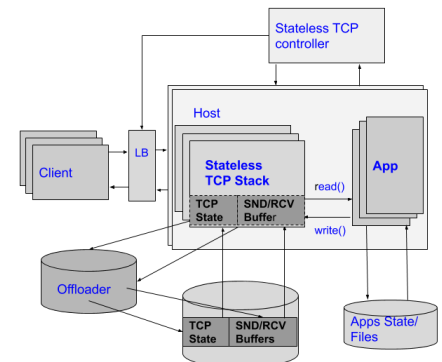## TCP buffer recovery

Designed in two levels

The **first level** allows **correct L7 protocol operation** during recovery and scaling events

The **second level** allows applications to interact with the remote data buffers

Architecture

Each server runs a series of **Stateless TCP** stacks

Clients access their services through a load balancer or through an SDN switch

Prototype and Future work

Prototype is being built on the top of mTCP and DPDK

Redis as the remote datastore

Evaluating using HTTP server and Apache Bench as the benchmark

Early results show that Stateless TCP reduces throughput by less than 20% comparing with vanilla mTCP
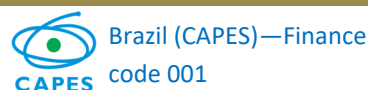
Finish the prototype

What use cases and functionalities should we consider as future work?

Contacts

marcelo.deabranches@colorado.edu

eric.keller@colorado.edu

Acknowledgements