Improving TCP's performance using programmable networks

Belma Turkovic and Fernando Kuipers

RoN++ Meeting, December 16th, 2020

TUDelft

TCP complexity is increasing!

- New protocols and congestion control algorithms are continuously being developed

Extensions

- Deploying TCP extensions is difficult
  - Can take more than a decade (e.g., timestamp, SACK)

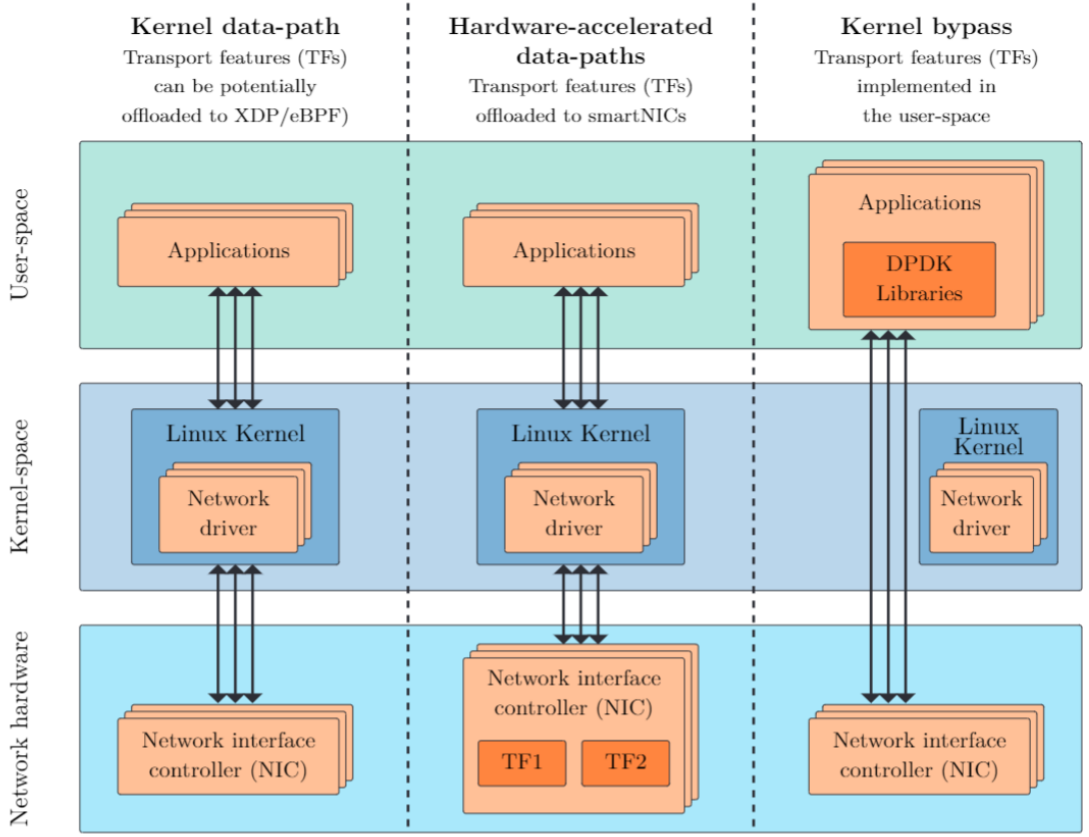
Extensions

- Deploying TCP extensions is difficult
  - Can take more than a decade (e.g., timestamp, SACK)
- Only options:
  - Socket options
  - Modules

Goal

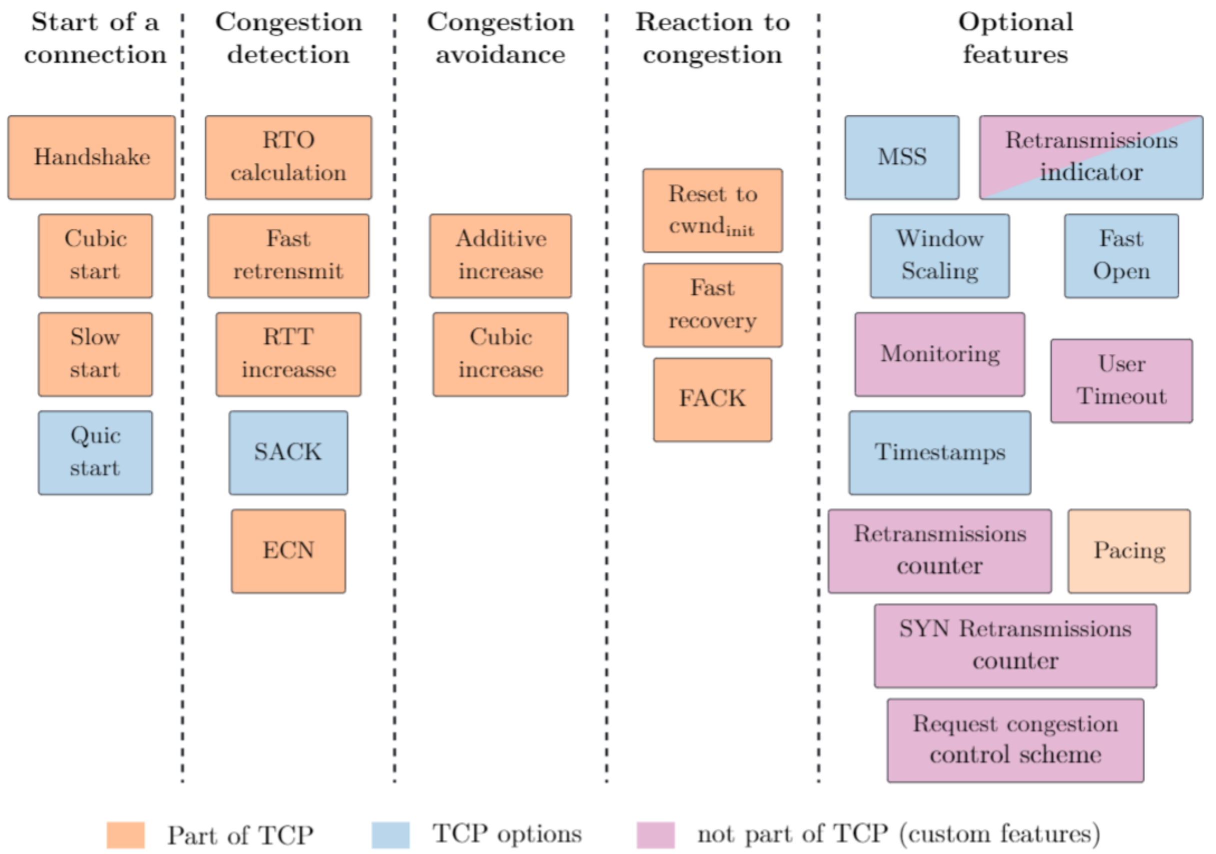
- Investigate how different data-plane techniques can be used to introduce programmability in the end-hosts

Data-plane techniques



Kernel data-path
Transport features (TFs) can be potentially offloaded to XDP/eBPF)
Hardware-accelerated data-paths
Transport features (TFs) offloaded to smartNICs
Kernel bypass
Transport features (TFs) implemented in the user-space
User-space
Kernel-space
Network hardware
Applications
Applications
Applications
DPDK Libraries
Linux Kernel
Network driver
Linux Kernel
Network driver
Linux Kernel
Network driver
Network interface controller (NIC)
Network interface controller (NIC)
TF1
TF2
Network interface controller (NIC)

Transport features



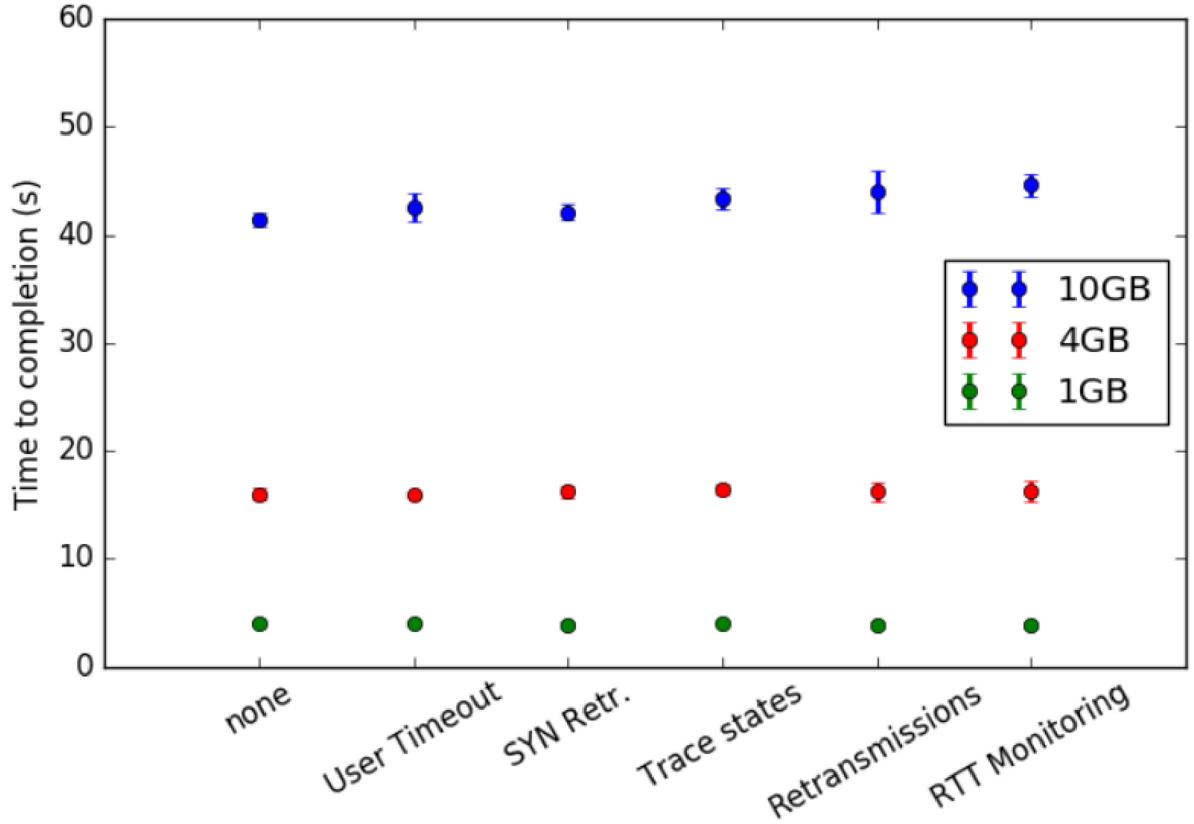
eBPF

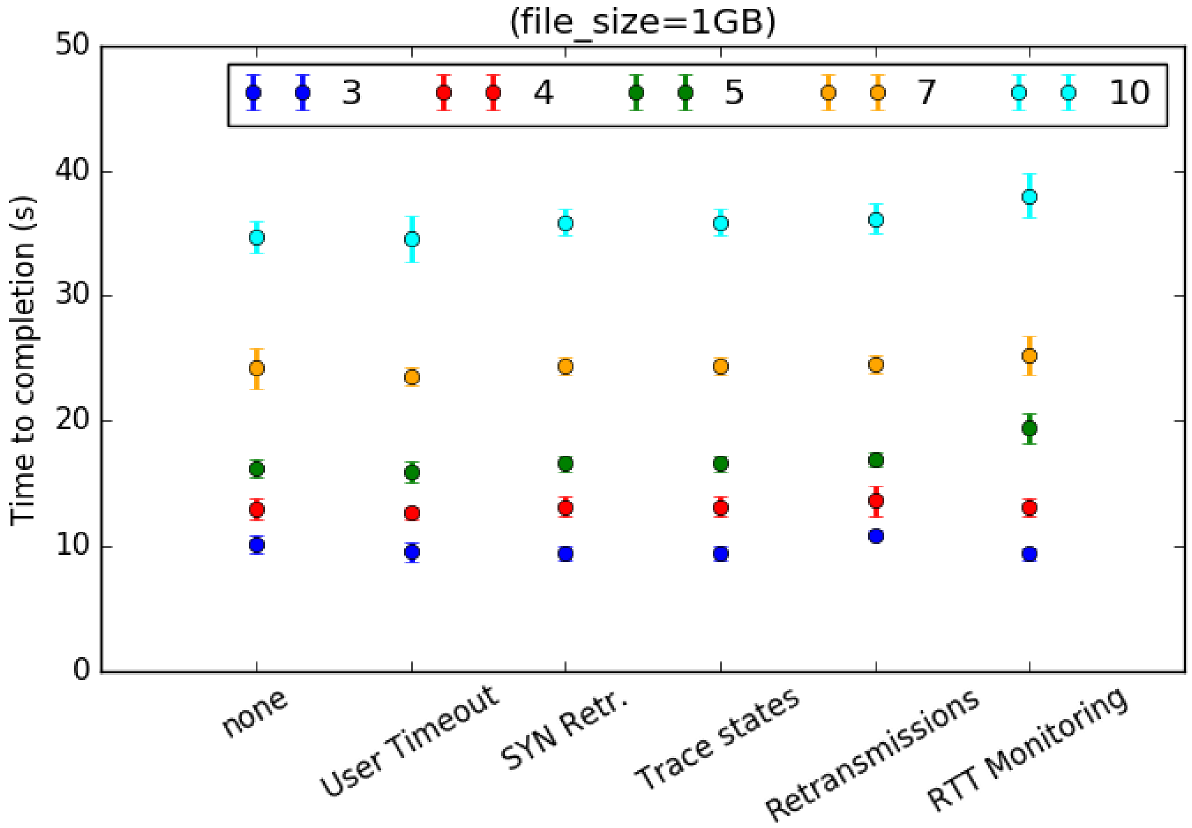
- Lightweight VM
- Enables running sandboxed programs in the Linux kernel
- No need to change kernel source-code or load kernel modules

Source: https://ebpf.io/

eBPF evolution

- BPF
- eBPF
- Exposure to user-space
- TCP-BPF
- Support for user-defined TCP options


Time to completion (s)
0
10
20
30
40
50
60
none
User Timeout
SYN Retr.
Trace states
Retransmissions
RTT Monitoring
10GB
4GB
1GB


(file_size=1GB)
Time to completion (s)
0
10
20
30
40
50
3
4
5
7
10
none
User Timeout
SYN Retr.
Trace states
Retransmissions
RTT Monitoring

👍

- ⊕ Easy to implement
- ⊕ Small overhead

👎

- ⊖ Limited program size
- ⊖ Limited hooks
- ⊖ Uses the “slow” kernel
- ⊖ Needs a custom kernel
  (for custom TCP options, TCP-BPF supported since 4.15)

DPDK

- Kernel bypass technique
- Set of libraries to accelerate packet processing

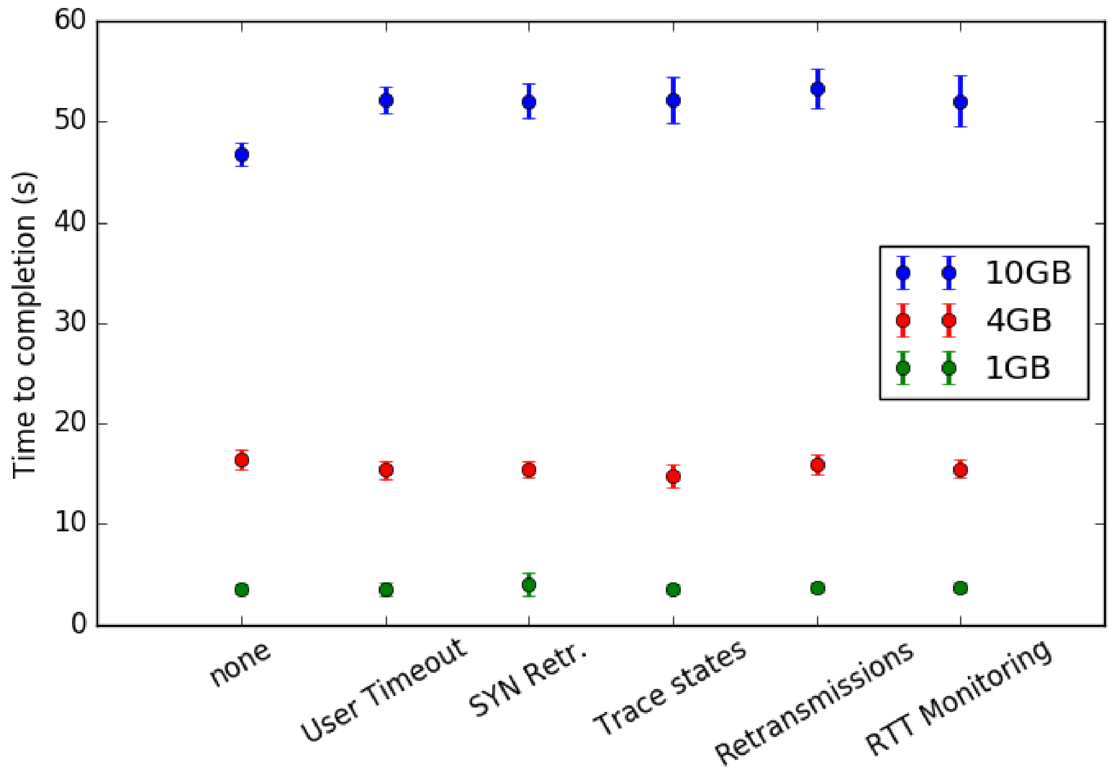
Source: https://medium.com/@lhamthomas45/dpdk-19-11-is-out-why-you-should-update-and-how-to-do-so-7395810f71e

DPDK

- Tested two different frameworks:
  - F-stack
  - TLDK

DPDK

- Tested two different frameworks:
  - F-stack
  - TLDK



Time to completion (s)
0
10
20
30
40
50
60
none
User Timeout
SYN Retr.
Trace states
Retransmissions
RTT Monitoring
10GB
4GB
1GB

DPDK
DATA PLANE DEVELOPMENT KIT

+ Fast performance

− Does not have TCP/IP stack out of the box

− Debugging is difficult
(Few tools)

− Specialized hardware

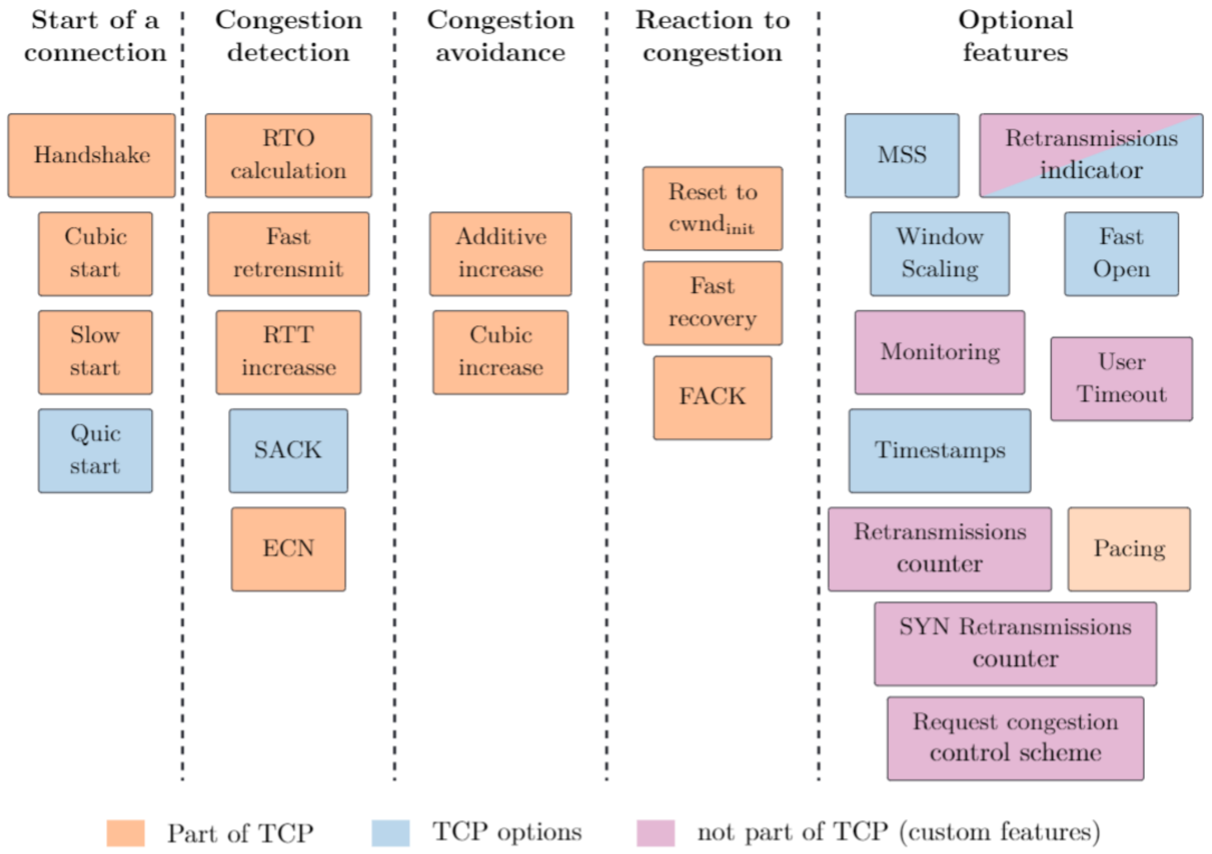
− Steep learning curve

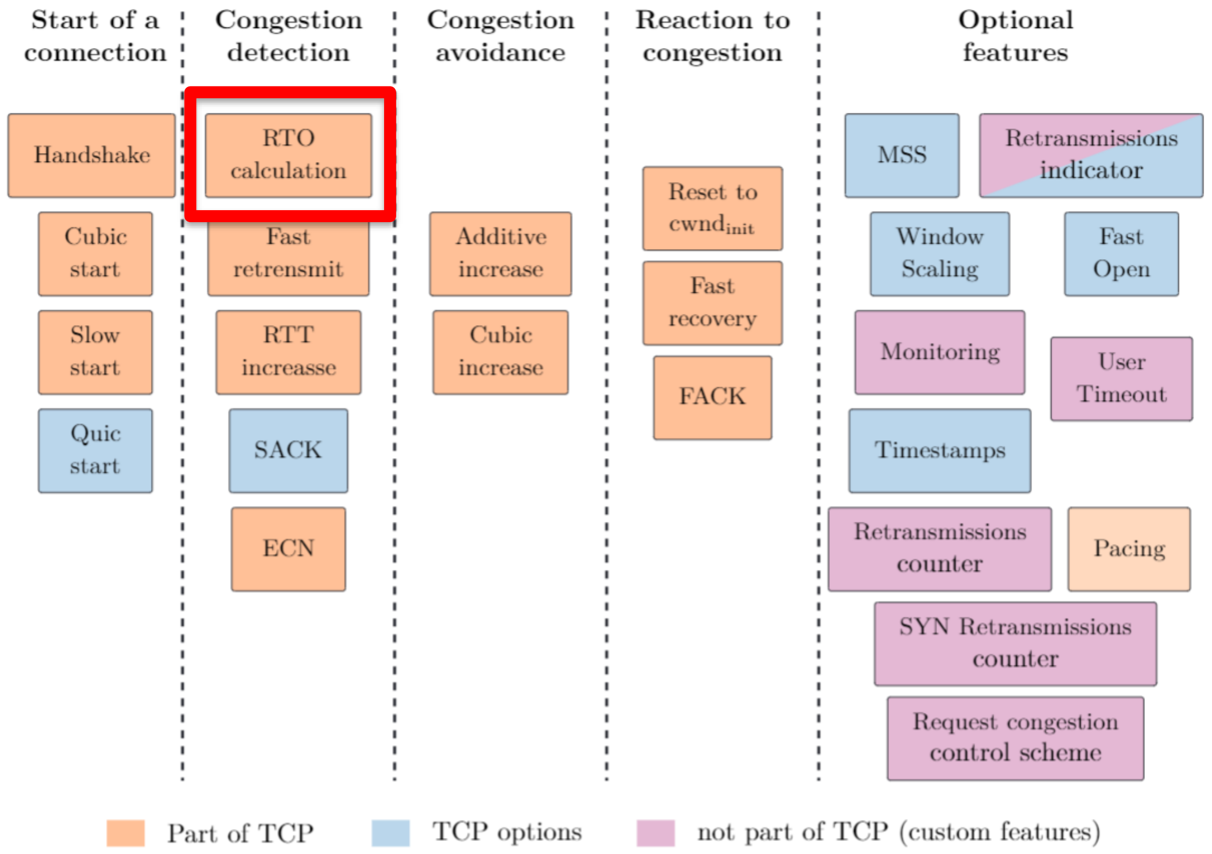
TU Delft

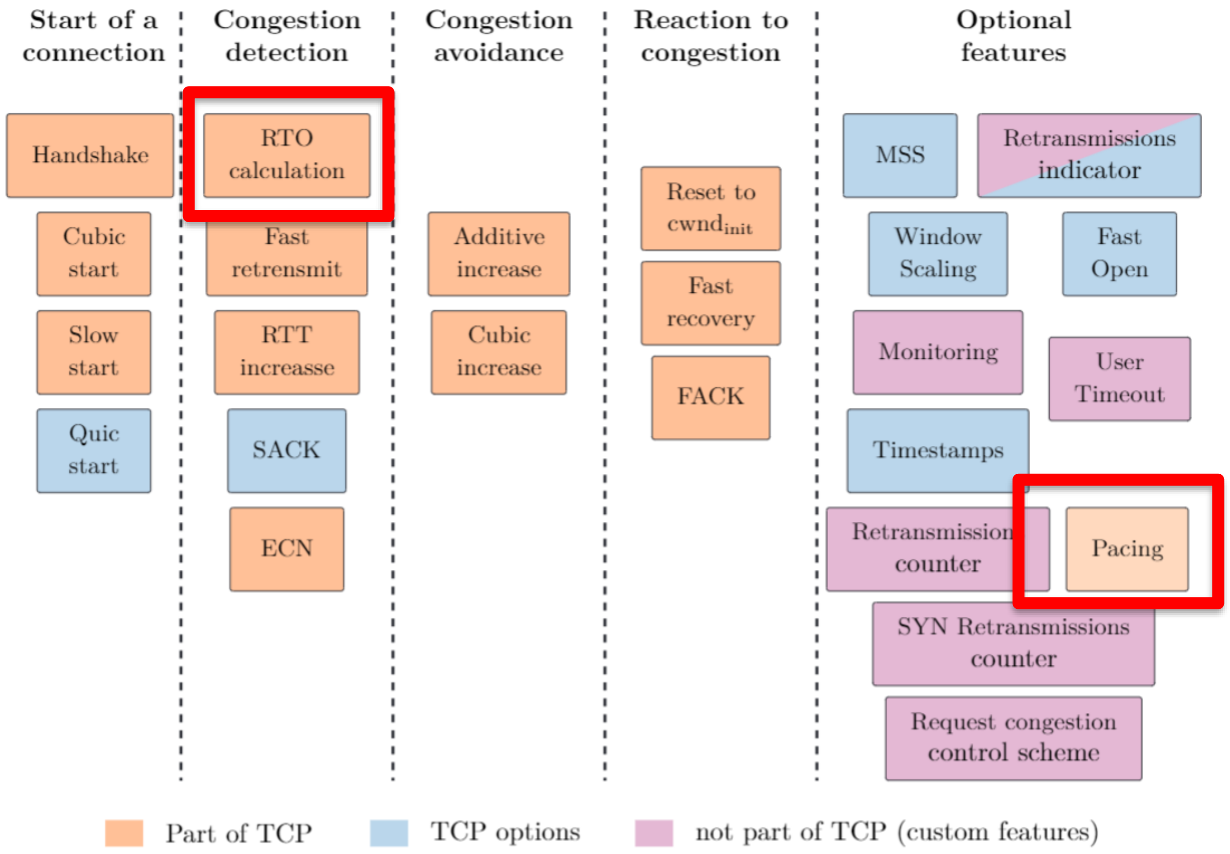
P4

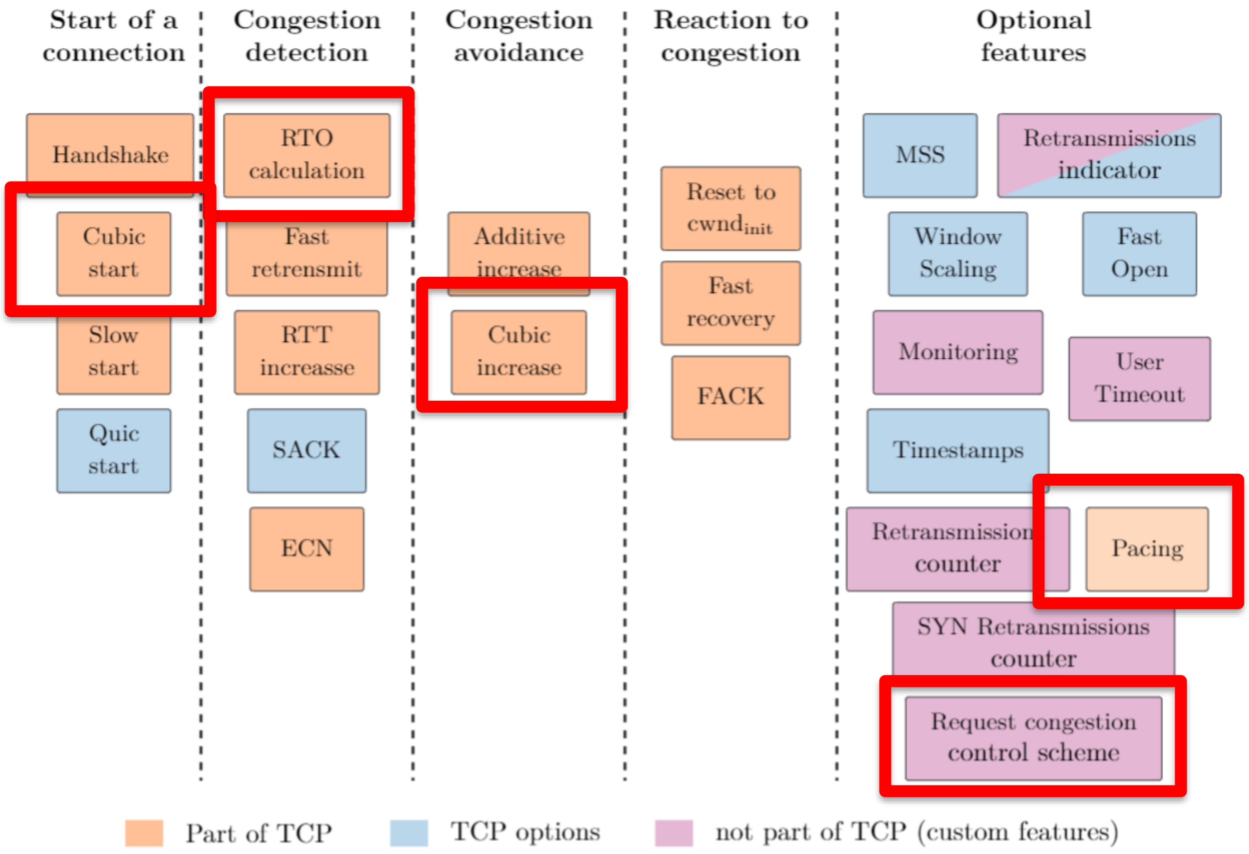
- High-level language
- Defines how packet processors are configured and operated on network packets

Source: https://plvision.eu/wp-content/uploads/2018/10/p4.jpg


Start of a connection
Congestion detection
Congestion avoidance
Reaction to congestion
Optional features
Handshake
Cubic start
Slow start
Quic start
RTO calculation
Fast retrensmit
RTT increasse
SACK
ECN
Additive increase
Cubic increase
Reset to cwnd$_{init}$
Fast recovery
FACK
MSS
Retransmissions indicator
Window Scaling
Fast Open
Monitoring
User Timeout
Timestamps
Retransmissions counter
Pacing
SYN Retransmissions counter
Request congestion control scheme
Part of TCP
TCP options
not part of TCP (custom features)


Start of a connection
Handshake
Cubic start
Slow start
Quic start
Congestion detection
RTO calculation
Fast retrensmit
RTT increasse
SACK
ECN
Congestion avoidance
Additive increase
Cubic increase
Reaction to congestion
Reset to cwnd$_{init}$
Fast recovery
FACK
Optional features
MSS
Retransmissions indicator
Window Scaling
Fast Open
Monitoring
User Timeout
Timestamps
Retransmissions counter
Pacing
SYN Retransmissions counter
Request congestion control scheme
Part of TCP
TCP options
not part of TCP (custom features)


Start of a connection
Congestion detection
Congestion avoidance
Reaction to congestion
Optional features
Handshake
Cubic start
Slow start
Quic start
RTO calculation
Fast retrensmit
RTT increasse
SACK
ECN
Additive increase
Cubic increase
Reset to cwnd$_{init}$
Fast recovery
FACK
MSS
Retransmissions indicator
Window Scaling
Fast Open
Monitoring
User Timeout
Timestamps
Retransmission counter
Pacing
SYN Retransmissions counter
Request congestion control scheme
Part of TCP
TCP options
not part of TCP (custom features)


Start of a connection
Congestion detection
Congestion avoidance
Reaction to congestion
Optional features
Handshake
Cubic start
Slow start
Quic start
RTO calculation
Fast retrensmit
RTT increasse
SACK
ECN
Additive increase
Cubic increase
Reset to $\text{cwnd}_{\text{init}}$
Fast recovery
FACK
MSS
Retransmissions indicator
Window Scaling
Fast Open
Monitoring
User Timeout
Timestamps
Retransmission counter
Pacing
SYN Retransmissions counter
Request congestion control scheme
Part of TCP
TCP options
not part of TCP (custom features)

- Easy to implement
- Fast performance
- Can be combined with the kernel stack
  (Only offload TCP options)

- Limited functionality
  (No floating-point arithmetic, limited accesses to registers)
- Specialized hardware
- Can have high memory utilization
- Can only react to packets

Source: p4.org

Conclusion

- Three techniques to improve the flexibility of the TCP stack
  - Small overhead
  - Increased flexibility

P4air

TCP complexity is increasing!

- New protocols and congestion control algorithms are continuously being developed

→ It is impossible to take their interactions with other protocols and algorithms into account

Goal

Improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

Goal

Improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

- From within the data-plane

Goal

Improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

- From within the data-plane

  → and by taking into account limitations on actions and/or memory accesses

TU Delft

P4air

- P4air: Increasing Fairness among Competing Congestion Control Algorithms

- Conference: IEEE ICNP 2020
- YouTube video: https://youtu.be/udXrPi6GVtk

TU Delft