

# Extending SDN to the Data Plane

Anirudh Sivaraman, Keith Winstein, Suvinay Subramanian,  
Hari Balakrishnan

M.I.T.

<http://web.mit.edu/anirudh/www/sdn-data-plane.html>

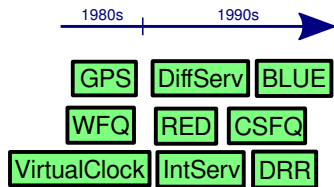
Two key decisions on a per-packet basis:

- ▶ **Scheduling:** Which packet to transmit next?
- ▶ **Queue Management:** How long can queues grow? Which packet to drop?

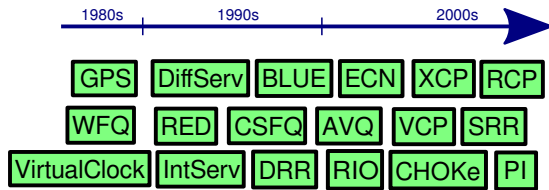
# The long lineage of in-network algorithms



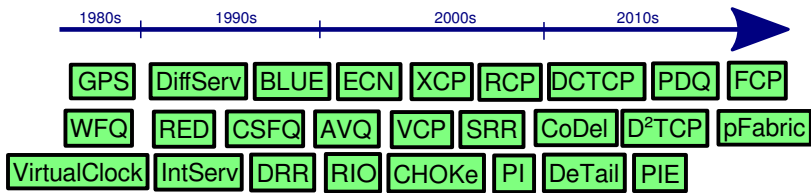
# The long lineage of in-network algorithms



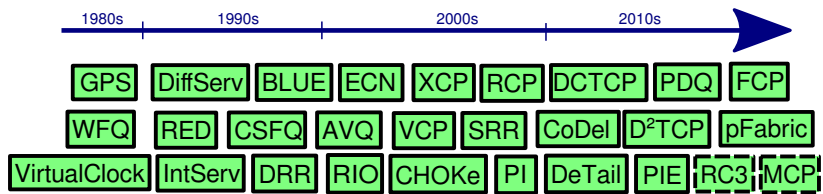
# The long lineage of in-network algorithms



# The long lineage of in-network algorithms



# The long lineage of in-network algorithms



# The Data Plane is continuously evolving

- ▶ Each scheme wins in its own evaluation.
- ▶ Quest for a “silver bullet” in-network method.



## We disagree: There is no silver bullet!

- ▶ Different applications care about different objectives.
- ▶ Applications use different transport protocols.
- ▶ Networks are heterogeneous.

## Our work:

- ▶ Quantify non-universality of in-network methods.
- ▶ Extend SDN to the Data Plane to handle in-network diversity.

## Quantifying “No Silver Bullet”: Network Configurations

<u>Configuration</u>	<u>Description</u>
<b>CoDel+FCFS</b>	One shared FCFS queue with CoDel
<b>CoDel+FQ</b>	Per-flow fair queueing with CoDel on each queue (Nichols 2013)
<b>Bufferbloat+FQ</b>	Per-flow fair queueing with deep buffers on each queue

## Quantifying “No Silver Bullet”: Workloads and Objectives

<u>Workload</u>	<u>Description</u>	<u>Objective</u>
<b>Bulk</b>	Long-running bulk transfer flow	Max. throughput
<b>Web</b>	Switched flow with ON/OFF periods	Min. 99.9 %ile flow completion time
<b>Interactive</b>	Long-running interactive flow	Max. $\frac{\text{throughput}}{\text{delay}}$

# Quantifying “No Silver Bullet”

CoDel+FCFS

CoDel+FQ

Bufferbloat+FQ

# Quantifying “No Silver Bullet”

CoDel+FCFS

CoDel+FQ

Bufferbloat+FQ

# Quantifying “No Silver Bullet”

CoDel+FCFS

Experiment configuration:  
Workload: 1 Bulk flow + 1 Web Flow  
Network: LTE link with 150 ms min. RTT

CoDel+FQ

Bufferbloat+FQ

# Quantifying “No Silver Bullet”

CoDel+FCFS

Experiment configuration:  
Workload: 1 Bulk flow + 1 Web Flow  
Network: LTE link with 150 ms min. RTT

Bulk Tpt: 3.9 Mbps

CoDel+FQ

Bufferbloat+FQ

Web Tail FCT: 43 s



# Quantifying “No Silver Bullet”

CoDel+FCFS

Experiment configuration:  
Workload: 1 Bulk flow + 1 Web Flow  
Network: LTE link with 150 ms min. RTT

Bulk Tpt: 3.9 Mbps

CoDel+FQ

Web Tail FCT: 43 s

Bulk Tpt: 11.2 Mbps

Bufferbloat+FQ

Web Tail FCT: 21 s

# Quantifying “No Silver Bullet”

CoDel+FCFS

Experiment configuration:  
Workload: 1 Bulk flow + 1 Web Flow  
Network: LTE link with 150 ms min. RTT

Bulk Tpt: 3.9 Mbps

CoDel+FQ

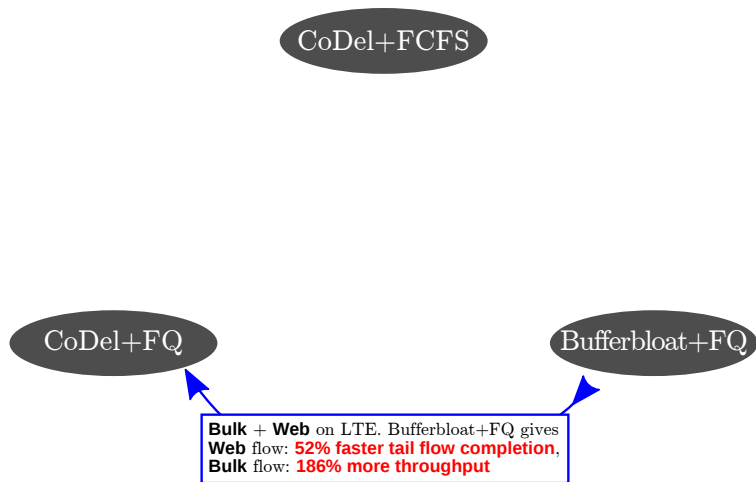
Web Tail FCT: 43 s

Bulk Tpt: 11.2 Mbps

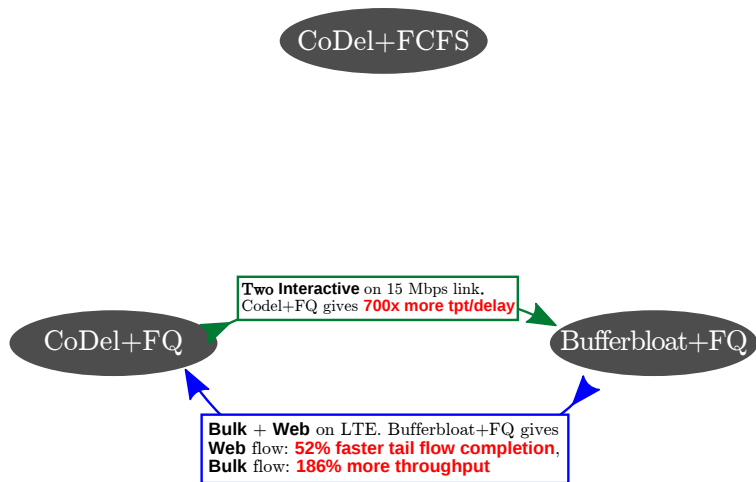
Bufferbloat+FQ

Web Tail FCT: 21 s

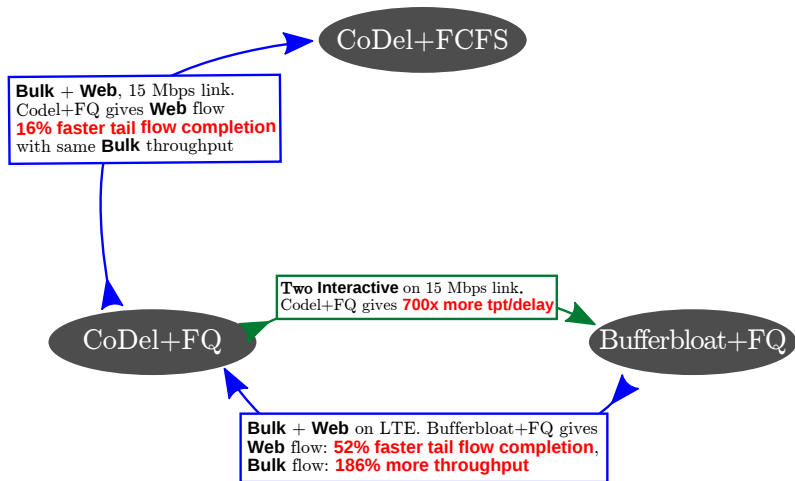
# Quantifying “No Silver Bullet”



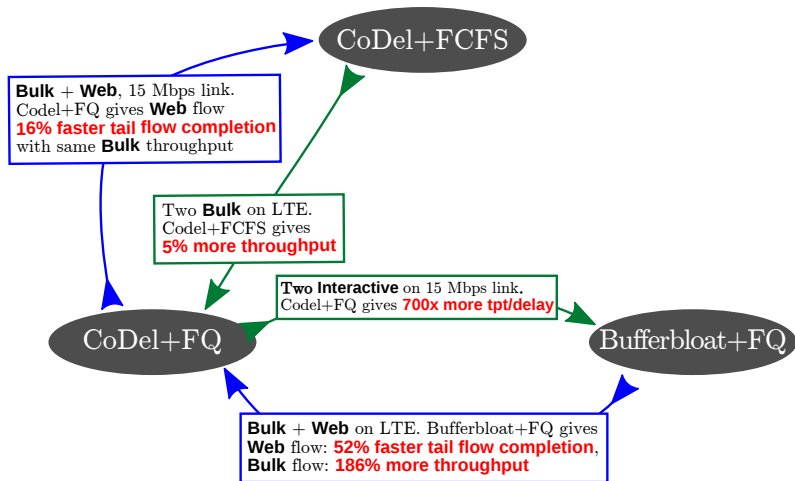
# Quantifying “No Silver Bullet”



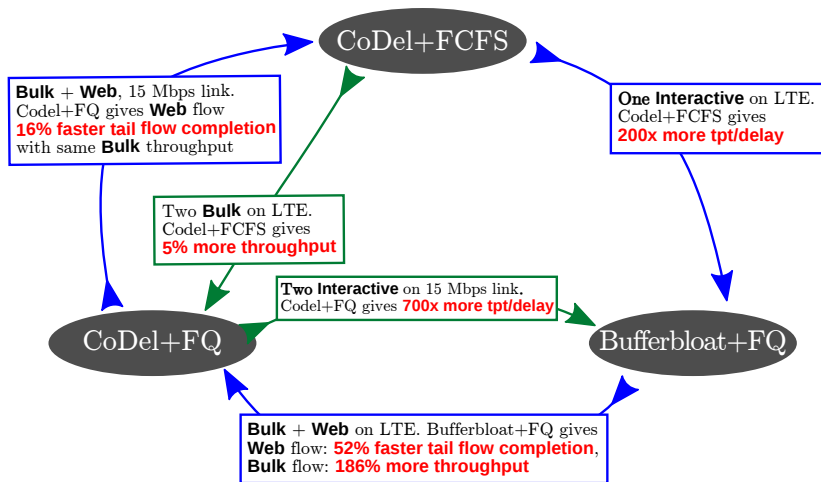
# Quantifying “No Silver Bullet”



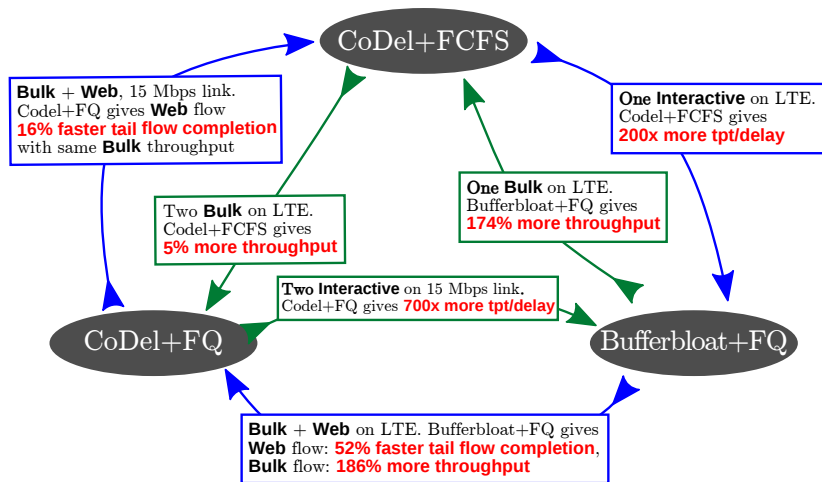
# Quantifying “No Silver Bullet”



# Quantifying “No Silver Bullet”



# Quantifying “No Silver Bullet”





## Why is no single data plane configuration the best?

- ▶ Bufferbloat gives the best throughput on variable-rate links.
- ▶ FCFS is preferable to Fair Queuing with homogenous objectives.
- ▶ Fair Queuing is preferable with heterogeneous objectives.

## So what should the network designer do?

- ▶ Don't strive for the best in-network behaviour.
- ▶ Instead, architect for evolvability.
- ▶ Conceptually, extend SDN to include the data plane as well.

## Flexibility without sacrificing performance

- ▶ Provide interfaces only to the head and tail of queues
- ▶ Operators specify only queue-management/scheduling logic
- ▶ No access to packet payloads.

# Building such a data plane in four parts

## ▶ Hardware gadgets

- ▶ Random number generators (RED, BLUE)
- ▶ Binary tree of comparators (pFabric, SRPT)
- ▶ Look-up tables for function approximation (CoDel, RED)

# Building such a data plane in four parts

- ▶ Hardware gadgets
  - ▶ Random number generators (RED, BLUE)
  - ▶ Binary tree of comparators (pFabric, SRPT)
  - ▶ Look-up tables for function approximation (CoDel, RED)
- ▶ I/O interfaces
  - ▶ Drop/mark head/tail of queue
  - ▶ Interrupts for enqueue/dequeue
  - ▶ Rewrite packet fields

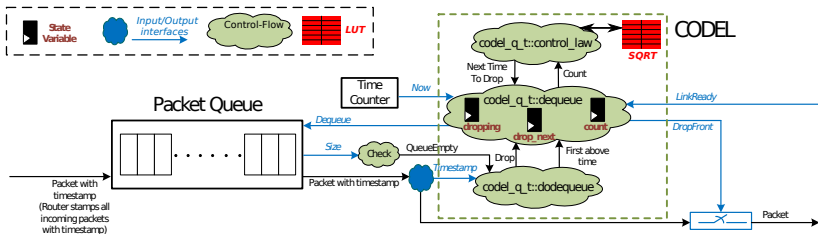
# Building such a data plane in four parts

- ▶ Hardware gadgets
  - ▶ Random number generators (RED, BLUE)
  - ▶ Binary tree of comparators (pFabric, SRPT)
  - ▶ Look-up tables for function approximation (CoDel, RED)
- ▶ I/O interfaces
  - ▶ Drop/mark head/tail of queue
  - ▶ Interrupts for enqueue/dequeue
  - ▶ Rewrite packet fields
- ▶ State maintenance
  - ▶ Per-flow (WFQ, DRR)
  - ▶ Per-dst address (PF)

# Building such a data plane in four parts

- ▶ Hardware gadgets
  - ▶ Random number generators (RED, BLUE)
  - ▶ Binary tree of comparators (pFabric, SRPT)
  - ▶ Look-up tables for function approximation (CoDel, RED)
- ▶ I/O interfaces
  - ▶ Drop/mark head/tail of queue
  - ▶ Interrupts for enqueue/dequeue
  - ▶ Rewrite packet fields
- ▶ State maintenance
  - ▶ Per-flow (WFQ, DRR)
  - ▶ Per-dst address (PF)
- ▶ A domain-specific instruction set
  - ▶ Expresses control flow
  - ▶ Implements new functions unavailable in hardware

# Feasibility study: CoDel





## Synthesis numbers on the Xilinx Kintex-7

Resource	Usage	Fraction
Slice logic	1,256	1%
Slice logic dist.	1,975	2%
IO/GTX ports	27	2%
DSP slices	0	0%
Maximum speed	12.9 million pkts/s ~10 Gbps	

- ▶ Small fraction of the FPGA's resources.
- ▶ Can be improved by pipelining or parallelizing.

# Conclusion

- ▶ No silver bullet to in-network resource allocation.
- ▶ Algorithms will evolve: Data Plane should help
- ▶ Reproduce our results:  
<http://web.mit.edu/anirudh/www/sdn-data-plane.html>