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

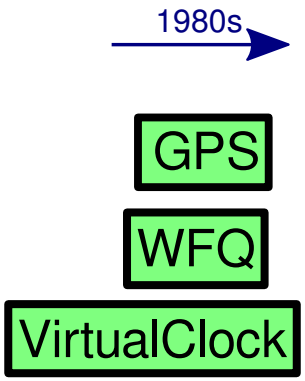
November 7, 2013

Switch Data Planes today

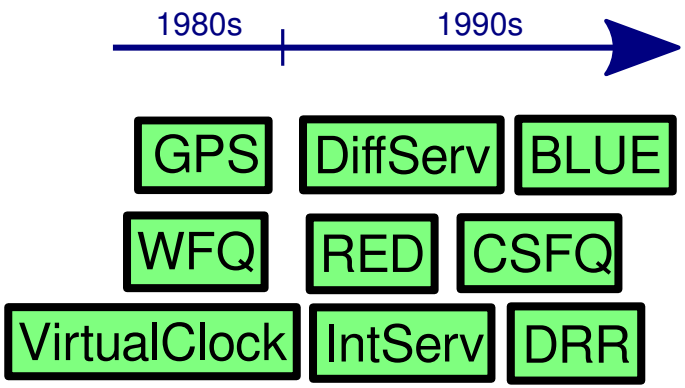
Two key decisions on a per-packet basis:

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

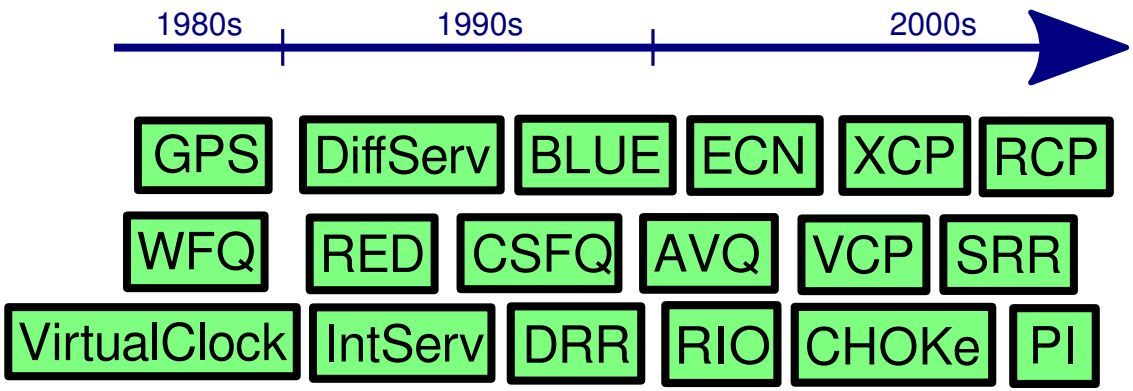
The long lineage of in-network algorithms



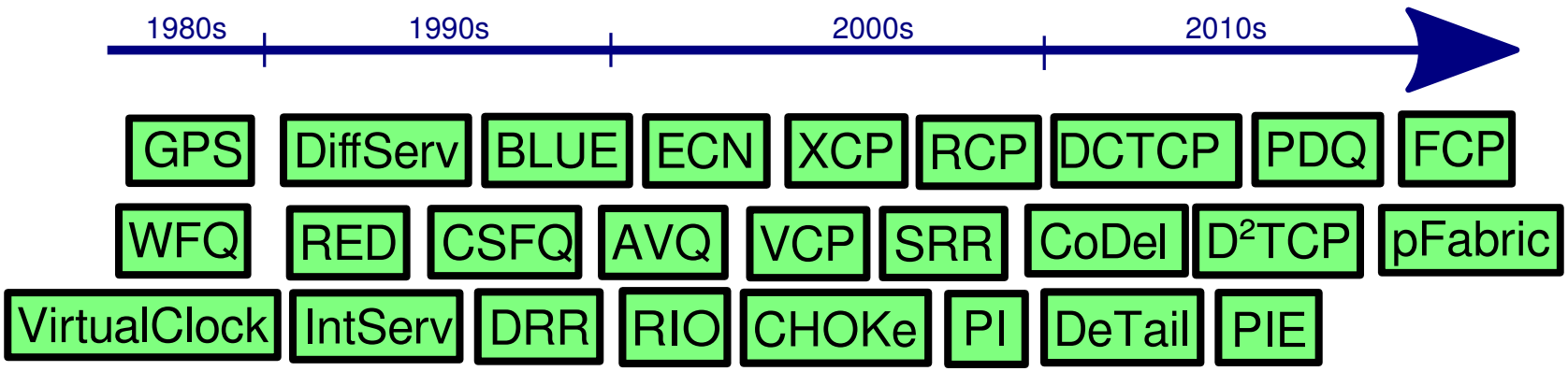
The long lineage of in-network algorithms



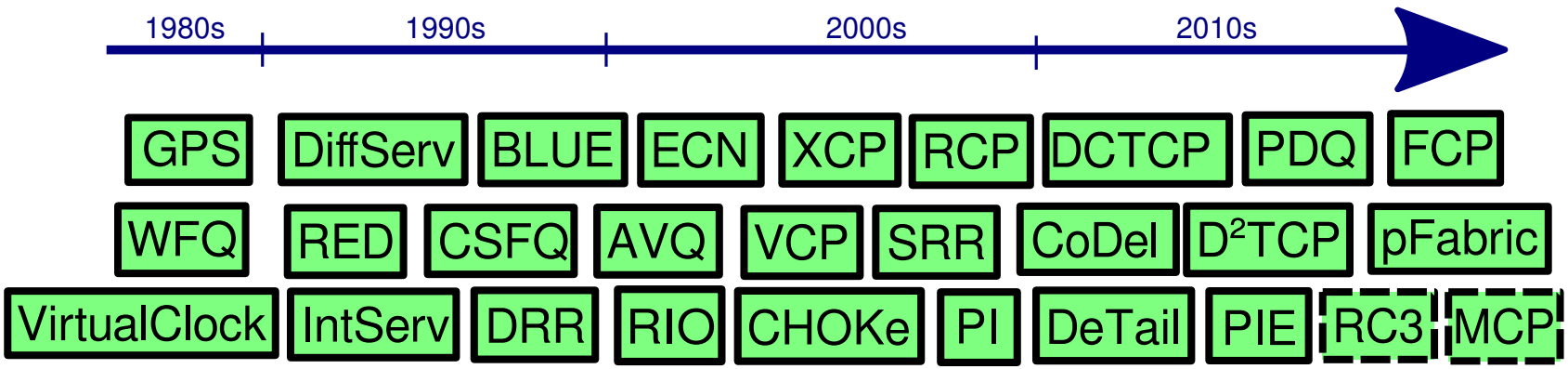
The long lineage of in-network algorithms



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.
- Some believe in a “silver bullet” knobless 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.

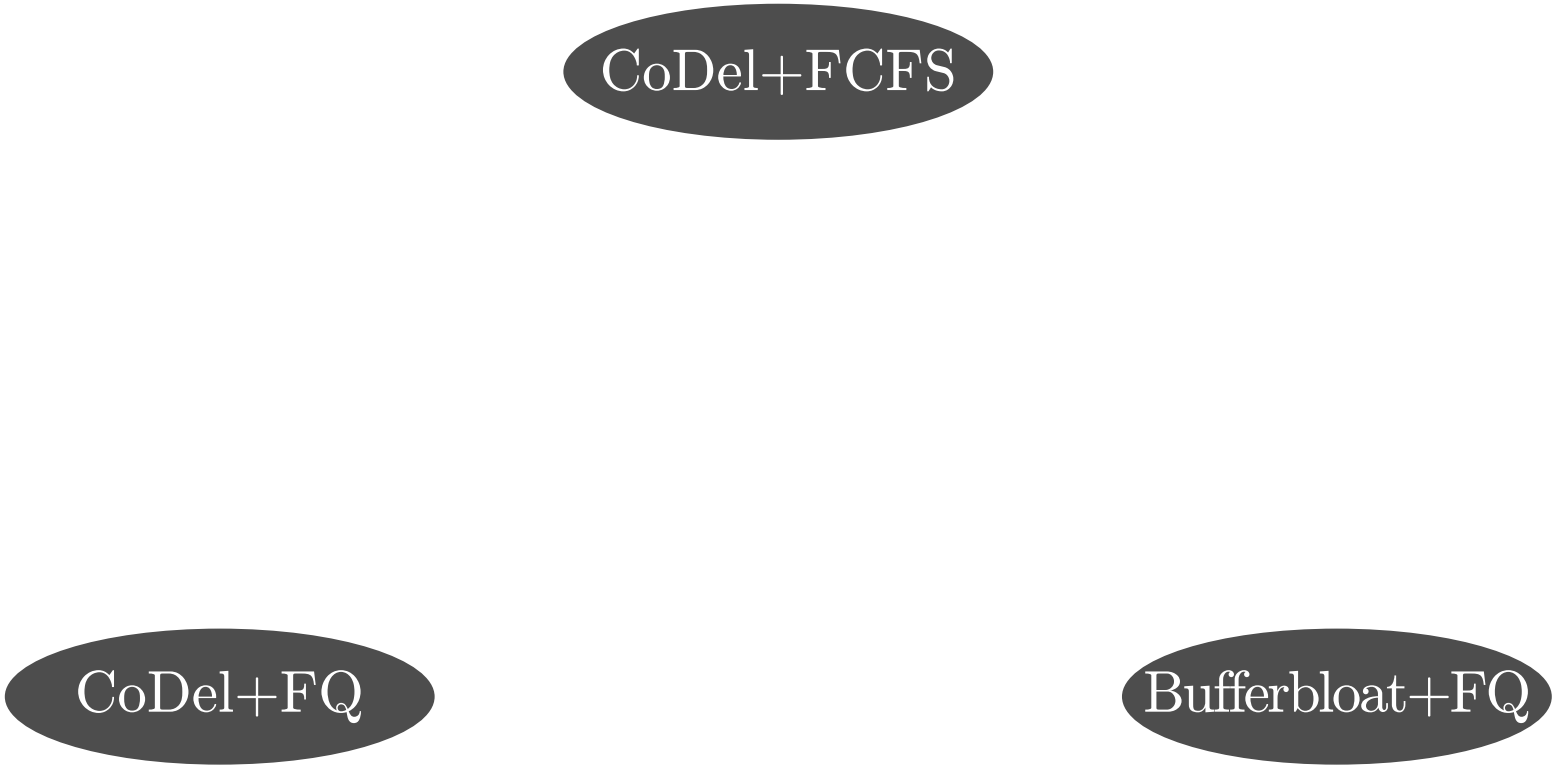
Quantifying "No Silver Bullet": Network Configurations

Configuration	**Description**
CoDel+FCFS	One shared FCFS queue with CoDel
CoDel+FQ	Per-flow fair queueing with CoDel on each queue
Bufferbloat+FQ	Per-flow fair queueing with deep buffers on each queue

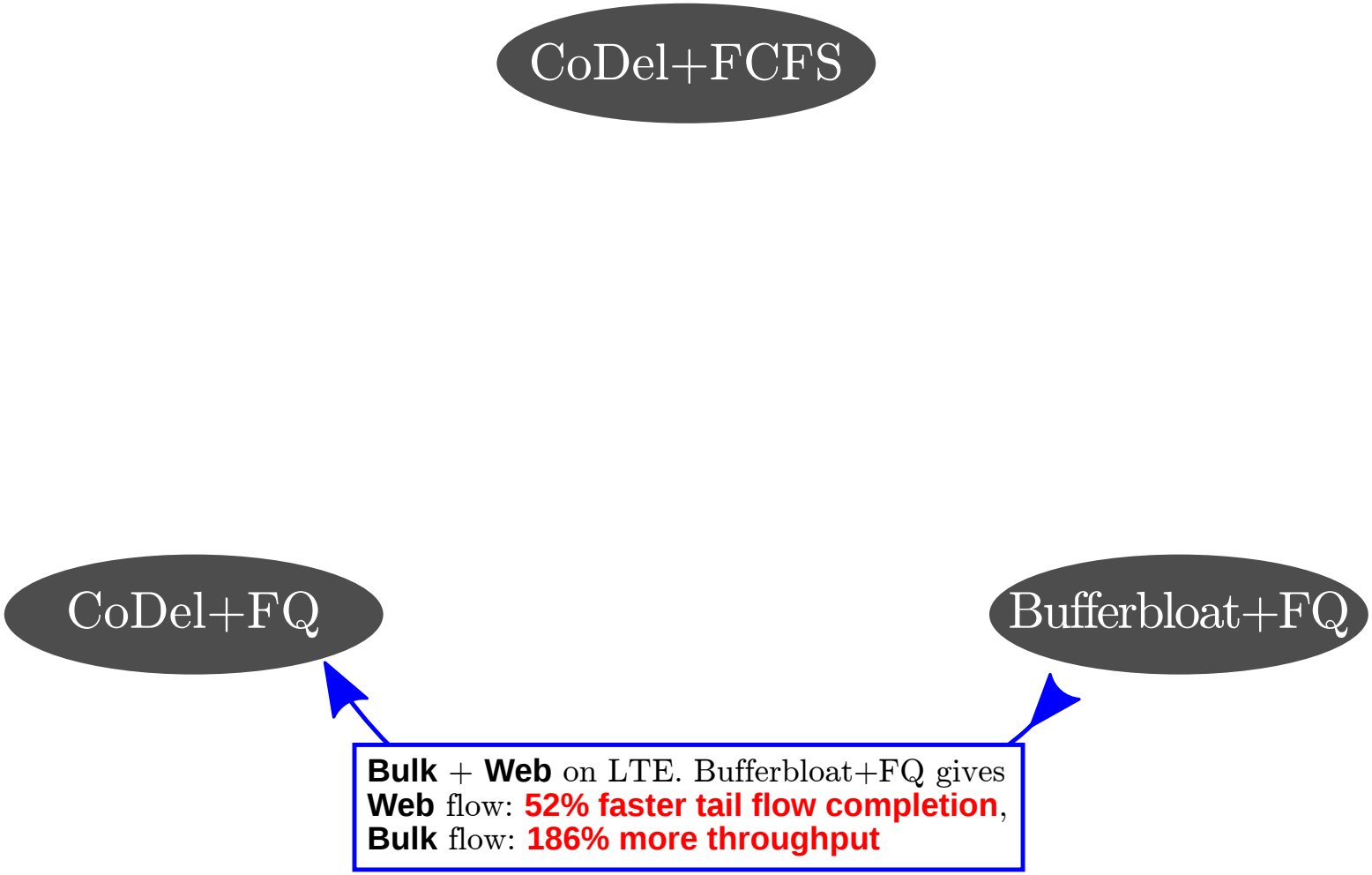
Quantifying "No Silver Bullet": Workloads and Objectives

Workload	Description	Objective
Bulk	Long-running bulk transfer flow	Max. throughput
Web	Switched flow with ON and OFF periods	Min. 99.9 %ile flow completion time
Interactive	Long-running interactive flow	Max. $\frac{\text{throughput}}{\text{delay}}$ (power)

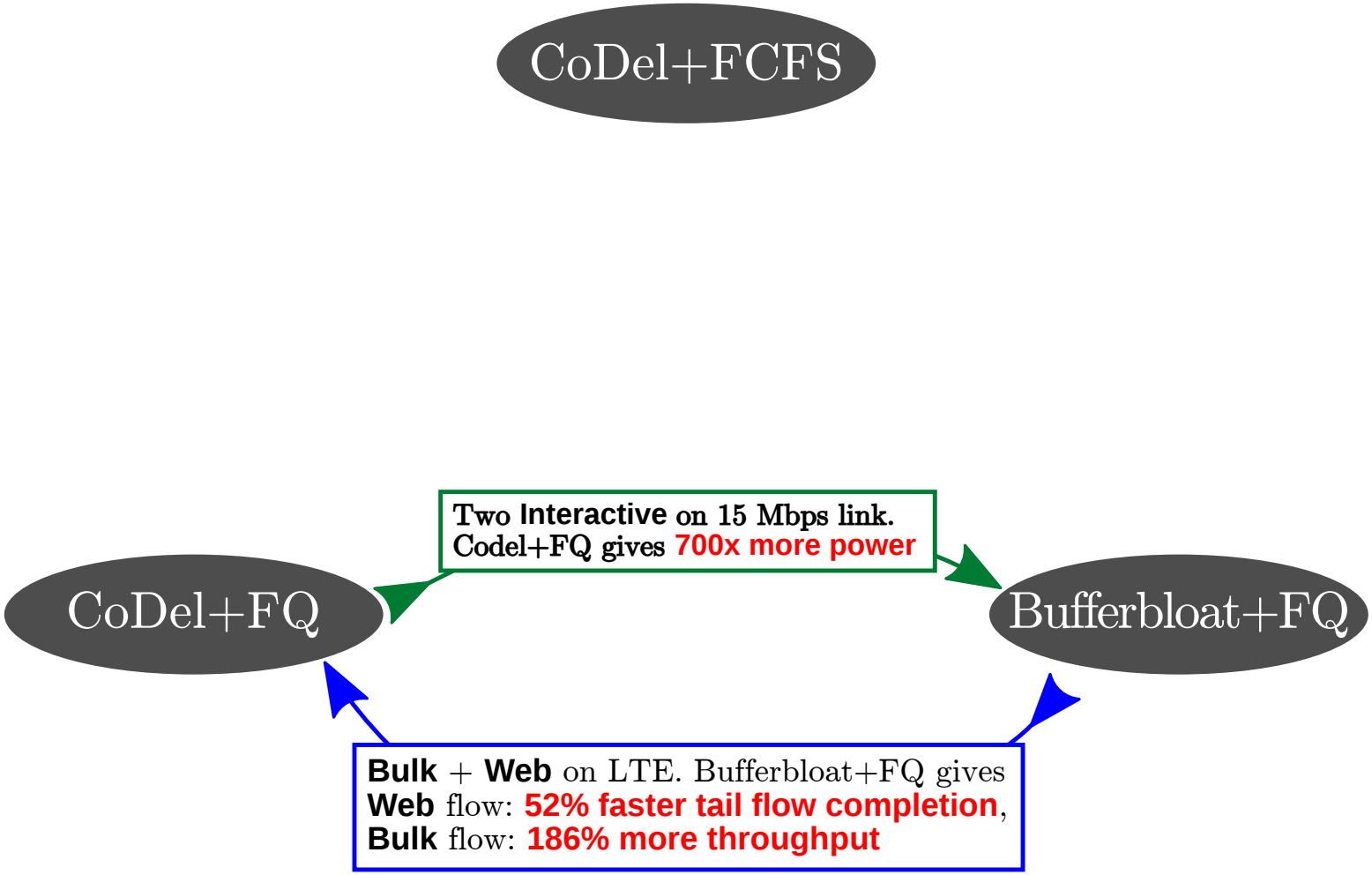
Quantifying "No Silver Bullet"



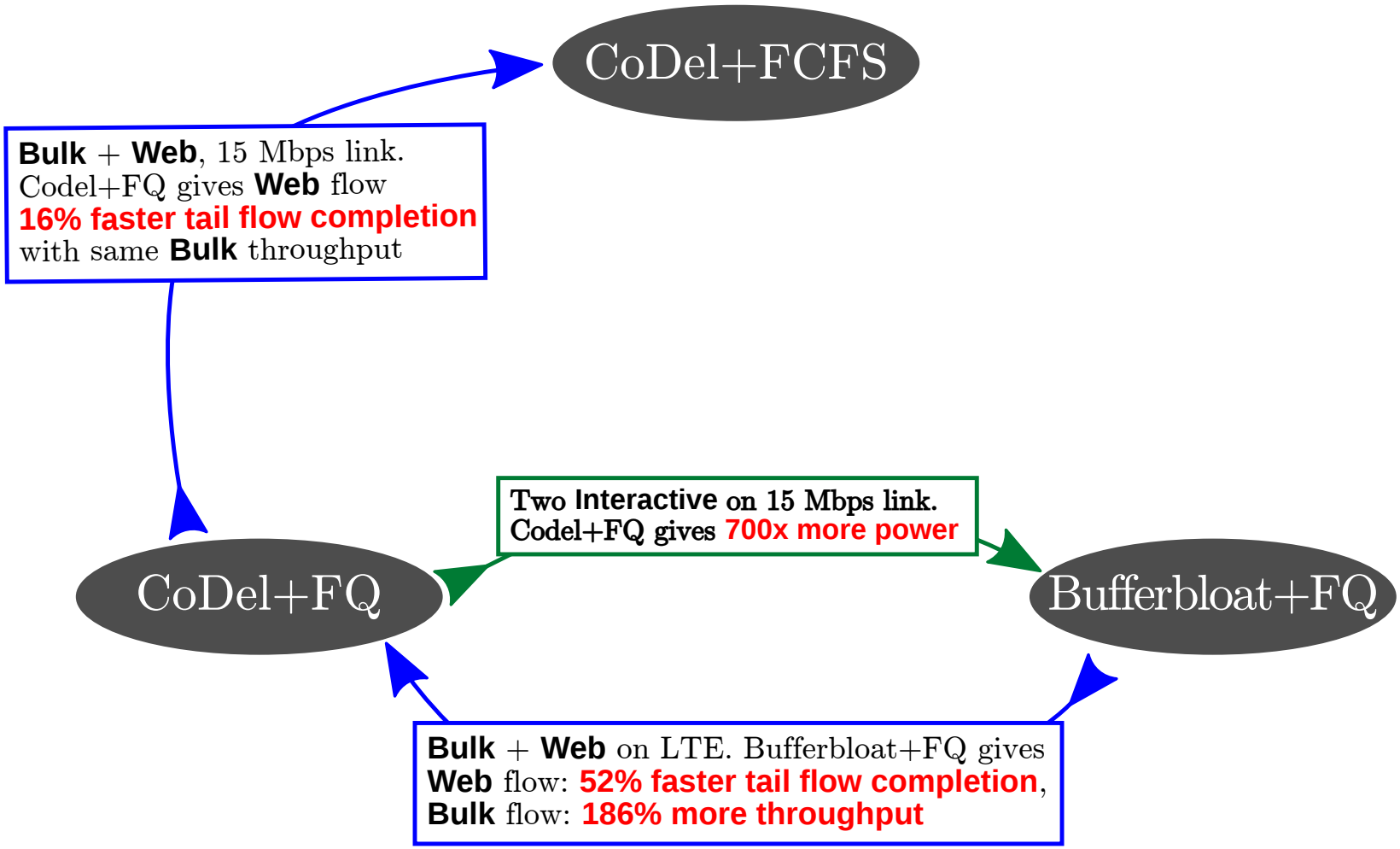
Quantifying "No Silver Bullet"



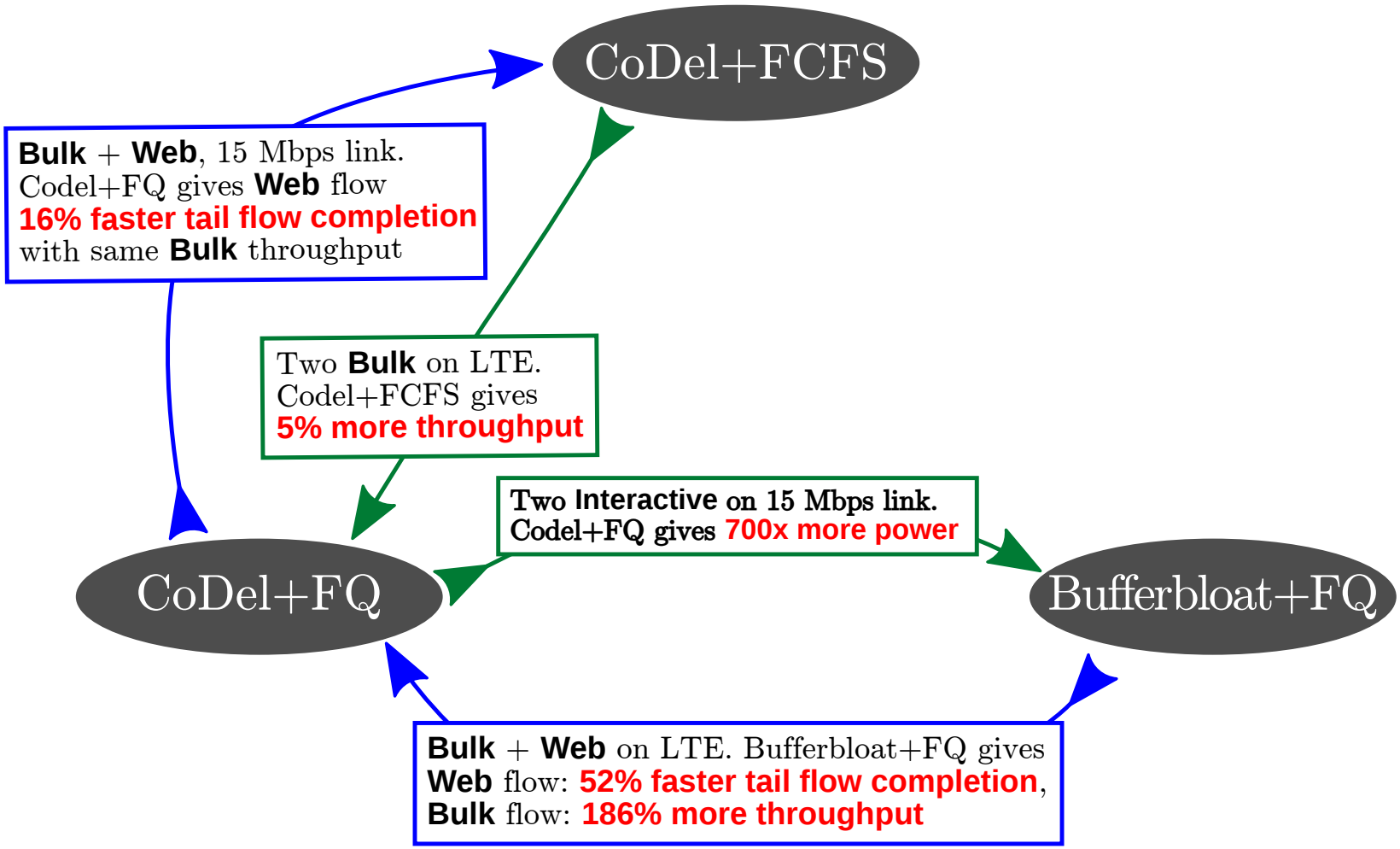
Quantifying "No Silver Bullet"



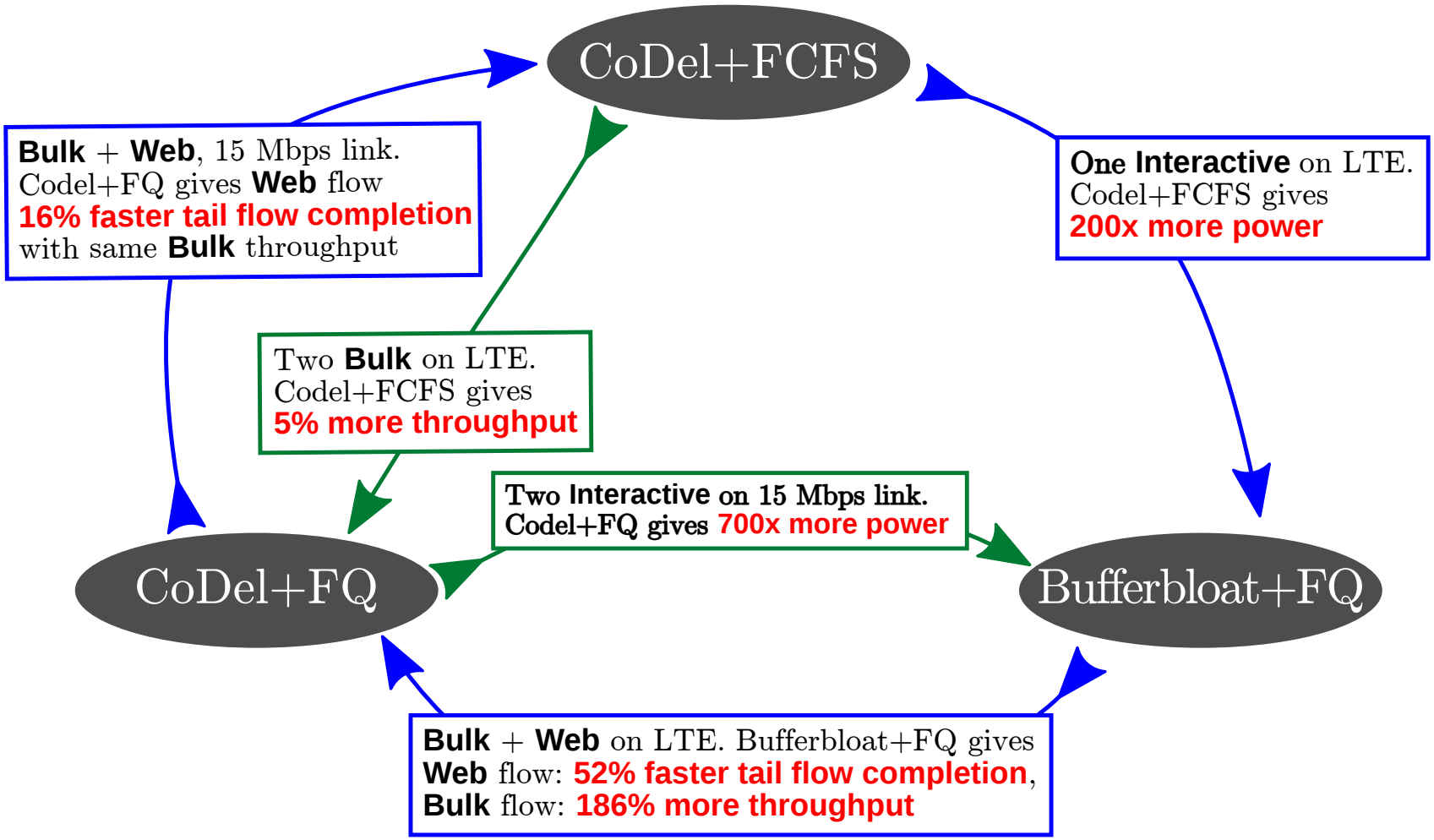
Quantifying "No Silver Bullet"



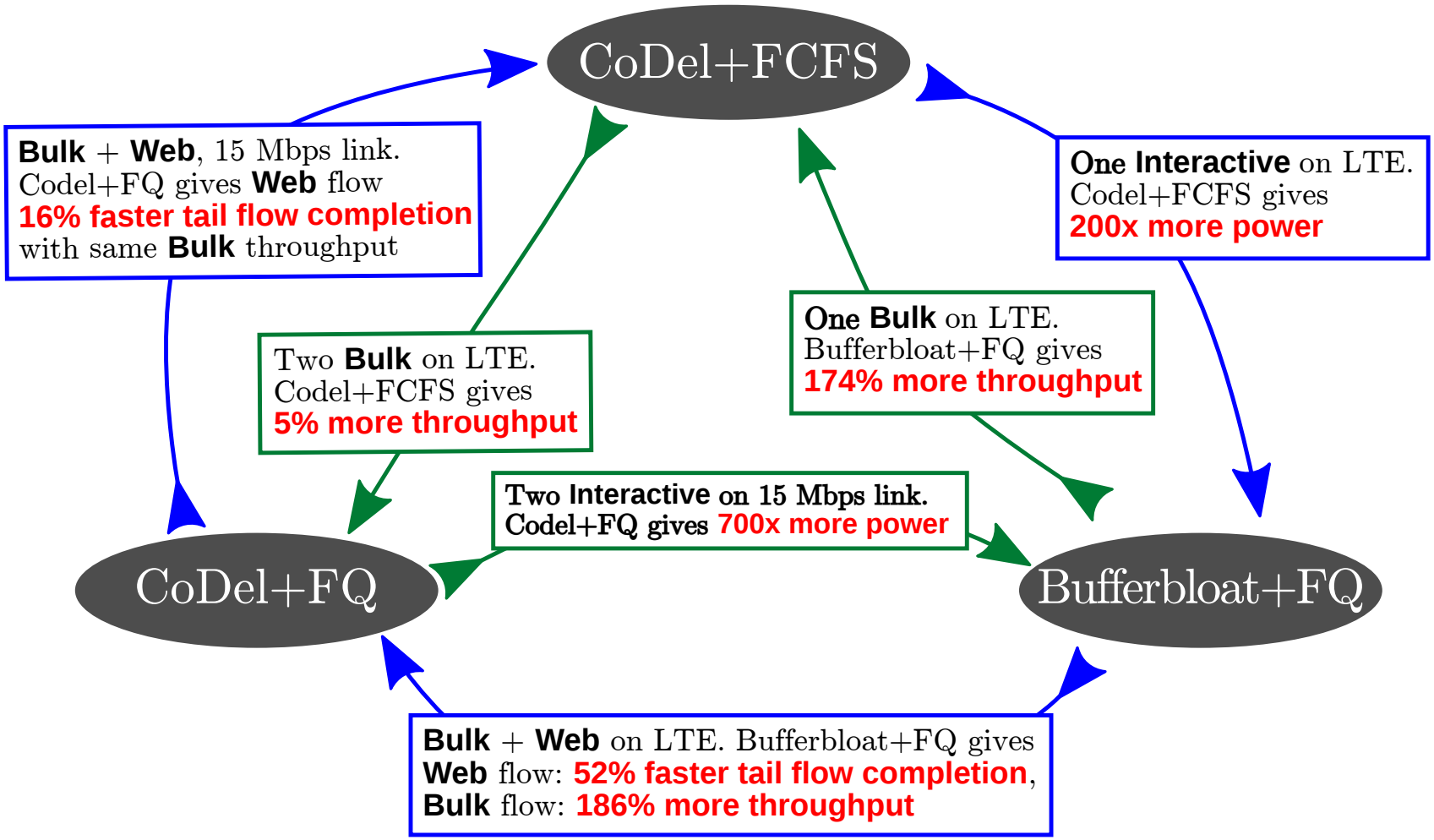
Quantifying "No Silver Bullet"



Quantifying "No Silver Bullet"



Quantifying "No Silver Bullet"



Why is no single data plane configuration the best?

- Bufferbloat on variable-rate links helps throughput!
  - Variable-rate links have an inherent delay-throughput tradeoff
- FCFS is preferable to Fair Queuing in some cases
  - When equally aggressive flows compete, they don't need protection from each other
  - Helps reduce tail packet delay
- Fair Queuing is required in some cases
  - When competing flows aren't equally aggressive, isolation helps

So what should the network designer do?

Architect a flexible data plane

- Programmable queue management and scheduling
- Not just for selecting among pre-built choices, but to change behavior in the field
- Because there is no silver bullet and innovation will continue!

Controlled flexibility: Want performance, security

- 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)
- I/O interfaces
  - Drop/mark head/tail of queue
  - Interrupts for enqueue/dequeue
- 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 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 $\times 10^6$ pkts/s ~10gbps	

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

Limitations and Practical Considerations:

- Cannot express several network functions that need payloads.
- How do applications signal objectives to the network?
- Feasibility at 10G on high port-density switches.
- Energy and Area overheads.

Related Work:

- Active Networking, e.g., ANTS
- Software Routers, e.g., Click
- Software-Defined Networking, e.g., OpenFlow

Conclusion

- There is no silver bullet to in-network resource control because of application and network diversity
- Algorithms will continue to evolve: the data plane should help
- Directions to reproduce results: http://web.mit.edu/anirudh/www/sdn-data-plane.html