DIBS: Just-in-time congestion mitigation for Data Centers

Kyriakos Zarifis, Rui Miao, Matt Calder,
Ethan Katz-Bassett, Minlan Yu, Jitendra Padhye

University of Southern California
Microsoft Research

Summary

Data center traffic patterns can cause **congestion**.

When switches can't buffer packets to forward, they **drop** them.

Instead of dropping packets, DIBS **detours** to neighboring switches.

This **shares buffer capacity** across buffers on different switches.

DIBS minimizes packet drops and retransmissions, which speeds up **job completion time**.

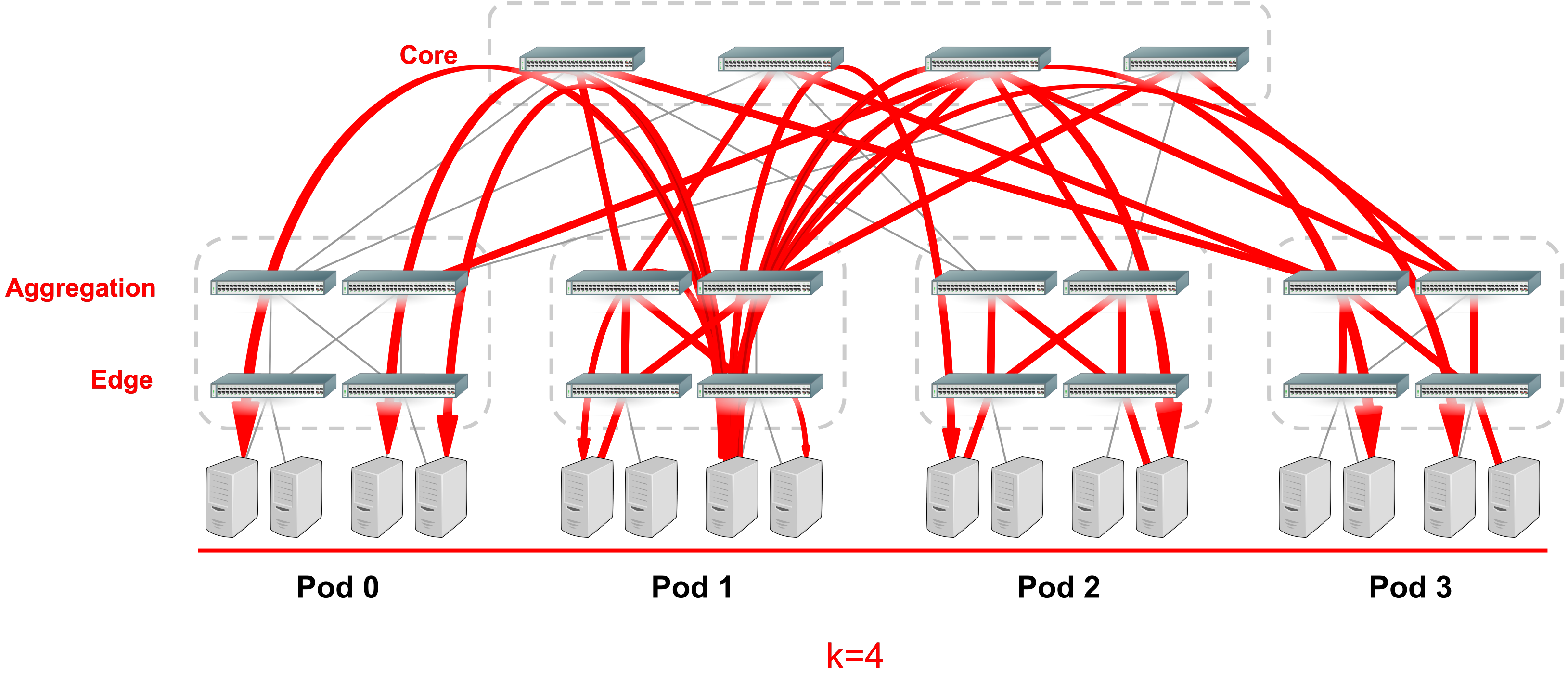
DIBS

Motivation

Design

Evaluation

A Data Center FatTree Topology



Problem Definition

Data center networks must be efficient with workloads and applications of variable throughput and latency requirements.

Traffic congestion degrades the performance of data center applications.

Ways to deal with congestion

1. **Workload-level** (> RTT timescales)

   Hedera [NSDI’10], Orchestra [SIGCOMM’11] ...

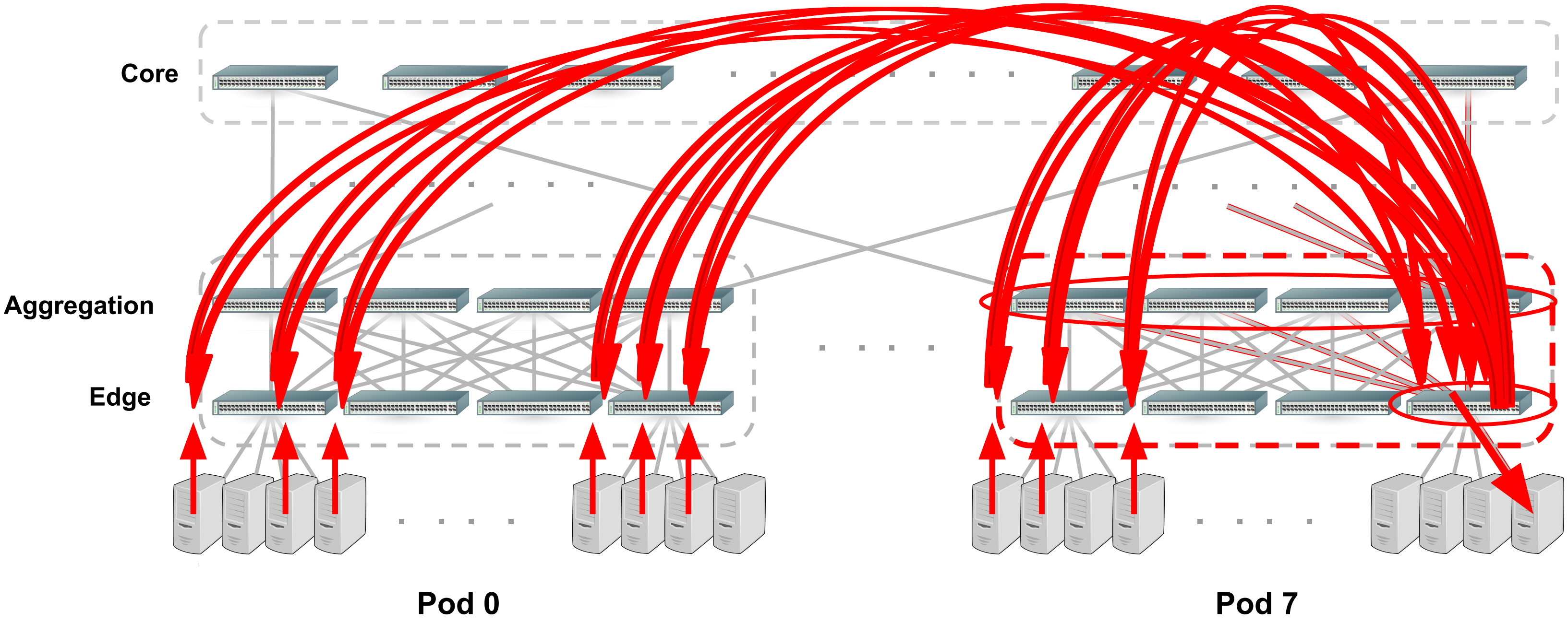
2. **Flow-level** (RTT timescales)

   DCTCP [SIGCOMM’10], Cutting Payload [NSDI'14] ...

3. **Packet-level** (< RTT timescales)

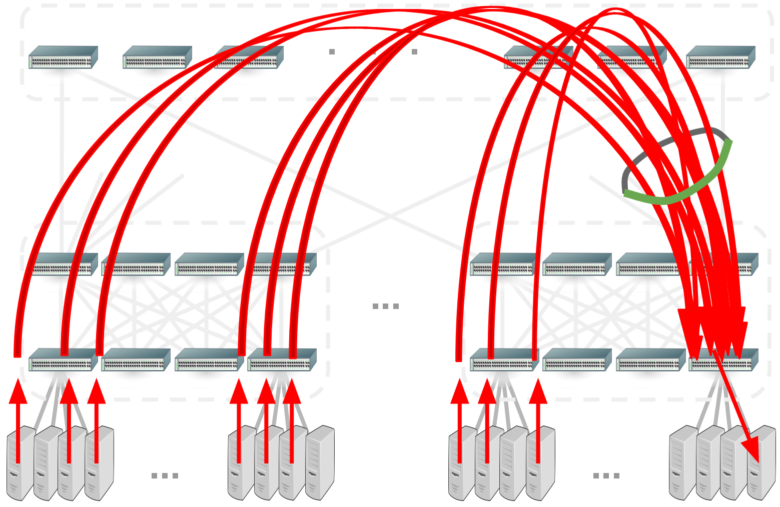
   DIBS

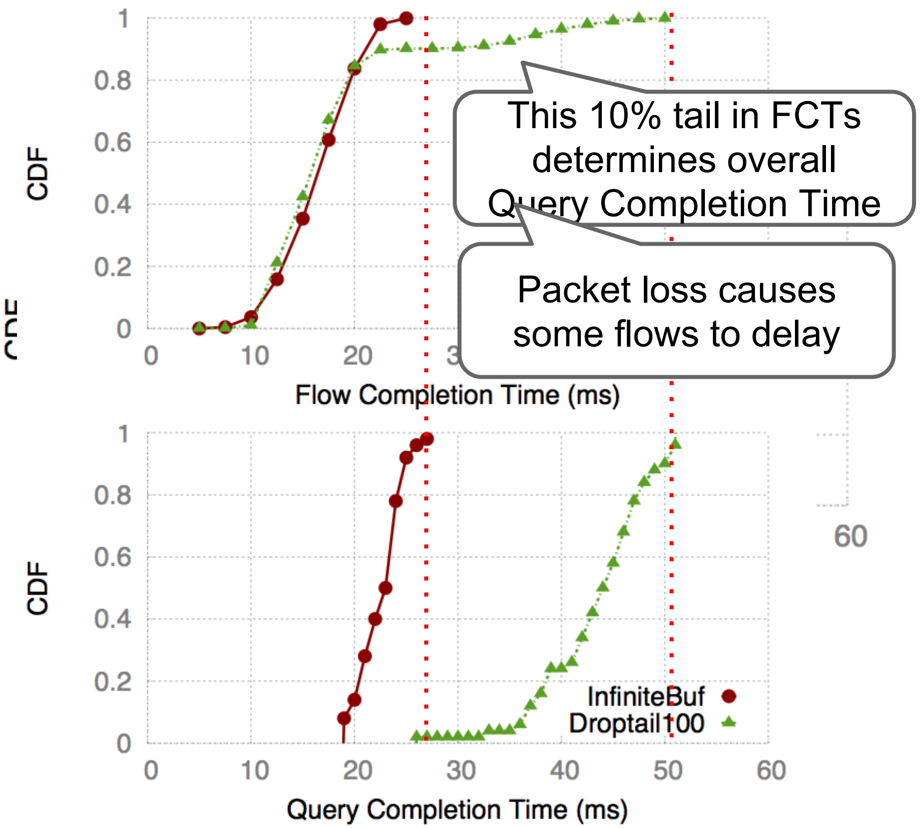
An example of extreme congestion



k=8

An example of extreme congestion





3. Packet-level

(< RTT timescales)

React just-in-time before packet loss

Congestion mitigation (not avoidance, like previous two)

Orthogonal to workload-level approaches (1)

Does not replace congestion control mechanisms(2)

… it **requires** one and complements it.

DIBS

Motivation

Design

Evaluation

Design Overview

Problem:

Bursty congestion causes packet loss and slow responses.

Approach:

"Detour-**I**nduced **B**uffer **S**haring**":** Instead of dropping packets when a buffer is full, detour them to nearby switches with spare capacity.

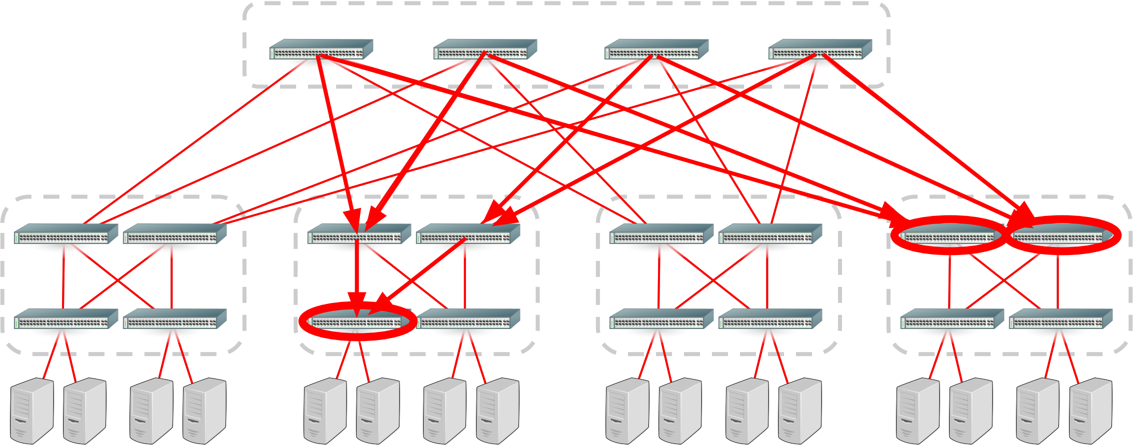
Buffer Size

Buffers need to be:

- Deep enough to absorb sudden **bursts** (simultaneous flows)
- Shallow enough for low **latency** (short queueing delays)

DIBS provides a way to share buffers across switches when needed

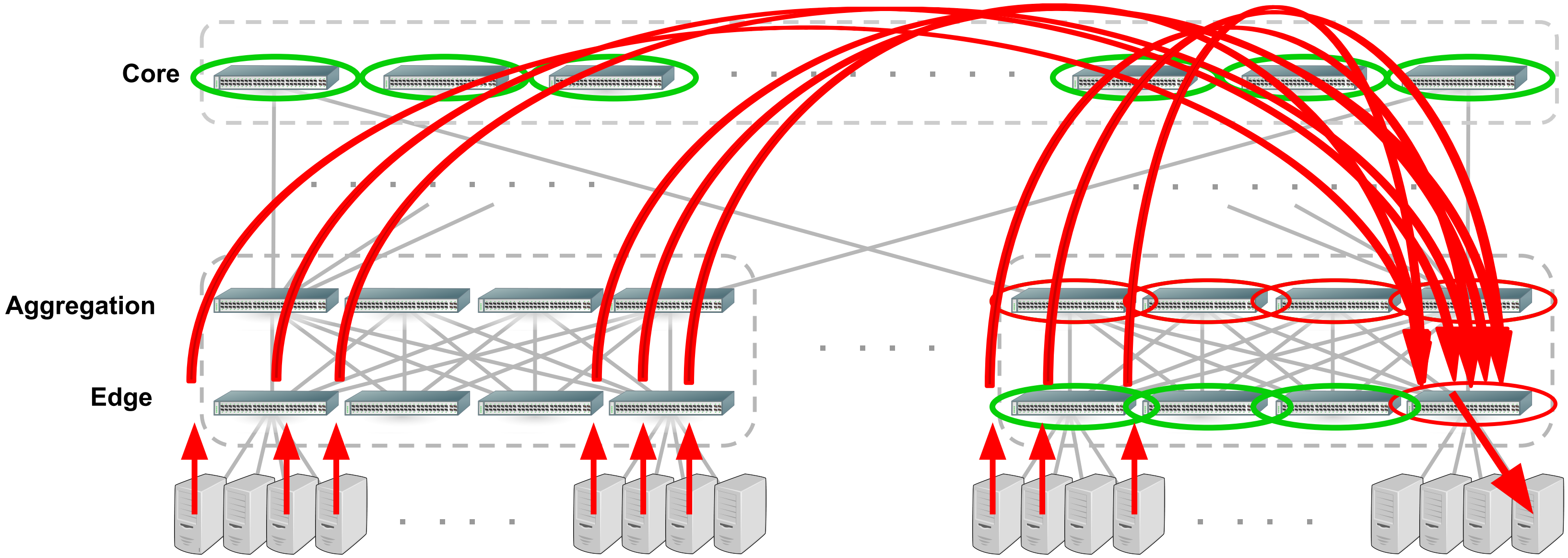
Why DIBS works



Observations:

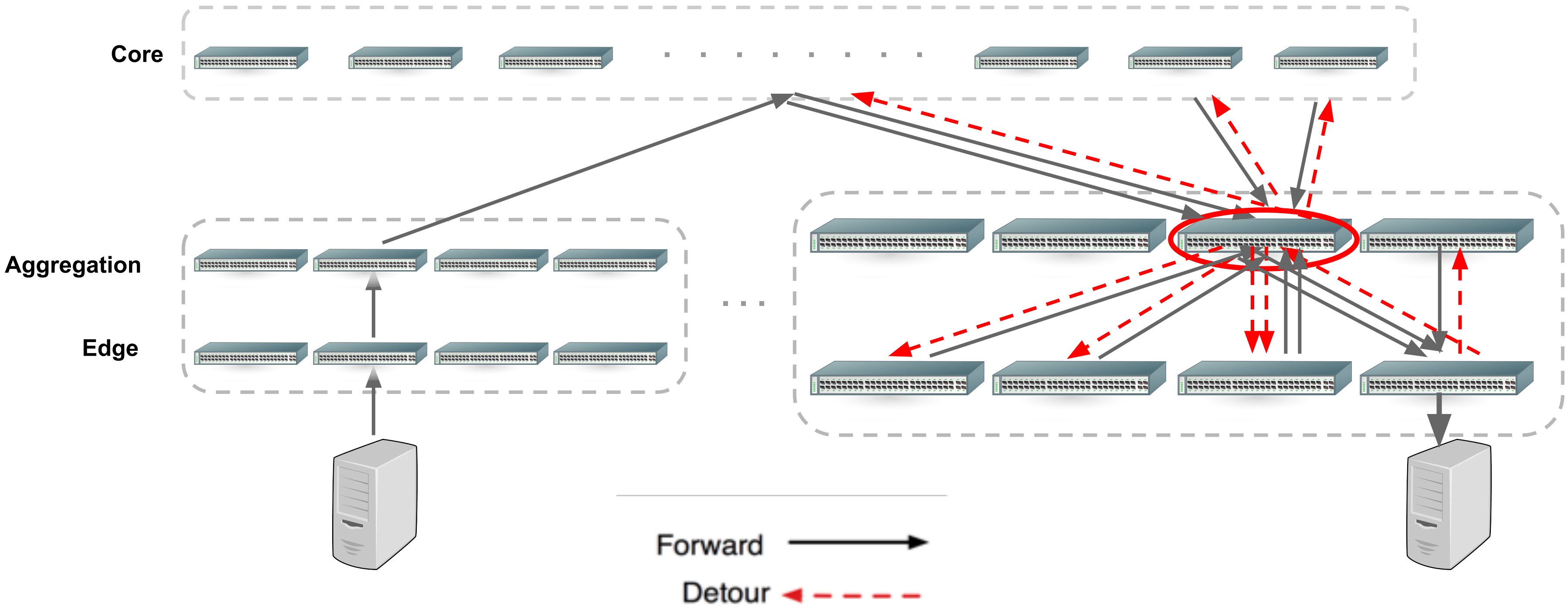
1. Congestion is usually **localized**, with spare buffering capacity nearby.
2. Links are **high-capacity**, so detouring doesn't add much latency.
3. Topology is densely connected, with **multiple paths** between hosts.

How DIBS works

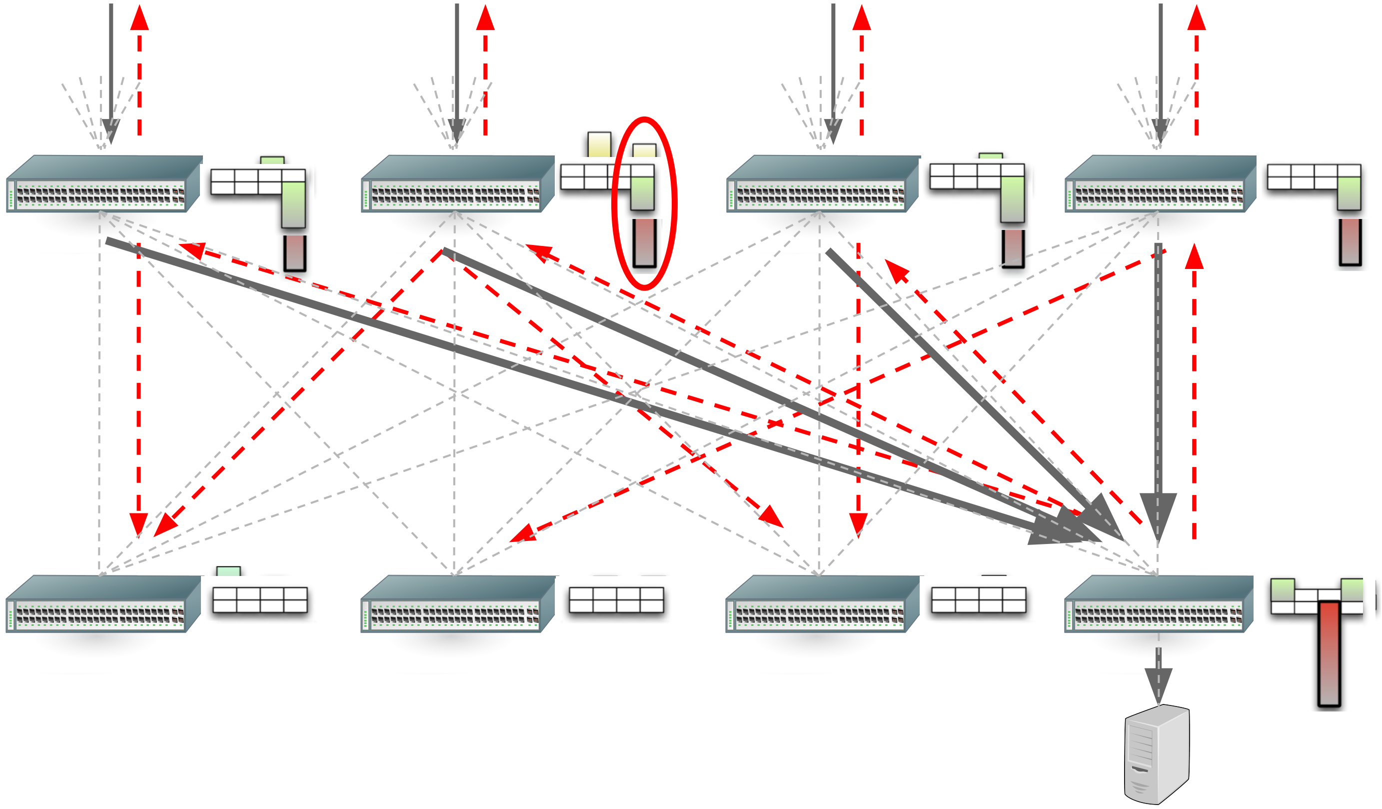


Detour excessive packets to neighboring switches
Use nearby buffering capacity to absorb a burst

Detouring Example



Buffer Occupancy



close
to the receiver.
started.

g to
build up close to the
receiver.

DIBS

Motivation

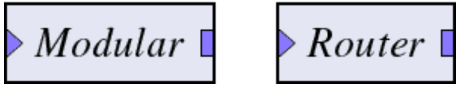
Design

Evaluation

Implementation

Hardware
netFPGA

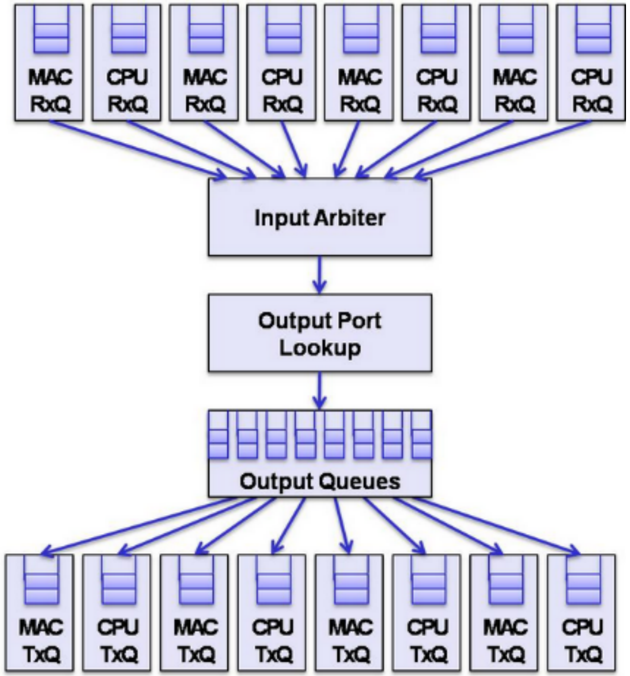
Software Router
Click!



Simulation
NS3

ns-3 NETWORK SIMULATOR

Implementation - NetFPGA



Pipelined modules

Modified "Output Port Lookup"

~50 LoC with additional logic

Zero additional latency and throughput

Implementation - Click Router

Click modular router: Easily extendable software router

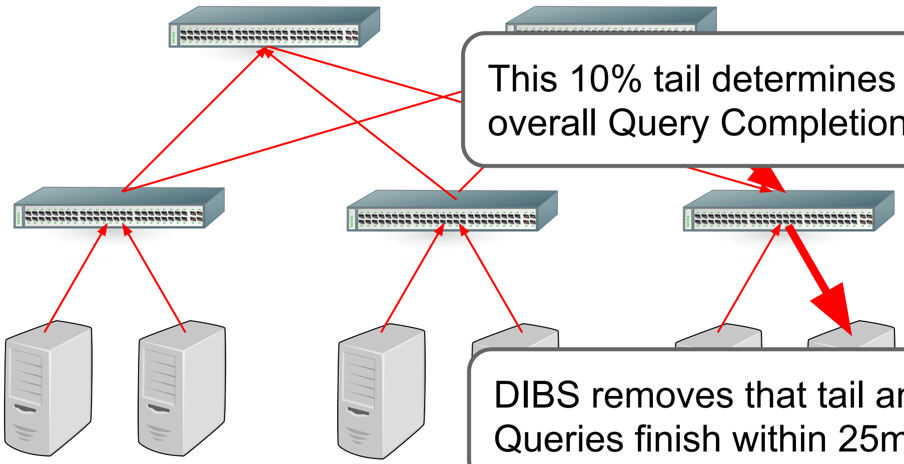
Extended existing RED module to detour instead of dropping

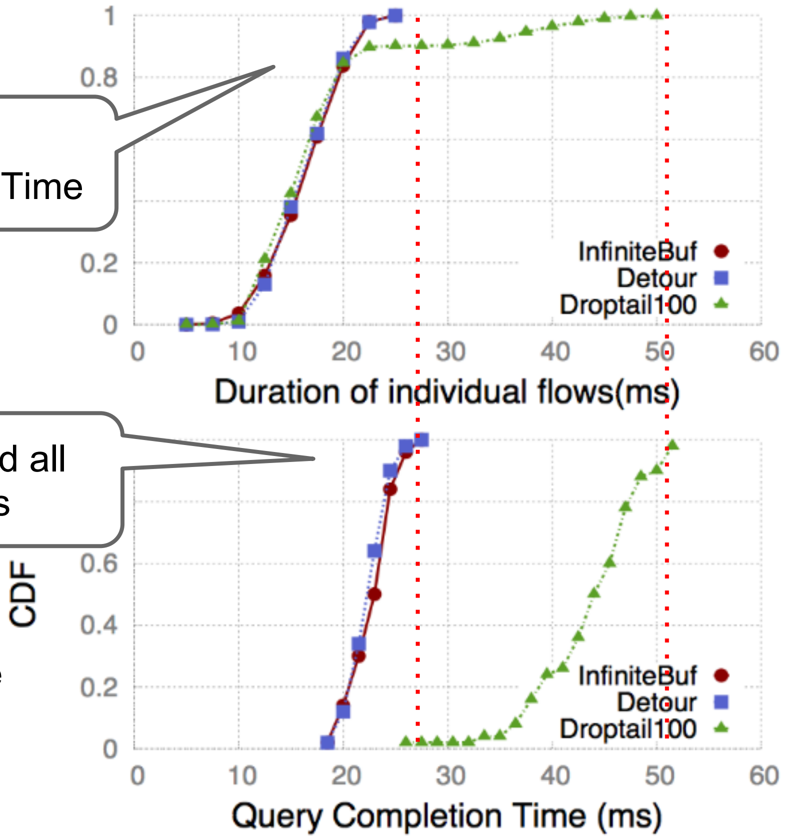
Before enqueuing to output queue, check whether queue is full
If so, enqueue to random output queue

~ 100 extra LoC

Implemented in a physical testbed of 5 switches / 6 hosts in EmuLAB

Click Testbed - FCT vs QCT





This 10% tail determines overall Query Completion Time

DIBS removes that tail and all Queries finish within 25ms

5 simultaneous flows to the same receiver

Delayed individual flows determine Query Completion Time

Enabling DIBS removes the FCT tail which minimizes QCT

NS3 Simulation

Large scale (k=8 FatTree, 128 servers)

Combination of two workloads:

1. Query Traffic (short, latency critical, many-to-one flows)
2. Background Traffic (longer, one-to-one flows)

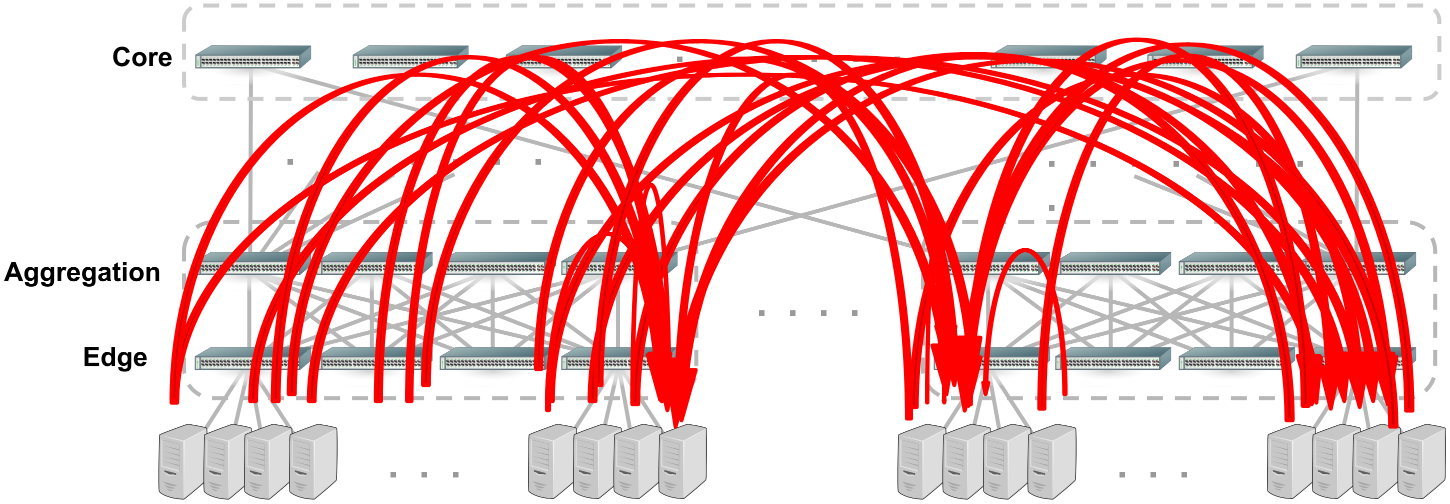
Wide range of tunable parameters:

1. Query Traffic: Queries per second (QPS), # of senders, response size, buffer size
2. Background traffic: Flow inter-arrival time

Over DCTCP for congestion control

NS3 Simulation - Workloads

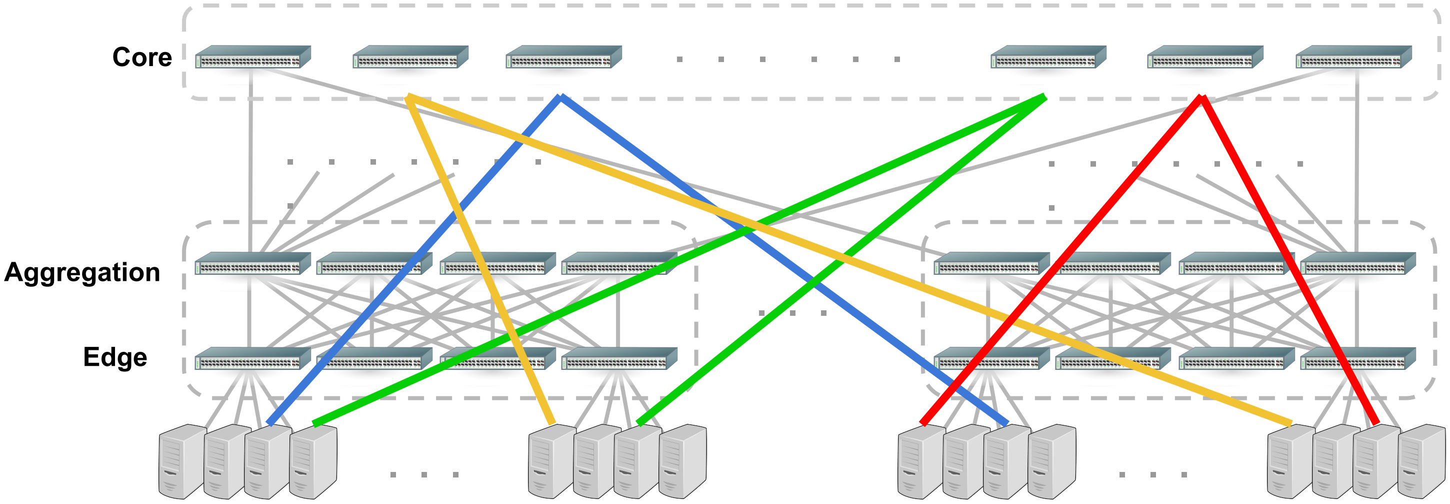
1. Query Traffic (Latency critical, many-to-one flows)



Query Traffic parameters: Queries per second (QPS), number of senders, flow sizes

NS3 Simulation - Workloads

2. Background Traffic (multiple background one-to-one flows)



Background Traffic parameters: Flow inter-arrival time, flow sizes

Query Traffic + Background Traffic

Mixed Query and Background traffic

Traffic settings according to production data centers

	Setting	Min	Max
Background traffic	BG inter-arrival (ms)	10	**120**
Query traffic	QPS	**300**	2000
	Response size (KB)	**20**	50
	Incast degree	**40**	100

Parameter sweep: Vary one factor while keeping the others fixed.

Evaluation goals

Does DIBS improve the performance of latency-critical jobs?
metric : **Completion time of Query traffic**

Does DIBS affect the performance of other flows?
metric : **Completion time of Background traffic**

How often do detours happen?
What traffic is detoured the most?
Does buffer sizes matter?
When does DIBS break?

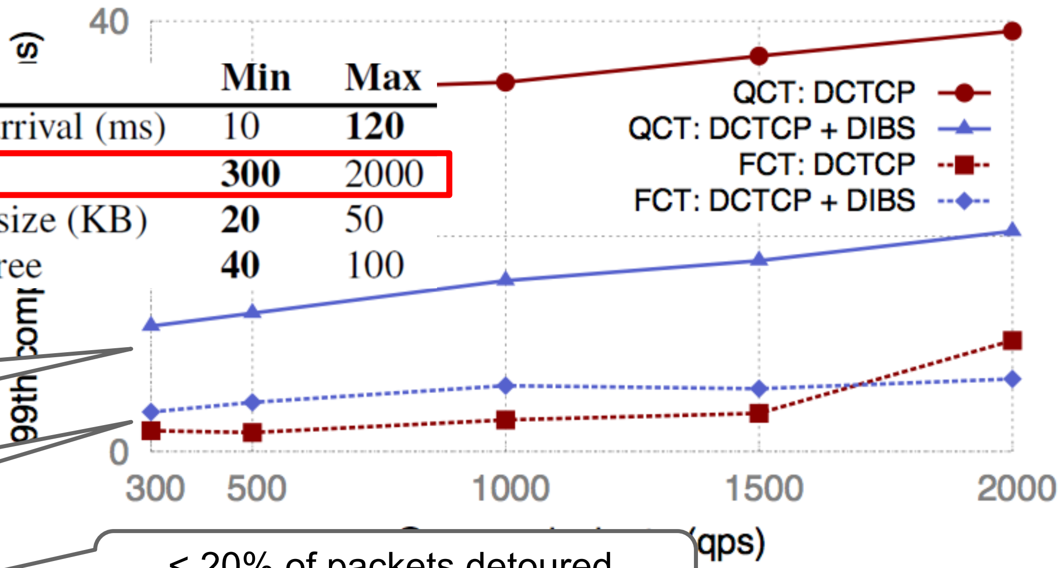
In the paper, also:
DIBS fairness
Impact of different TTL thresholds
Impact of oversubscription

Query Traffic + Background Traffic

Impact of Query Inter-arrival Time

Setting	Min
BG inter-arrival (ms)	10
QPS	**300**
Response size (KB)	**20**
Incast degree	**40**

Setting	Min	Max
BG inter-arrival (ms)	10	**120**
QPS	**300**	2000
Response size (KB)	**20**	50
Incast degree	**40**	100

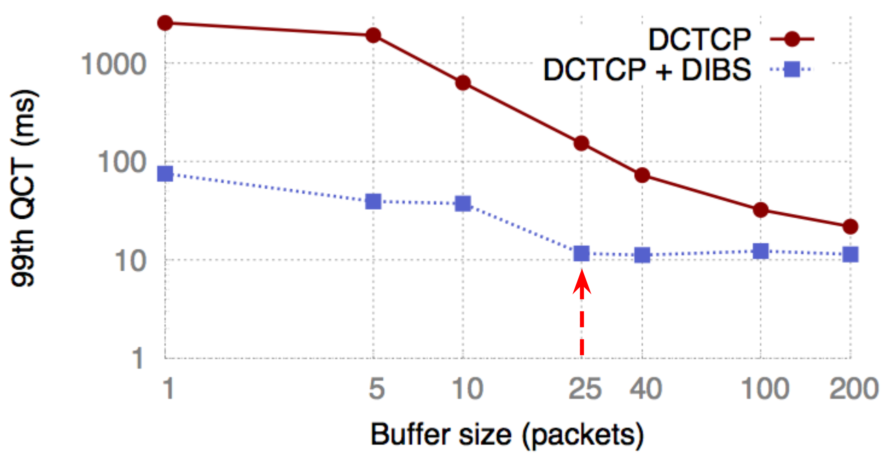


Query completion time decreased by 20ms.

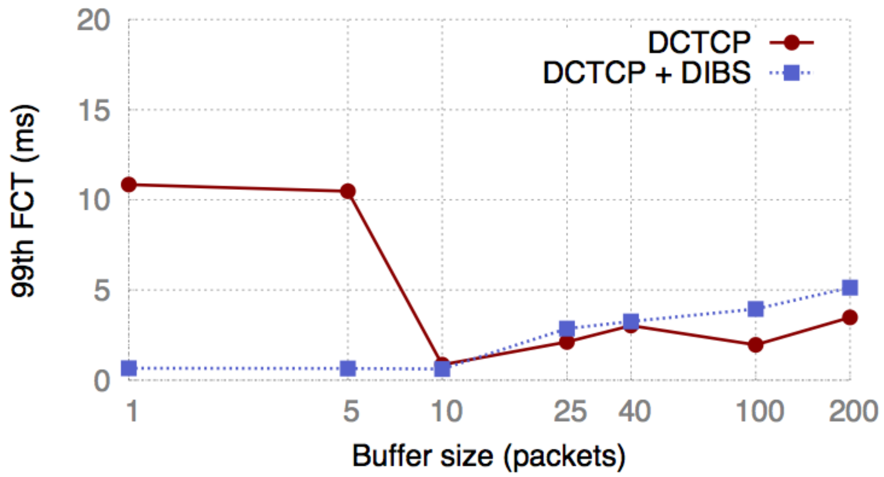
Slight increase to Background completion time.

< 20% of packets detoured, 90% of which were Query

Impact of buffer size



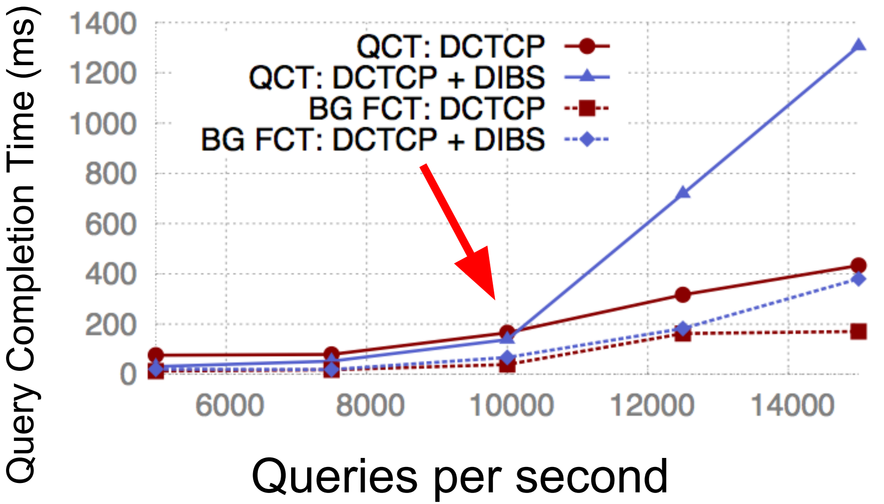
Query traffic



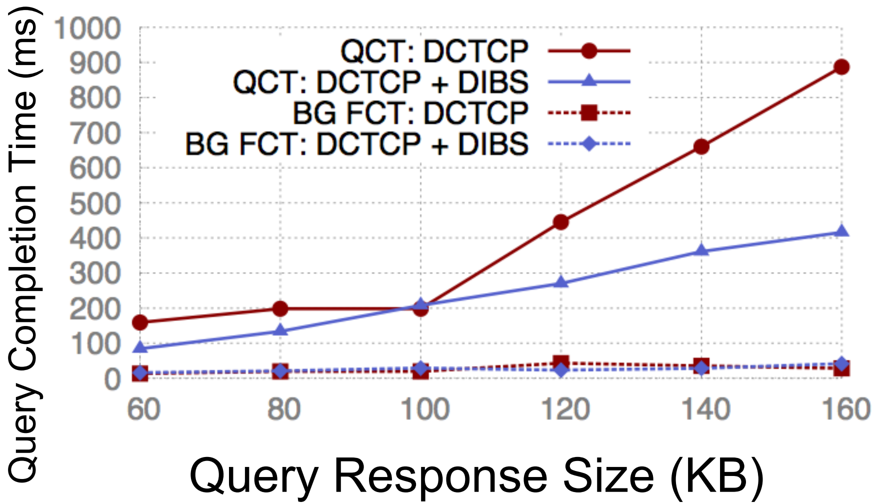
Background traffic

DIBS makes buffer size less relevant

When does DIBS break?



Breaking point exists at unrealistic query traffic intensity



DIBS does not break because for larger flows DCTCP has time to react

Results

Query Completion Times of latency-critical jobs consistently decreased significantly

Flow Completion Times of background flows only slightly increased in some cases

DIBS

Detours excessive packets to neighboring switches with spare buffering capacity to mitigate bursty congestion

Minimizes packet loss, speeding up job completion times

Interferes minimally with background traffic

Adds minimal overhead on hardware

Related work

Hedera/Orchestra: global load-balancing to minimize overlappintg

DCTCP: senders slow down according to congestions

Less is More (HULL): Phantom queues to pre-signal congestion

D3: prioritize flows based on deadlines

Per-packet ECMP, MPTCP

- can coexist

EFC/PFC/Infiniband: send pause msg to previous hop

- hard to tune, requires inter-switch communication, only previous hop

DeTail: per-packet load balancing and flow prioritization (PFC).

- requires larger switch changes, larger input buffer for pushback

Related work

Cutting payload (NSDI'14)

FastLane: Agile drop notification for DCs

pFabric: prioritized packets, aggressive retransmissions

optimize on retransmissions (but drops still happen and packets still get at the end of the aggregate queue. DIBS does the same, without the retrasmits)

Deflection/hot-potato: Bufferless/optical

not focusing on DC

1. Workload-level

(> RTT timescales)

Centralized flow scheduler

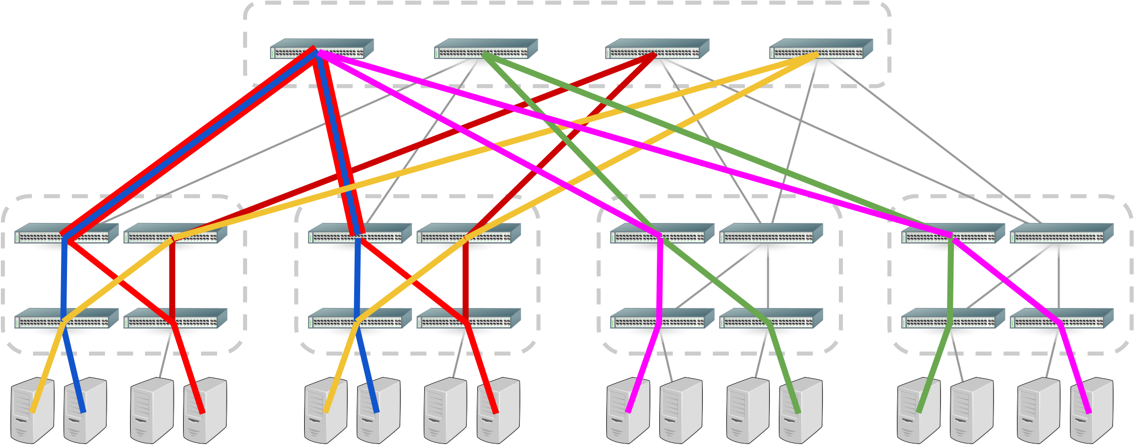
Global view of topology

Periodically estimate Traffic Matrix

Route flows dynamically to minimize overlapping paths, prevent congestion and maximize overall throughput

Ways to deal with congestion

1. **Workload-level** (flow duration timescales)



Route flows dynamically to minimize overlapping paths, prevent congestion and maximize overall throughput

2. Flow-level
(~ RTT timescales)

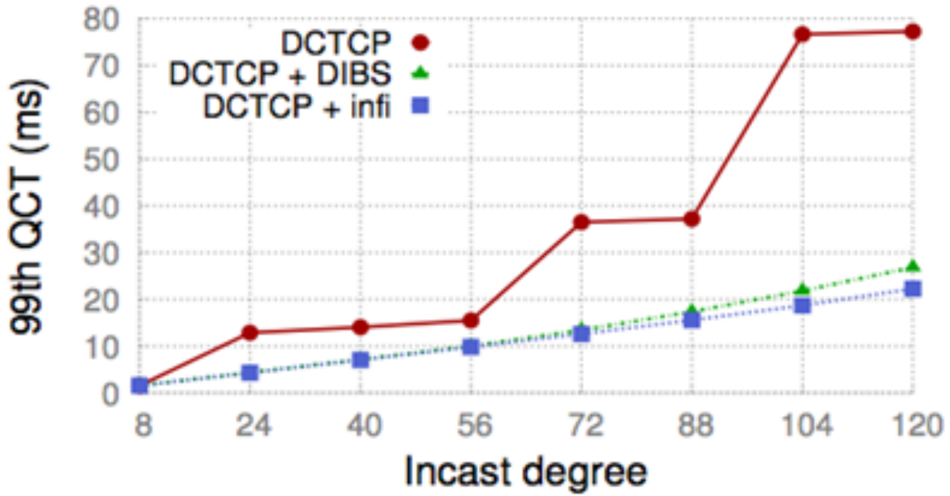
DCTCP [SIGCOMM 2010]

Act on each flow separately

Use ECN to notify sender to slow down according to the level of current congestion

Requires at least one RTT

1. Query Traffic

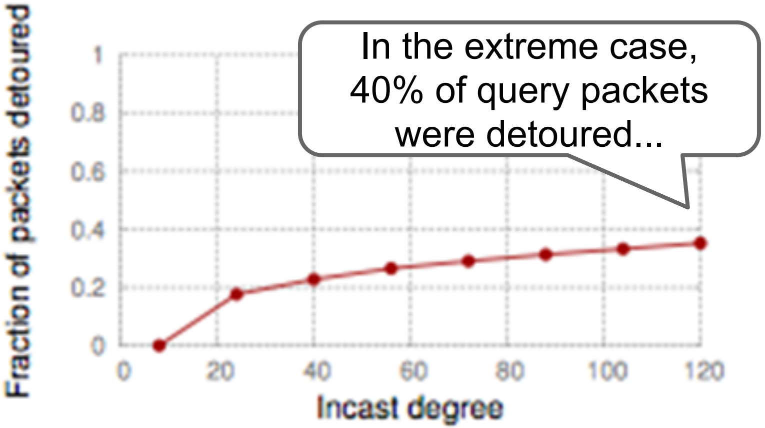


(number of senders per query)

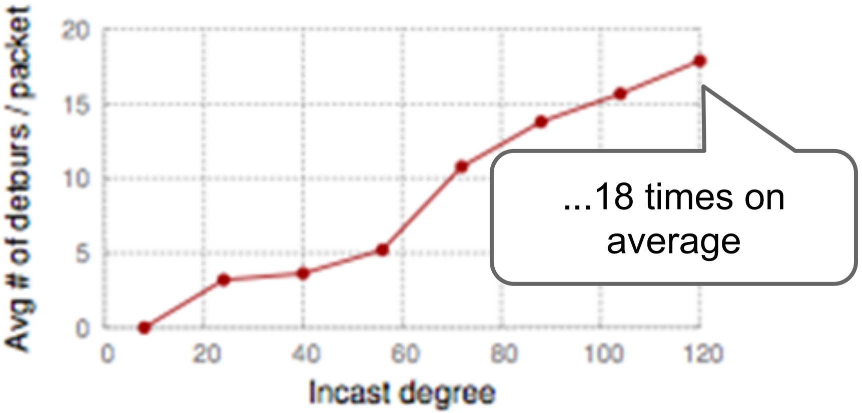
Query Completion Times for Incast traffic

The performance gap becomes bigger as the incast traffic becomes heavier

Number of Detours (Query Traffic)

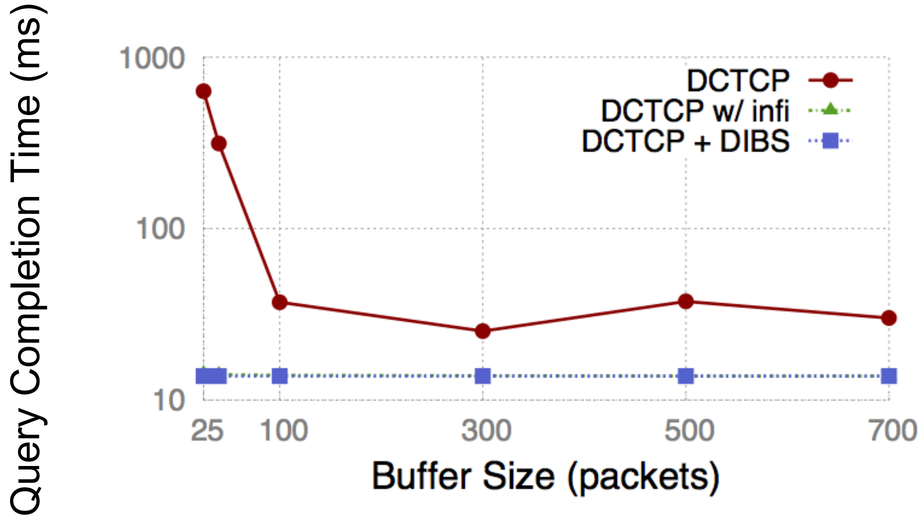


Fraction of packets detoured



Number of times each packet was detoured on average

Impact of buffer size



DIBS performs equally well regardless of buffer size

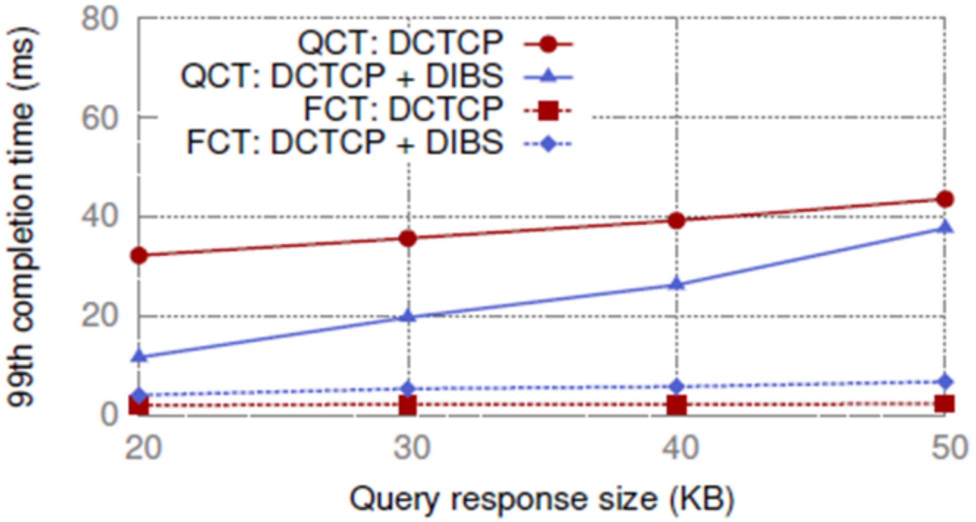
2. Query Traffic + Background Traffic

Impact of Query Response Size

Setting	Min	Max
BG inter-arrival (ms)	10	**120**
QPS	**300**	2000
Response size (KB)	**20**	50
Incast degree	**40**	100

QCTs of Query traffic decreased

DIBS less effective as flow sizes grow



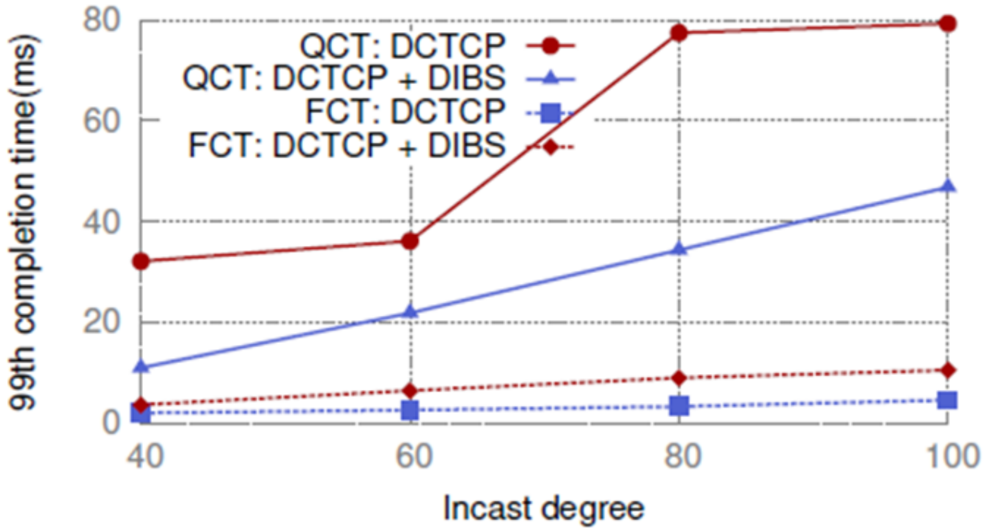
2. Query Traffic + Background Traffic

Impact of Incast Degree (# of senders)

Setting	Min	Max
BG inter-arrival (ms)	10	**120**
QPS	**300**	2000
Response size (KB)	**20**	50
Incast degree	**40**	100

QCTs of Query traffic decreased

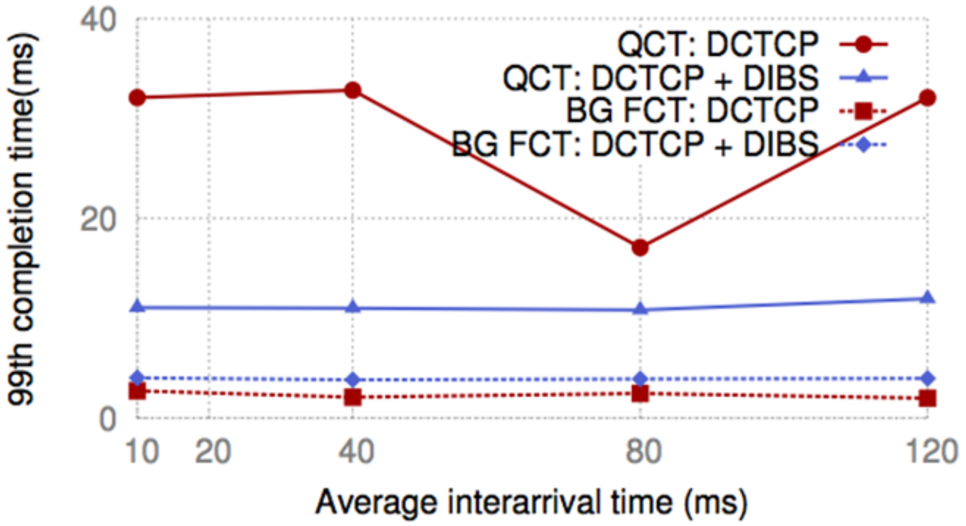
Performance gap increases with incast degree



2. Query Traffic + Background Traffic

Impact of Background Traffic

Setting	Min	Max
BG inter-arrival (ms)	10	**120**
QPS	**300**	2000
Response size (KB)	**20**	50
Incast degree	**40**	100



QCT of Query decreased up to 20ms

Slight increase to Background FCTs