Pushing the Performance Limits of Datacenter Networks

Vamsi Addanki, Oliver Michel, Stefan Schmid
Brief context of electrical systems
Brief context of electrical systems

Voltage
Brief context of electrical systems

Current

Voltage
Brief context of electrical systems

Power = Voltage x Current
Analogy to networked systems

Bottleneck Queue

Sender

Feedback

Receiver
Analogy to networked systems

Voltage = BDP + queue length
Analogy to networked systems

Voltage = BDP + queue length

Current = Rate
Analogy to networked systems

Voltage = BDP + queue length

Current = Rate

Power?

Upnext... stay tuned!
PowerTCP in a Nutshell

- **Power**-based congestion control
- Quickly reacts to congestion **without losing throughput**
- Rapidly converges **within 1 RTT**
- Fair and **asymptotically stable**
- Reduces FCTs for short flows **by up to 90%**
How do we measure Power?
The debate over congestion signals

Microsoft says **ECN** is better [dctcp]

Google says **delay** is simple and effective [Timely, Swift]

Alibaba says **INT** is accurate [HPCC]

**ECN, Delay or INT are essential**

What matters more: what we do with it
The debate over feedback signals

A debate over how to use the feedback
Rare glimpse of Google datacenter
Rare glimpse of Google datacenter
Fear of the buffer

Buffer per unit capacity (KB/Gbps)

- Br. 56538
- Trident+
- Trident II
- Tomahawk
- Tofino

KB per Gbps
DC workloads and short flows

(a) Web search workload

(b) Data mining workload
DC workloads and short flows

Majority traffic volume is from long flows

Majority Flows are short

(a) Web search workload

(b) Data mining workload
DC workloads and short flows

I have a phobia that throughput is always low

I have a constant fear that delay is always high

(b) Data mining workload
Not just queueing but **quickly utilizing available bandwidth is important too**

e.g., Emerging Reconfigurable Datacenter Networks (RDCNs)
Fine-grained congestion control is important for datacenter performance
Timeline of congestion control in datacenters

- Reno, Cubic
- DCTCP, DCQCN
- Timely
- HPCC
- Swift
Timeline of congestion control in datacenters

- **Voltage-based** (BDP + Queue Length)
  - ECN/Loss (eg., DCTCP)
  - RTT based (eg., Swift)
  - Inflight based (eg., HPCC)
- **Current-based** (Total transmission rate)
  - RTT-gradient based (Eg., Timely)
Voltage-based

Reaction to queue length or RTT
Reaction to queue length or RTT

Loss/ECN
eg., DCTCP

Voltage-based
Reaction to queue length or RTT

Loss/ECN eg., DCTCP

Delay eg., Swift

Voltage-based
Loss/ECN eg., DCTCP
Delay eg., Swift
Inflight eg., HPCC
Voltage-based

Reaction to queue length or RTT
Current-based

- Loss/ECN (e.g., DCTCP)
- Delay (e.g., Swift)
- Inflight (e.g., HPCC)

Voltage-based

Reaction to queue length or RTT

Reaction to variations
Current-based

RTT gradient
eg., Timely

Loss/ECN
eg., DCTCP

Delay
eg., Swift

Inflight
eg., HPCC

Voltage-based

Reaction to queue length or RTT
Problems of existing approaches

Fundamentally limited to a single dimension
Problems of existing approaches

5 Packets
Problems of existing approaches
Problems of existing approaches

15 Packets
Problems of existing approaches
Problems of existing approaches

Increasing at 8 x BW

25 Packets

BW
Problems of existing approaches

- Increasing at 8x BW
- Draining at max rate

25 Packets
Problems of existing approaches

Increasing at 8x BW

25 Packets

50 Packets

Increasing at 8x BW

BW

BW
Problems of existing approaches

Fundamentally limited to a single dimension
Summary of Our Analysis

- **Voltage-based**
  - Can in-principle achieve near-zero queue equilibrium
  - Slow reaction

- **Current-based**
  - Unstable with no equilibrium
  - Fast Reaction
Current-based

DCTCP

Swift

HPCC

Reaction to variations

Timely

Voltage-based

Better inflight control

Reaction to queue length or RTT

Better inflight control
Timely

DCTCP
Swift
HPCC

Voltage-based

Current-based

Reaction to queue length or RTT

Better reaction time

Better inflight control

Reaction to variations

Timely
Current-based

Timely

Reaction to variations

Better reaction time

Voltage-based

Better inflight control

DCTCP

Swift

HPCC

Reaction to queue length or RTT

Better inflight control

Current-based

Voltage-based

DCTCP

Swift

HPCC

Reaction to queue length or RTT

Better reaction time
The notion of power

Power = Voltage \times Current

\[ \Gamma = (q(t) + b \times \tau) \times (q(t) + \mu(t)) \]

- Power
- Voltage: BDP+queue bytes
- Current: Total rate
The notion of power

Enqueue rate = queue-gradient + Dequeue rate
\[ \lambda(t - t^f) = \dot{q}(t) + \mu(t) \]
Sending rate = Window per RTT
\[ \lambda(t) = \frac{w(t)}{\theta(t)} \]
RTT = queueing delay + base RTT
\[ \theta(t - t^f) = \frac{a(t)}{b} + \tau \]
The notion of power

\[ b \times \omega(t - t^f) = \left( q(t) + b \times \tau \right) \times \left( \dot{q}(t) + \mu(t) \right) \]

**Voltage** \hspace{1cm} **Current**
The notion of power

A function of both queue length and variations
The notion of power

A function of both queue length and variations

- Detects increased queue lengths
The notion of power

A function of both queue length and variations

- Detects increased queue lengths
- Detects congestion onset and intensity
The notion of power

A function of both queue length and variations

- Detects increased queue lengths
- Detects congestion onset and intensity
- Detects rapid drop in queue lengths
- **Power-based CC**
  - Better reaction time
  - Better inflight control
  - Timely

- **Current-based**
  - Reaction to variations
  - Better reaction time

- **Voltage-based**
  - Reaction to queue length or RTT

- **Algorithm Comparison**
  - DCTCP
  - Swift
  - HPCC
Power-based CC

Timely

Voltage-based

Better inflight control

DCTCP

Swift

HPCC

Reaction to queue length or RTT

Current-based

Better reaction time
PowerTCP control law

\[ w_i(t + \delta t) = \gamma \cdot \left( w_i(t) \cdot \frac{e}{f(t)} + \beta \right) + (1 - \gamma) \cdot w_i(t) \]

New window size
PowerTCP control law

\[ w_i(t + \delta t) = \gamma \cdot \left( w_i(t) \cdot \frac{e}{f(t)} + \beta \right) + (1 - \gamma) \cdot w_i(t) \]

Old window size
PowerTCP control law

\[ w_i(t + \delta t) = \gamma \cdot \left( w_i(t) \cdot \frac{e}{f(t)} + \beta \right) + (1 - \gamma) \cdot w_i(t) \]

MIMD based on Power
(Multiplicative increase - multiplicative decrease)
PowerTCP control law

$$w_i(t + \delta t) = \gamma \cdot \left( w_i(t) \cdot \frac{e}{f(t)} + \beta \right) + (1 - \gamma) \cdot w_i(t)$$

Additive increase
PowerTCP control law

\[ w_i(t + \delta t) = \gamma \cdot \left( w_i(t) \cdot \frac{e}{f(t)} + \beta \right) + (1 - \gamma) \cdot w_i(t) \]

Exponential Weighted Moving Average (EWMA)
PowerTCP feedback

Power is measured via Inband Network Telemetry (INT)

- Queue lengths
- Timestamps
- Tx bytes
- Bandwidth
PowerTCP without switch support

- Power can be measured via delay signal
PowerTCP without switch support

- Power can be measured via delay signal

\[
\Gamma = b^2 \times \theta \times (\dot{\theta} + 1)
\]

- Power
- Voltage
- Current

- RTT
- RTT gradient
Evaluation - Incast

![Graph showing throughput and queue length over time. The x-axis represents time in milliseconds (ms), the y-axis represents throughput in Gbps, and the legend shows queue length in KB. The graph highlights an initial spike in throughput followed by a steady decrease.]
Evaluation - Incast

Throughput (Gbps)

0 1 2 3 4

Time (ms)

Queue length (KB)

Throughput (Gbps)

0 1 2 3 4

Time (ms)

Delay-PowerTCP
Evaluation - Incast

Throughput (Gbps)

Queue length (KB)

Time (ms)
Evaluation - Incast

Throughput (Gbps)

Queue length (KB)

Time (ms)

Throughput (Gbps)

Queue length (KB)

Time (ms)
Evaluation - Incast

Throughput (Gbps)

Queue length (KB)

Time (ms)

PowerTCP

HOMA
Evaluation - Fairness & Stability

Throughput (Gbps) vs. Time (s)
Evaluation - Fairness & Stability

Throughput (Gbps) vs. Time (s)

PowerTCP

HOMA
Evaluation - Fairness & Stability

**PowerTCP**

Throughput (Gbps)

Time (s)

**Delay-PowerTCP**

Throughput (Gbps)

Time (s)

**HOMA**

Throughput (Gbps)

Time (s)
Evaluation - Fairness & Stability

PowerTCP

Delay-PowerTCP

Throughput (Gbps)

Time (s)

HOMA

TIMELY

Throughput (Gbps)

Time (s)
Evaluation - Workload

PowerTCP | HPCC | Delay-PowerTCP

![Graph showing 99.9 pct FCT slowdown vs load (%)]
Evaluation - Workload

- PowerTCP
- HPCC
- Delay-PowerTCP

**Short flows**

**Long flows**

99.9-pct FCT slowdown vs. load (%)
Evaluation - Reconfigurable Networks

High-bandwidth Circuit

Throughput

Queue length (KB)

Time (Base-RTTs)
Evaluation - Reconfigurable Networks

High-bandwidth Circuit

Throughput (Gbps)

reTCP

High Throughput

High latency

HPCC

Low Throughput

Low latency
Evaluation - Reconfigurable Networks

High-bandwidth Circuit

Throughput (Gbps)

PowerTCP
High Throughput
Low latency

reTCP
High Throughput
High latency

HPCC
Low Throughput
Low latency

Time (Base-RTTs)
Conclusion

- Existing CC are fundamentally limited to a single dimension
- Power is an interesting and provably good measure for CC
- PowerTCP: a novel control law based on Power
- Improves FCTs for short flows and even for long flows
Thank you