CodedBulk: Inter-Datacenter Bulk Transfers Using Network Coding

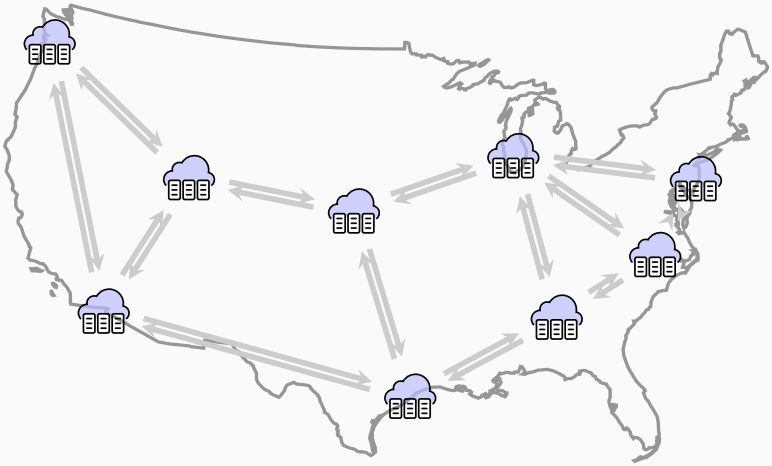
Shih-Hao Tseng[†]

joint work with Saksham Agarwal[†], Rachit Agarwal[†], Hitesh Ballani[‡], and Ao (Kevin) Tang[†]

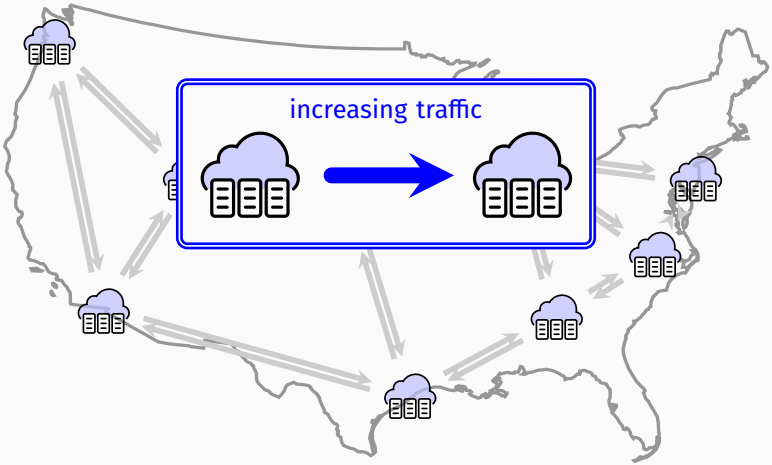
April 12, 2021

[†]Cornell University
[‡]Microsoft Research

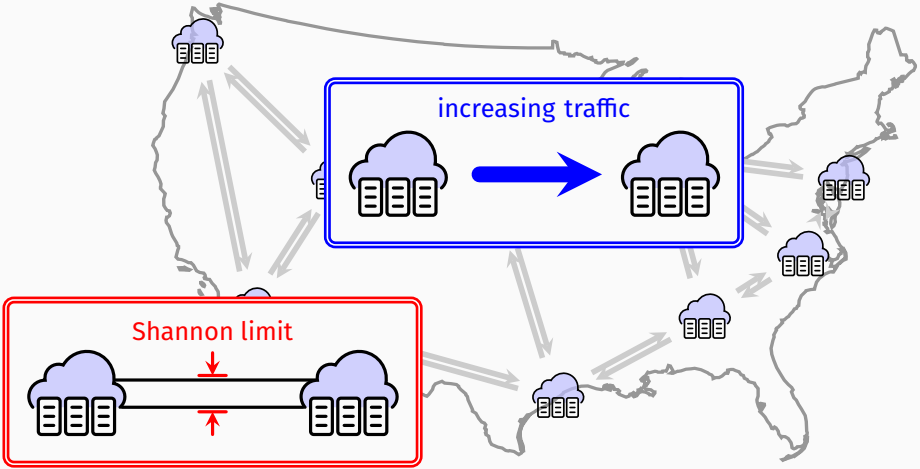
Inter-Datacenter WAN Traffic Continues to Increase



Inter-Datacenter WAN Traffic Continues to Increase

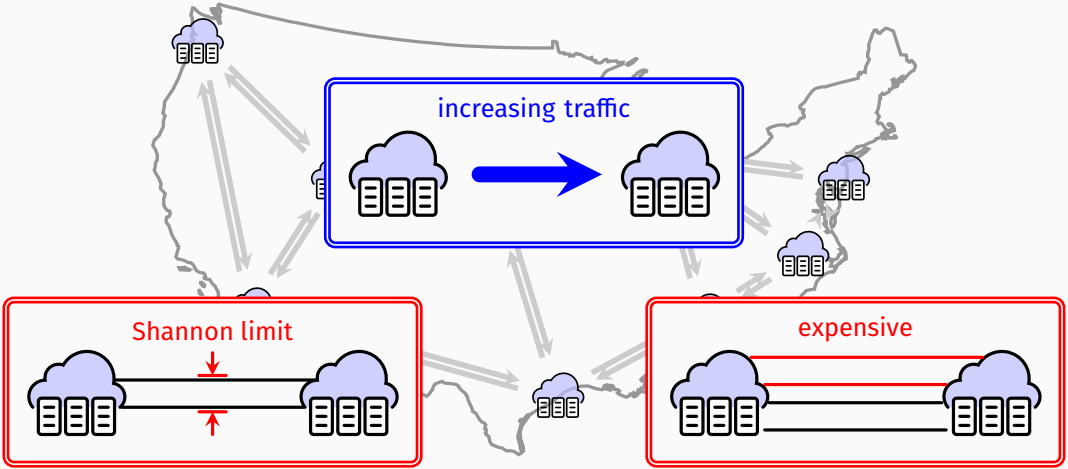


Inter-Datacenter WAN Traffic Continues to Increase



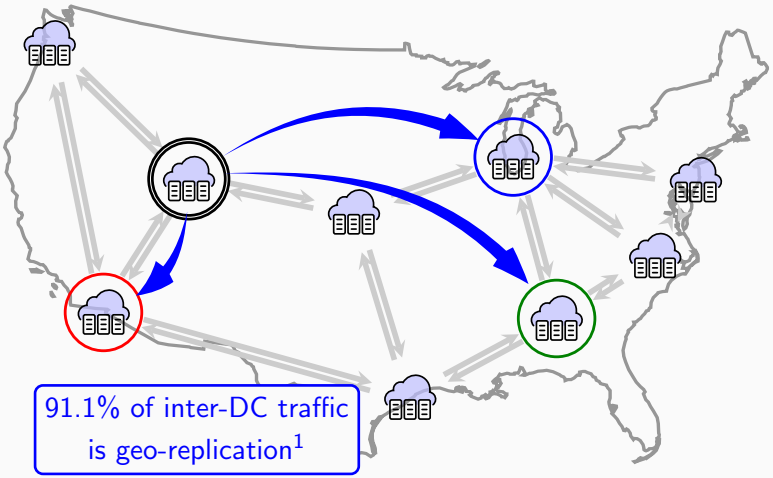
Meeting increasing inter-datacenter traffic demand is becoming increasingly hard/expensive

Inter-Datacenter WAN Traffic Continues to Increase



Meeting increasing inter-datacenter traffic demand is becoming increasingly hard/expensive

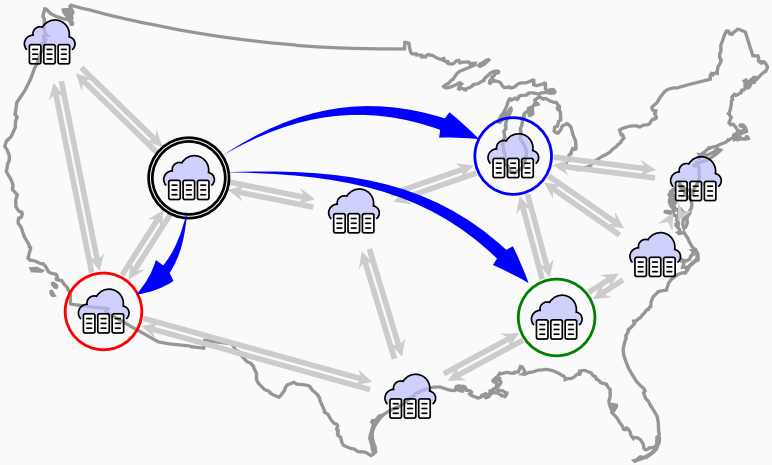
Most Inter-DC WAN Traffic is Bulk Transfers



Bulk transfer: Replication of large files from a source DC to multiple destination DCs

[1] Zhang et al., "BDS: A Centralized Near-Optimal Overlay Network for Inter-Datacenter Data Replication," EuroSys 2018.

Inter-DC WAN Bulk Transfers



Classical multicast problem: given a network topology, what is the maximum possible throughput for bulk transfers?

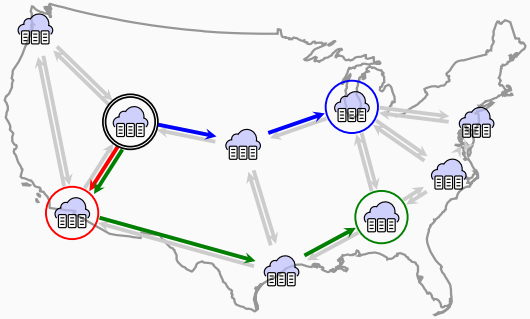
Existing Solutions for the Multicast Problem

1. Single path per destination
2. Multiple paths per destination
3. Steiner arborescence based solutions
4. Network coding

Let's understand them using simple examples

Multicast via Single-Path per Destination

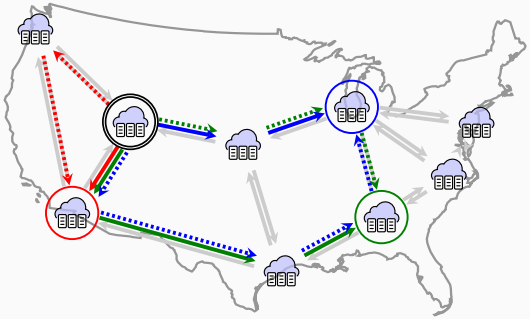
- Idea: Source transfers data to each destination using a single path



- Limitation: *Far from optimal* due to suboptimal bandwidth utilization

Multicast via Multiple Paths per Destination

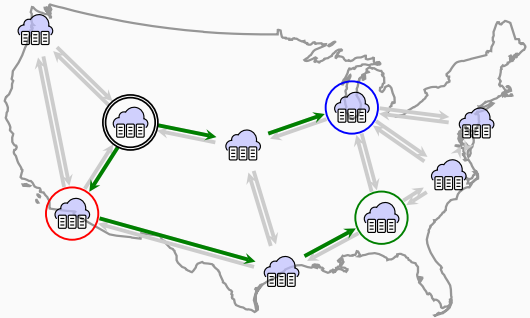
- Idea: Source transfers data to each destination using multiple paths independently computed



- Limitation: *Far from optimal* due to same data being transferred across overlapping paths.

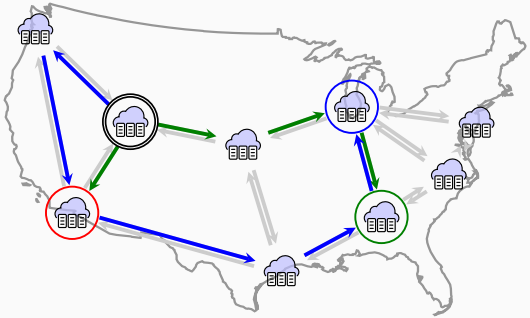
Multicast Through Steiner Arborescences to All Destinations

- Idea: Intermediate nodes can mirror/forward data to other destinations
  - Compute and "pack" multiple arborescences
  - One unit of data transferred per arborescence



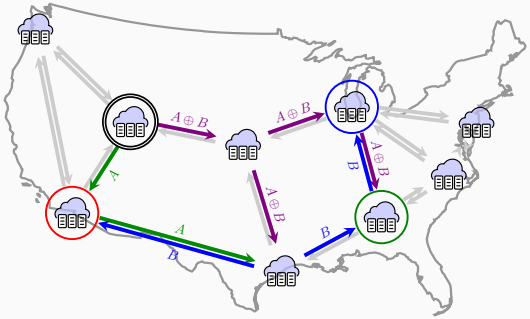
Multicast Through Steiner Arborescences to All Destinations

- Limitations:
  1. NP-hard to compute optimal (least-link) Steiner arborescence
  2. Approximation algorithms use bandwidth inefficiently
  3. Optimal (throughput) solution requires packing suboptimal (non-least-link) Steiner arborescences



Network Coding: Optimal Theoretical Throughput

- Idea: Allow intermediate nodes to perform computations on incoming data before forwarding



- Benefits:
  - Guarantees optimal throughput for bulk transfers
  - Solutions can be computed efficiently

CodedBulk

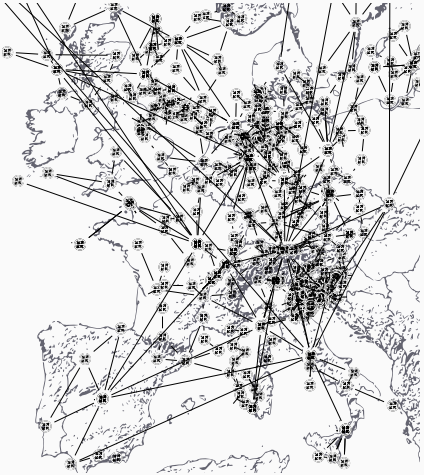
- End-to-end system for inter-DC bulk transfers
- Uses network coding to achieve near-optimal throughput
- On testbeds comprising $9 - 13$ geo-distributed DCs, 1.2 – $2.5\times$ higher throughput compared to state-of-the-art mechanisms that do not perform network coding
- No changes in the underlying transport/network layers

CodedBulk Contributions

- Use of network coding to wired networks has faced several *pragmatic* and *fundamental* challenges.
- CodedBulk alleviates the pragmatic challenges by exploiting unique properties of inter-DC WAN networks.
- CodedBulk resolves the fundamental challenges by using a custom-designed hop-by-hop flow control mechanism.

Pragmatic Challenges

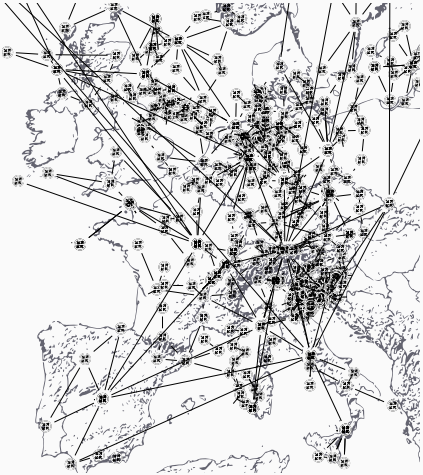
Challenges:



Pragmatic Challenges

Challenges:

lack of resources

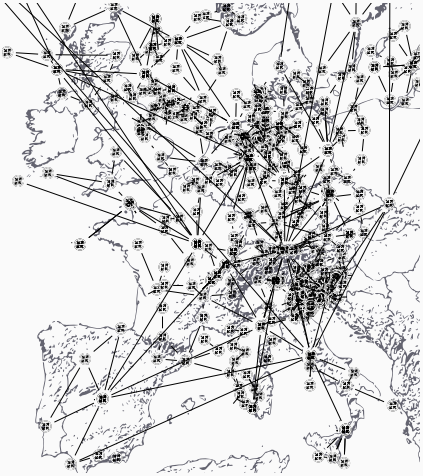


Pragmatic Challenges

Challenges:

lack of resources

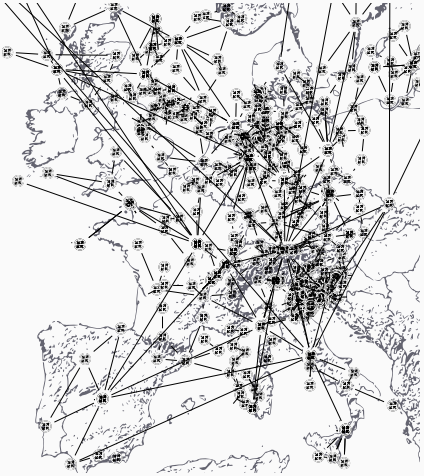
computation complexity



Pragmatic Challenges

Challenges:

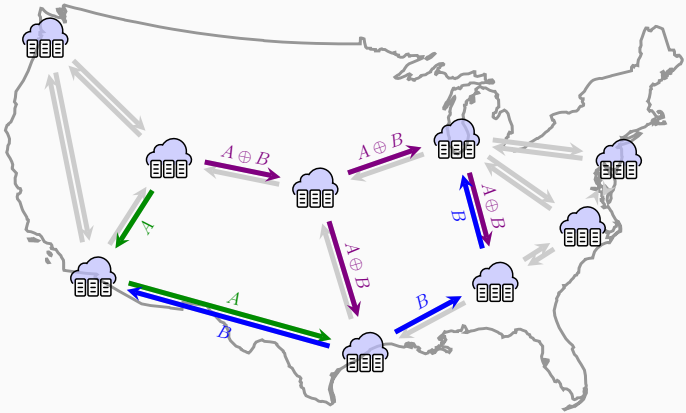
- lack of resources
- computation complexity
- lack of centralized control



CodedBulk Exploits Inter-DC WAN Properties to Alleviate Pragmatic Challenges

Challenges:

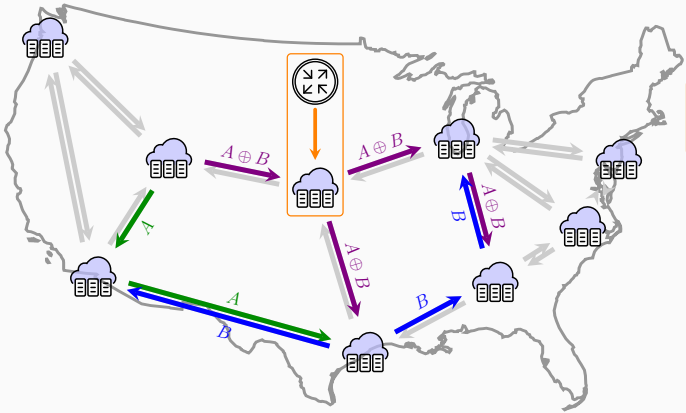
- lack of resources
- computation complexity
- lack of centralized control



CodedBulk Exploits Inter-DC WAN Properties to Alleviate Pragmatic Challenges

Challenges:

- lack of resources
- computation complexity
- lack of centralized control



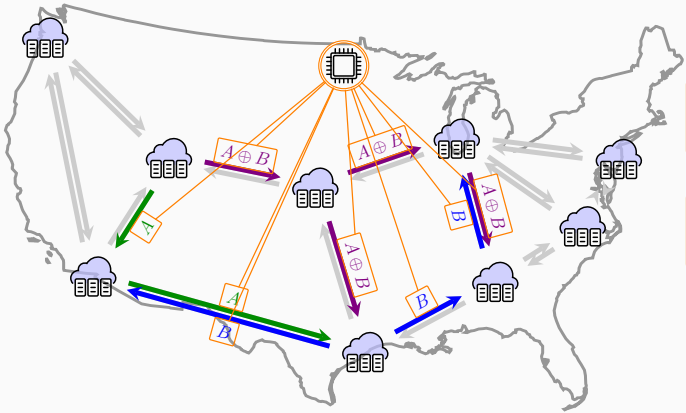
Opportunities:

- plenty of resources

CodedBulk Exploits Inter-DC WAN Properties to Alleviate Pragmatic Challenges

Challenges:

- lack of resources
- computation complexity
- lack of centralized control



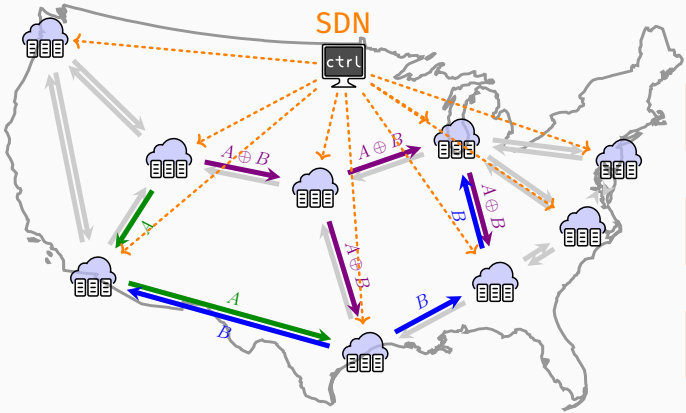
Opportunities:

- plenty of resources
- small sized network

CodedBulk Exploits Inter-DC WAN Properties to Alleviate Pragmatic Challenges

Challenges:

- lack of resources
- computation complexity
- lack of centralized control

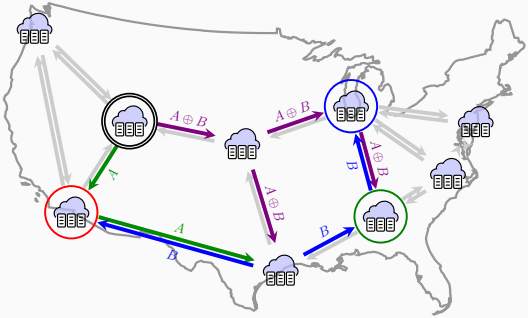


Opportunities:

- plenty of resources
- small sized network
- SDN centralized control

Fundamental Challenges – Asymmetric Link Problem

Asymmetric links invalidate Network coding literature



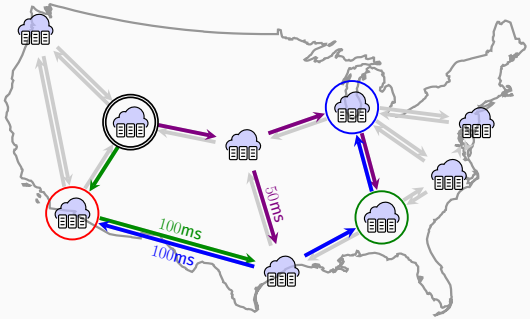
Fundamental Challenges – Asymmetric Link Problem

Asymmetric links due to

- non-uniform link delay

Network coding literature

- assumes packets arriving at each node at the same time



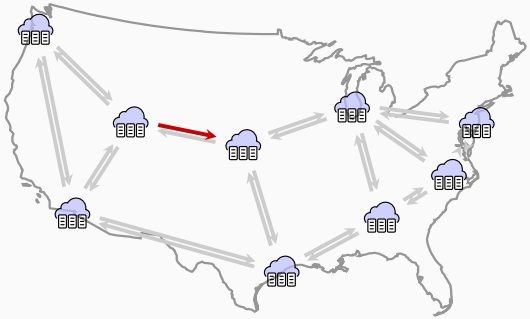
Fundamental Challenges – Asymmetric Link Problem

Asymmetric links due to

- non-uniform link delay
- interactive traffic

Network coding literature

- assumes packets arriving at each node at the same time
- considers no interactive traffic



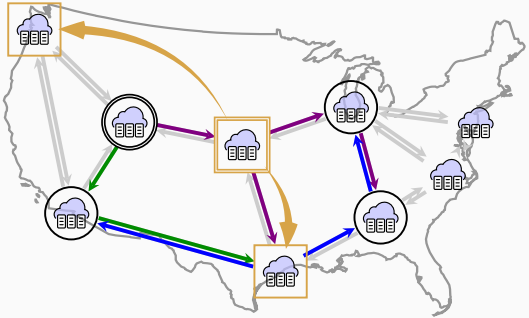
Fundamental Challenges – Asymmetric Link Problem

Asymmetric links due to

- non-uniform link delay
- interactive traffic
- other bulk transfers

Network coding literature

- assumes packets arriving at each node at the same time
- considers no interactive traffic
- considers only a single bulk transfer



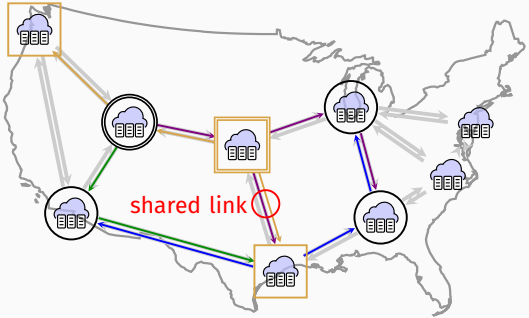
Fundamental Challenges – Asymmetric Link Problem

Asymmetric links due to

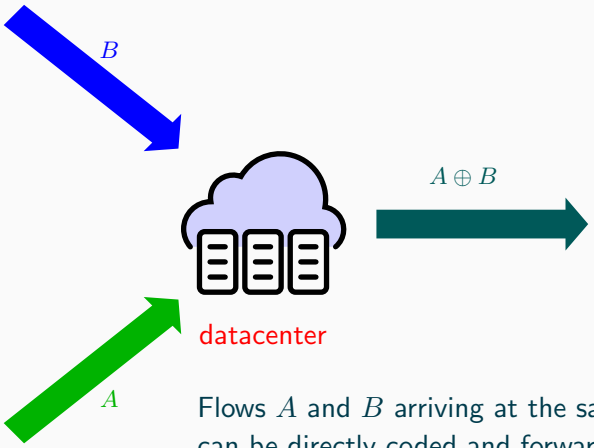
- non-uniform link delay
- interactive traffic
- other bulk transfers

Network coding literature

- assumes packets arriving at each node at the same time
- considers no interactive traffic
- considers only a single bulk transfer

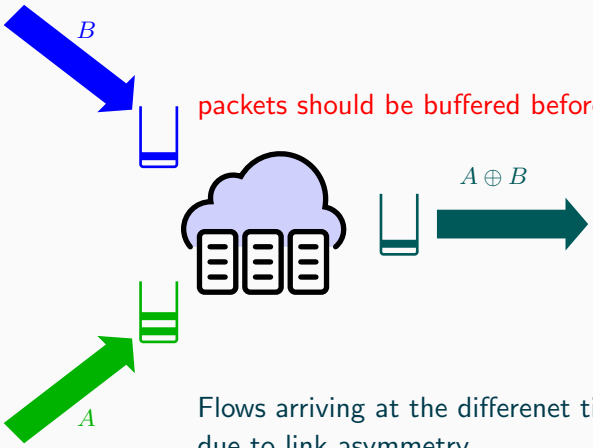


Use Buffers to Handle Asymmetric Links

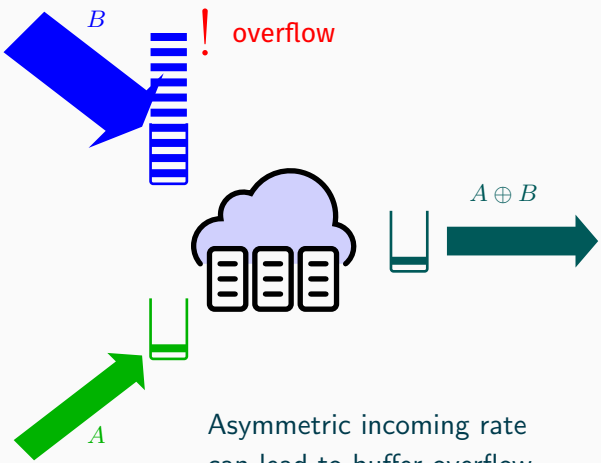


Flows A and B arriving at the same time and rate can be directly coded and forwarded

Use Buffers to Handle Asymmetric Links

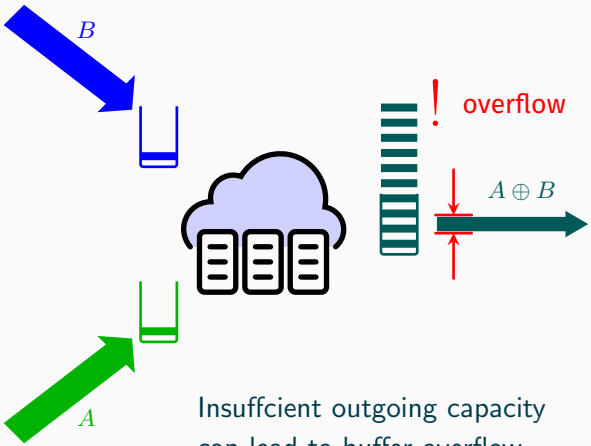


Buffer Overflow and Hop-by-Hop Flow Control



Asymmetric incoming rate
can lead to buffer overflow

Buffer Overflow and Hop-by-Hop Flow Control

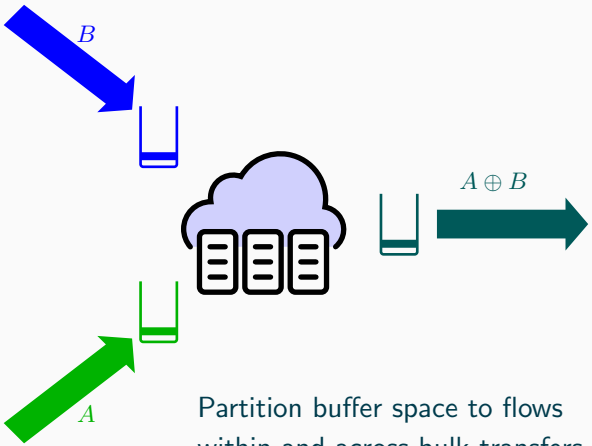


Insuffcient outgoing capacity
can lead to buffer overflow

Why New Customized Hop-by-hop Flow Control?

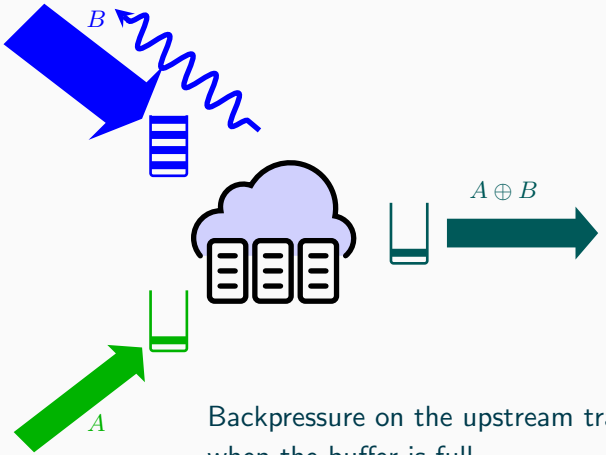
- Traditional hop-by-hop flow control mechanisms operate on individual flows
  - each downstream flow depends on precisely one upstream flow
- Network-coded flows require multiple upstream flows to be encoded at intermediate nodes
  - each downstream flow may depend on multiple upstream flows
  - each upstream flow may impact multiple downstream flows
- CodedBulk employs customized hop-by-hop flow control
  - all incoming flows that need to be coded converge to the same rate
  - all flows on different bulk transfers converge to max-min fair rate
  - the network is deadlock-free

CodedBulk Hop-by-Hop Flow Control



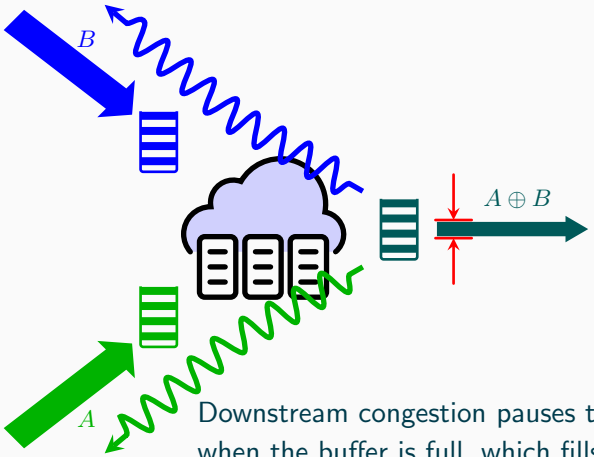
Partition buffer space to flows within and across bulk transfers.

CodedBulk Hop-by-Hop Flow Control



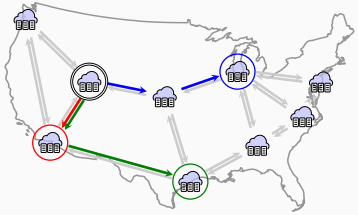
Backpressure on the upstream traffic when the buffer is full

CodedBulk Hop-by-Hop Flow Control

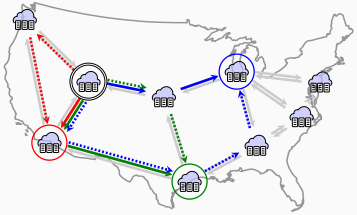


Downstream congestion pauses the coding when the buffer is full, which fills the upstream buffers and backpressures

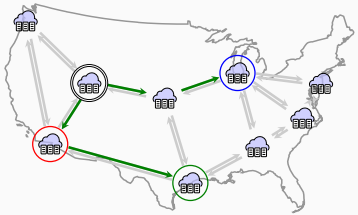
Evaluation – Methods to Compare



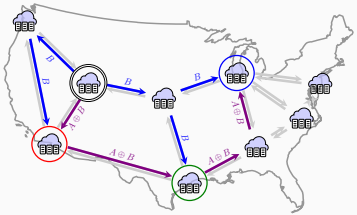
(a) Single-Path



(b) Multi-Path



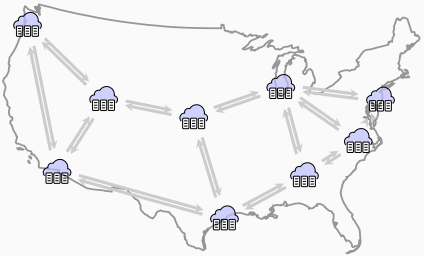
(c) Steiner Arborescence



(d) CodedBulk

Evaluation – Setup

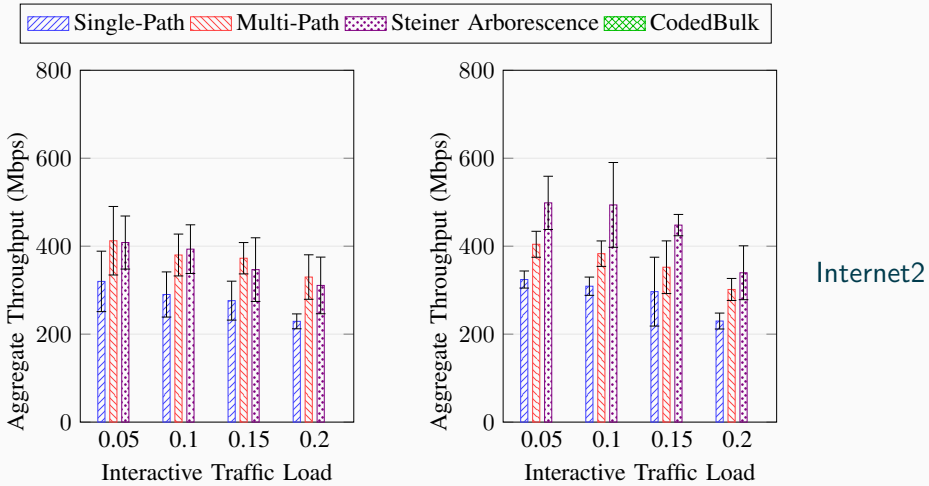
- no simulation; implementations on real cloud testbeds
- topologies: 9-node Internet2 and 13-node B4
- baseline: 6 bulk transfers, 3 destinations each, under 0.1 interactive traffic load
- varying interactive traffic load level, number of concurrent bulk transfers, number of destinations per bulk transfer



9-node Internet2

13-node B4

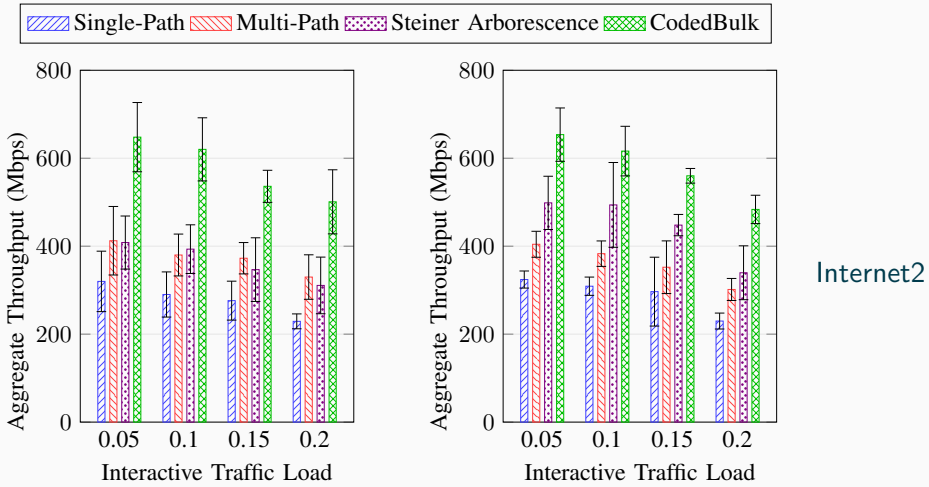
Varying Interactive Traffic Load



Steiner arborescence outperforms single-path as it avoids overlapping paths

It outperforms multi-path when the path diversity is low (such as Internet2)

Varying Interactive Traffic Load



CodedBulk significantly outperforms all mechanisms across varying interactive traffic load

CodedBulk

Inter-DC Bulk Transfers Using Network Coding

open-sourced at

https://github.com/synergy-cornell/codedbulk

Shih-Hao Tseng
shtseng@caltech.edu
https://shih-hao-tseng.github.io/