THE OHIO STATE UNIVERSITY

MURI Meeting
November 21, 2016

Analysis and Design of Complex Systems with Multivariate Heavy Tail Phenomena

Ness B. Shroff

Depts. ECE & CSE, The Ohio State University

E-mail: shroff@ece.osu.edu

Collaborators: R. Srikant (UIUC), Joohyun Lee (OSU), Kyunghan Lee (UNIST), Jaemin Jo (UNIST), Eujin Jeong (UNIST), Irem Korpulu (OSU), Yoora Kim (UNIST), Yin Sun (OSU), Prasun Sinha (OSU), Kyu Han Kim (HP Labs), C. Emre Koksal (OSU)

Overall Outline

- Overview of Research Directions
  - Analyzing statistical metrics of different mobility models (Lévy, Explore return model, etc.): first exit time, contact time…
  - Exploiting opportunities (node mobility, channel variation, predictability) for resource allocation in wireless nets.
  - Modeling and Analyzing Influence Propagation on Evolving Social Networks
  - Developing scheduling algorithms for data center/cloud computing systems
  - Developing context-aware scheduling for mobile devices
- Optimal coding/scheduling for clouds & wireless networks

Lévy mobility

THE OHIO STATE UNIVERSITY

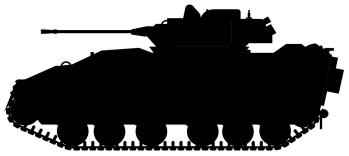
Analyzing statistical metrics of Lévy mobility: first exit time, contact time, and inter-contact time

- Lévy mobility is a class of random walks that are observed in ecology for different animal species (including humans)
- Location is characterized by multivariate heavy tails in two or more dimensions
- Characterized First exit time for Lévy flight model in in $\mathbf{R}^n$ [Journal of Applied Probability, vol. 52, no. 2, 2015]

- Ongoing work:
  - Extension to Explore/Return model for more detailed human mobility modeling (FET analysis completed)
  - Extension to directional drift model
  - First meeting time and inter-contact times

Impact: Deeper Understanding of Mobility Patterns

- For troops' mobility model
  - Estimate time until troops leave a (potentially hazardous) region
  - Characterize the time and frequency of contacts
  - Analyze the total moving distance until devices carried by troops reach a place for network access (e.g. WiFi area) or energy replenishment (e.g. battery charge)


- Mobile ad-hoc networks
  - Contact time critical in determining the delay and capacity of a network
  - Important in choosing various scheduling and forwarding algorithms
  - Inter-contact time: end-to-end delay in MANET

Wireless Resource Allocation/Control

Exploiting various opportunities for resource allocation in wireless networks

- Mobile data offloading
  - Cellular networks are highly constrained (esp. in military settings)
  - Use WiFi LANs, mmWave or direct contact opportunities for delivering data originally targeted for cellular networks (WiFi and cellular network interworking)
  - Design of data off-loading schemes: A coupled queueing problem with bi-variate heavy tailed on/off service time distribution
    - ON and OFF periods of WiFI service are Pareto and dependent
  - Analyzed reneging probability (probability of not using WiFi) and expected waiting time (ACM Mobihoc 2014) in these systems
- Ongoing Work:
  - Age of Information in wireless networks with unreliable channels

Data Centers and Cloud Computing/Storage

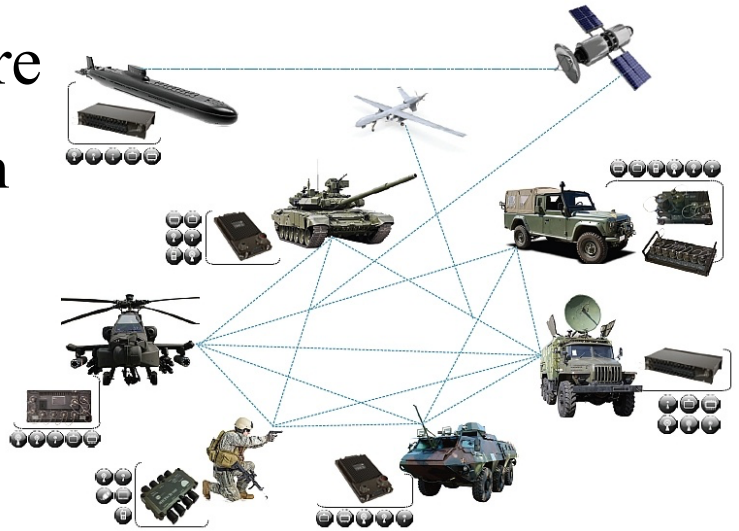
Developing scheduling algorithms for minimizing delay in data center, cloud computing, and storage systems

- Analysis and design of MaP/Reduce type of scheduling algorithms with multi-variate heavy tailed dependency for minimizing flow time
  - MaP/Reduce is ubiquitous programming paradigm for data center systems
  - # of Map tasks and length of reduce tasks are heavy tailed and dependent
- Proved that the flow time minimization problem is strongly NP-hard and does not yield a finite competitive ratio [IEEE INFOCOM'13]
- Developed 2-approximation probabilistic competitive ratio pre-emptive scheduler that is independent of the nature of job size distributions [IEEE INFOCOM'13]
- Low-latency algorithms in the large-system limit for both pre-emptive and non-pre-emptive schedulers that are robust to heavy tails and dependencies between Map and Reduce [IEEE INFOCOM'15]
- Ongoing Work (in Cloud systems)
  - Load balancing & scheduling: Optimal-latency computing/data retreival

Impact: Reliable and Faster Networked Systems

- Heterogeneous communication networks (e.g., satellite, cellular networks, and MANETs)
  - Load balancing among communication networks and minimizing the delay
  - Obtained high throughput and low delay in heterogeneous environments

- Better utilization of cloud infrastructure
  - Faster computation/data retrieval from clouds
  - More tasks can be offloaded to the data center/clouds
    - Power/energy efficiency
    - Better control of autonomous vehicles/drones



Scheduling in Mobile Computing Devices

Context Aware Application Scheduling for Increasing Battery Lifetime and Improving Application Response Times in Smartphones

- Analyzed device usage patterns in over 100 smart mobile users
  - The launching probabilities follow Zipf's law
  - The inter-running and running times show multi-variate heavy tailed dependency
- Revealed context-dependency on time, location, and previous actions
- Showed that the scheduling problem becomes constrained submodular minimization which is NP-hard
- Developed a local optimal scheduling algorithm
- Showed substantial improvements in both start-up latency and energy consumption [ACM UbiComp'16] by exploiting context information
- Ongoing Work
  - Non-stationary combinatorial Multi-Armed Bandit problem

Impact: Longer-lasting and responsive mobile devices

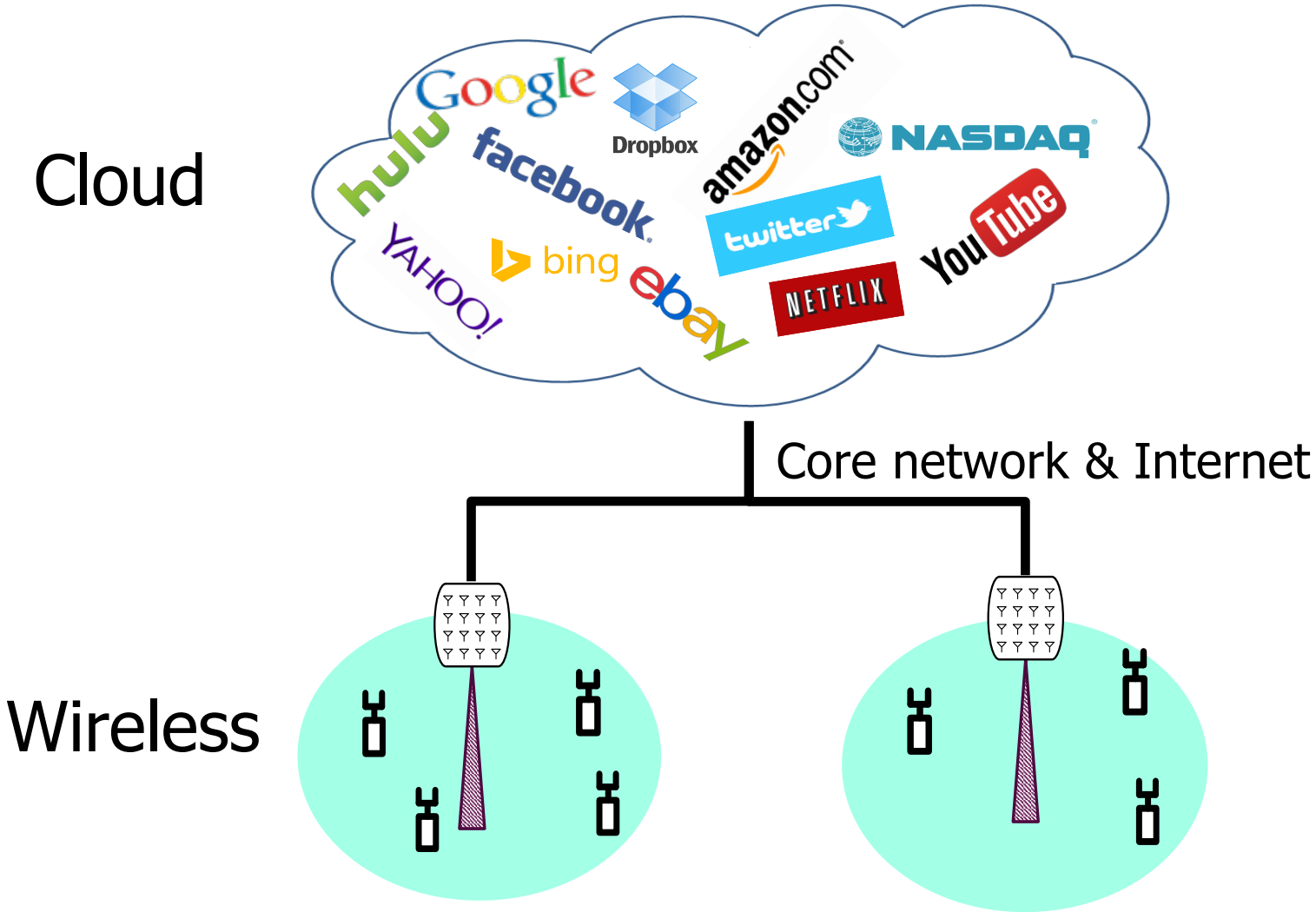
- Understanding device usage patterns of military entities
  - Statistical analysis about inter-running and running times
  - Context-dependency on environments (types of battle fields/missions, time, location, and previous actions)

- Enhancing mobile computing devices of military entities
  - Resource-efficient scheduling algorithm
  - Increasing the lifetime
  - Reducing the latency of tasks

Latency-Optimal Coding Control and Scheduling for Clouds and Wireless Networks

Collaborators: Yin Sun (OSU), C. Emre Koksal (OSU)

Cloud and Wireless Services are Ubiquitous



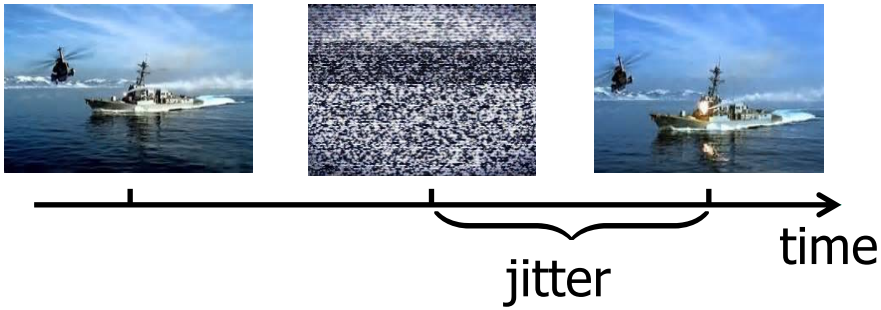
Form the basis of most mobile applications…

A big concern: Low interactive latency

Different Types of Latencies

- Delay
  - Data downloading

- Jitter
  - Video streaming



- Deadline
  - Real-time control

Different latencies require different control schemes

Key Question

How do we minimize the delays in clouds and wireless networks?

Outline

THE OHIO STATE UNIVERSITY

I. **Queueing model for coding in clouds and wireless networks**

II. **Problem formulation**

- Delay metrics, challenges, and contributions

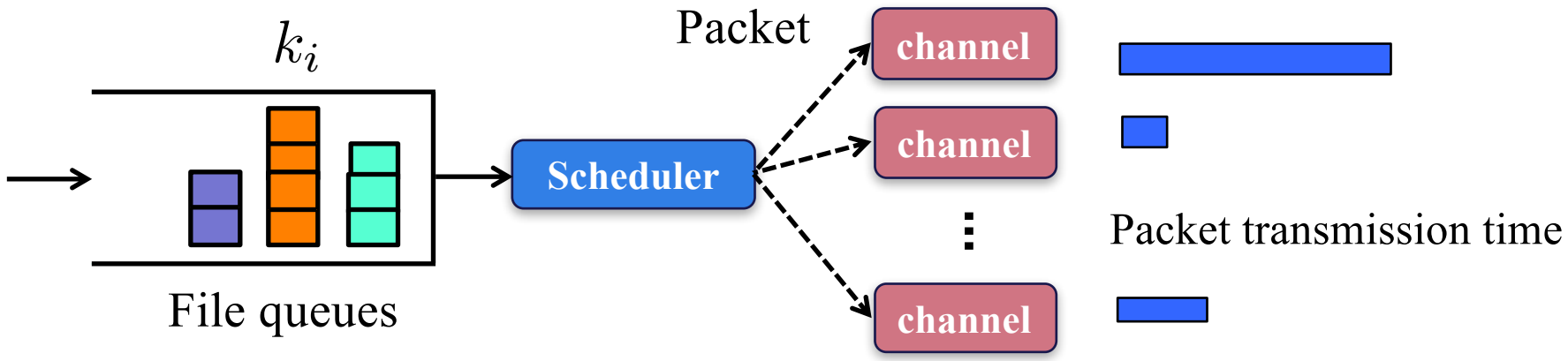
III. **Design of low-latency codes**

- Coding principle: Heavy tail vs. short tail
- Scheduling principle

IV. **Performance evaluation**

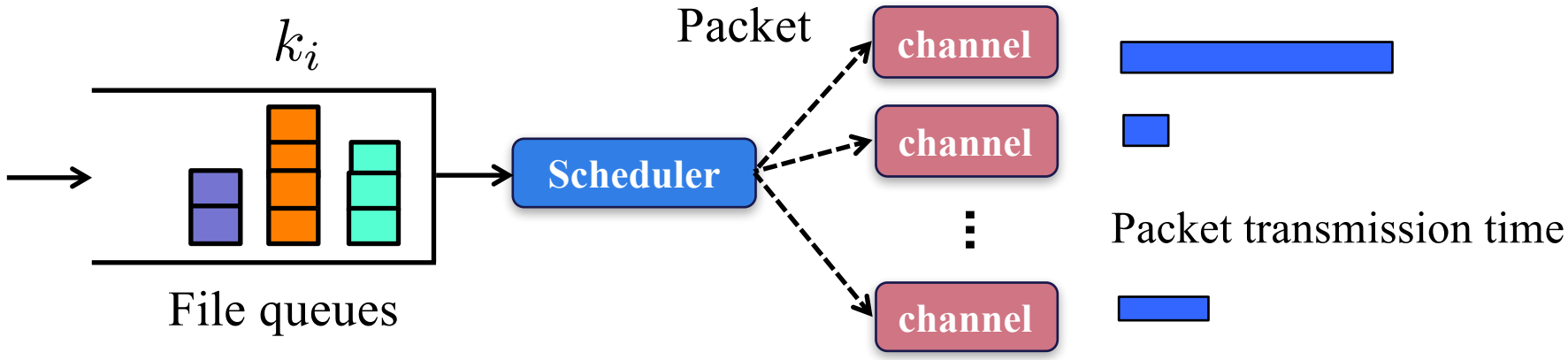
- Close to the optimum latency
- Latency reduction: Orders of magnitude

System Model: Wireless Networks



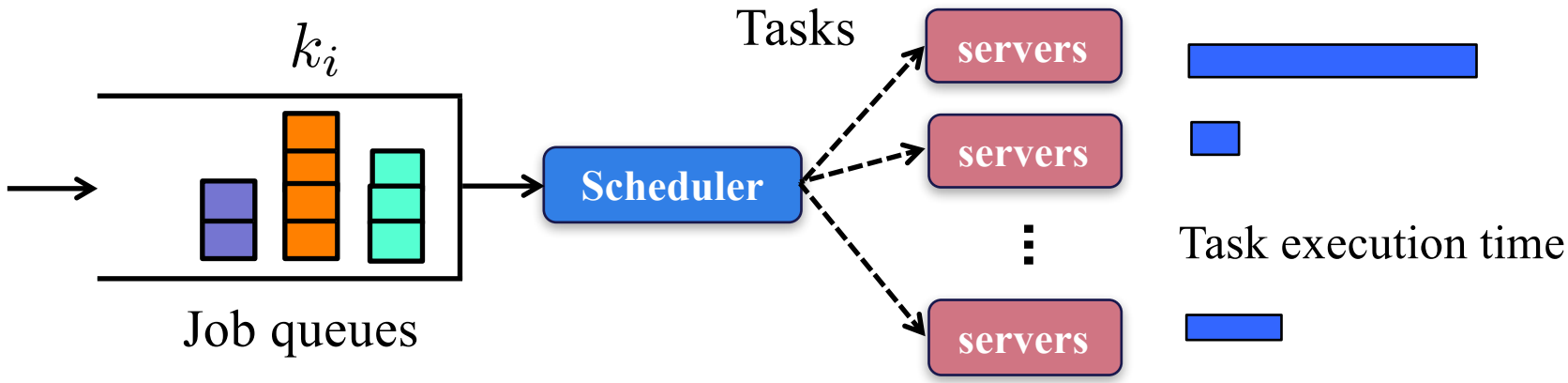
- Each node can be modeled by a **multi-server queueing system**
  - m "servers" (channels): could be different from node to node.
- Arbitrary arrivals: "Files" (message/video frames…) arrive sequentially at each node according to a general arrival process
  - Could be non-stationary non-ergodic arrivals (including multi-variate dependencies)
- Arbitrary file size: Each file i has k_i unit-size packets to be transmitted over the channels
- Allow channels to have different data rates and channel quality
  - Packet transmission times could follow heavy-tail distributions
  - Independent channels: theoretical research, correlated channels: simulations

System Model: Low-latency Coding



- Each packet can be transmitted using ARQ/H-ARQ/Rateless codes...
  - We do not assume that CSI is perfectly known ➔ packet transmission times are random and not known a-priori, and can differ from channel to channel
- Code across channels (to reduce delay):
  - Using an (n_i, k_i) **MDS code** to generate n_i packets and sent over the channels
  - When k_i packets are received, the file can be decoded and other transmissions cancelled
    - Replication corresponds to an $(n, 1)$ MDS code

System Model: Cloud Computing



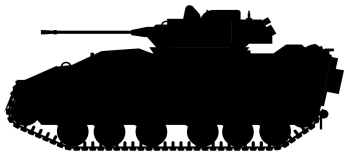
- Computing Jobs arrive to a scheduler bringing with them many tasks
- The tasks will be assigned to different servers
- Arbitrary job arrival process and arbitrary job sizes
- Task execution times follow correlated, possibly heavy-tailed distributions
- Coded computing: E.g., matrix multiplication $y = Ab$ where $A = [A_1, A_2]$
  - Divide into 2 tasks: $A_1 b$, $A_2 b$
  - (3,2) MDS code: define $A_3 = A_1 + A_2$

    Assign $A_1 b, A_2 b, A_3 b$ to 3 servers

    When two servers return their results, the job is completed

Relation to the project

- Computing/communications are essential in military environments
  - E.g., Soldiers, Tanks, Aircrafts
  - **Low latency** needed for military operations (or other applications soldiers may be interested in)


- **Goal:** Design ***latency-optimal coding & scheduling over multiple channels (servers)*** under different system loads and stationary & non-stationary network dynamics
  - Arbitrary arrivals (e.g., heavy tailed dependent inter-arrival times)
  - Light and heavy tailed service times
- **Outcome:** Simple Coding and scheduling principles can be applied to reduce latency in many military comm. & computing systems

System Model: Delay Metrics

- n arriving files (or jobs), n is either finite or infinite
- File i arrives at time a_i, is delivered at time $C_i(\pi)$ in policy π
- Delivery time vector: $\mathbf{C}(\pi) = [C_1(\pi), \ldots, C_n(\pi)]$
- Soft deadline:
  - Due time d_i: Is the promised file delivery time
  - Delivery after d_i is allowed, but with a penalty called lateness
  - Lateness: $C_i(\pi) - d_i$ (related to jitter) Delay: $C_i(\pi) - a_i$
- **Average delay:** $$D_{\text{avg}}(\mathbf{C}(\pi)) = \frac{1}{n}\sum_{i=1}^{n}[C_i(\pi) - a_i]$$

  Time difference between file arrival and delivery
- **Maximum lateness:** $$L_{\max}(\mathbf{C}(\pi)) = \max_{i=1,\ldots,n}[C_i(\pi) - d_i]$$
- **Maximum delay:** $$D_{\max}(\mathbf{C}(\pi)) = \max_{i=1,\ldots,n}[C_i(\pi) - a_i]$$

  Special case when $d_i = a_i$
- **Note:** Each delay metric is a **random variable**

Delay Metrics (cont'd)

More generally, we can consider the classes of delay metrics that are increasing and Schur-convex in the delay or lateness vectors. For example,

- **Average square lateness/tardiness:**

$$T_{\text{ms}}(\mathbf{C}(\pi)) = \frac{1}{n}\sum_{i=1}^{n}\max[C_i(\pi) - d_i, 0]^2$$

- **Average square delay:**

$$D_{\text{ms}}(\mathbf{C}(\pi)) = \frac{1}{n}\sum_{i=1}^{n}[C_i(\pi) - a_i]^2$$

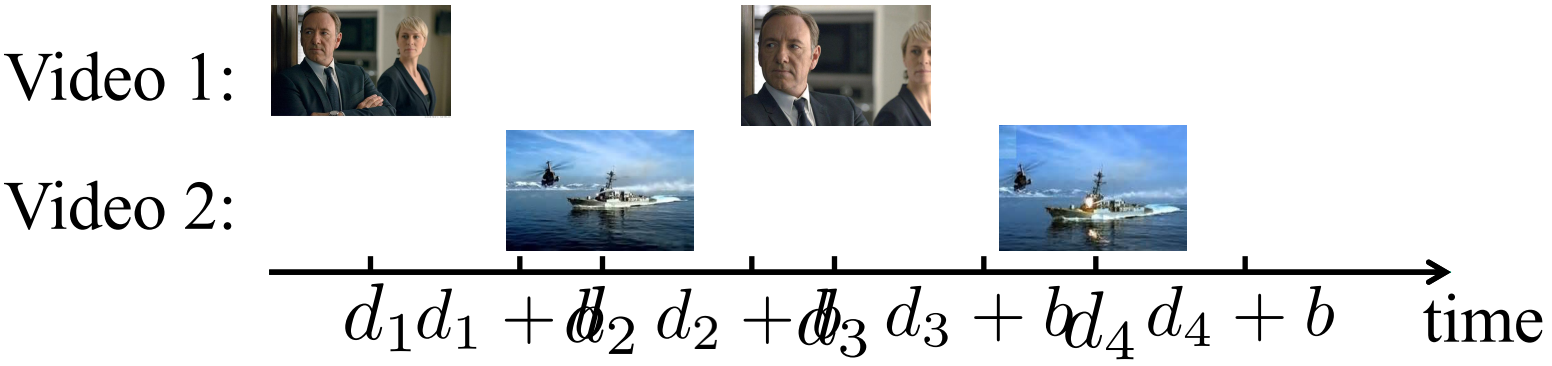
  - Note: avg. square delay is different from square of avg. delay

- In general, each delay metric can be expressed as

$$f(\mathbf{C}(\pi)) = f([C_1(\pi), \ldots, C_n(\pi)])$$

  - $f(\cdot)$ is an non-decreasing function

Example: Jitter Probability



- Consider several video streams with a total number of l packets
- Packet i is intended to be delivered at time d_i
- Buffering for b seconds to reduce jitter
- After that, Packet i is intended to be delivered at time $d_i + b$
- Jitter probability: $\Pr[\text{Jitter}] = \Pr[\text{At least one packet } i \text{ is delivered after } d_i+b]$

$$= \Pr\left[\cup_{i=1,\ldots,l}\{C_i(\pi) > d_i + b\}\right]$$

$$= \Pr\left[\max_{i=1,\ldots,l}[C_i(\pi) - d_i] > b\right]$$

$$= \Pr[L_{\max}(C(\pi)) > b]$$

Jitter probability = complementary CDF of maximum lateness

Goal

- **Ideally:** Minimize any moments and CCDF of these delay metrics over the set of all causal non-preemptive scheduling policies Π (including any MDS code).
- **Delay Optimality:**

Policy P is ***delay optimal in distribution*** for minimizing $f(\mathbf{C}(\pi))$, if for any file arrival parameters and any coding scheme,

$$\Pr[f(\mathbf{C}(P)) > x] = \min_{\pi \in \Pi} \Pr[f(\mathbf{C}(\pi)) > x], \forall\ x \geq 0.$$

where Π is the set of causal non-preemptive policies

- Difficult or unachievable even for mean average delay [Weiss'92'95], [Dacre, Glazebrook, Nino-Mora'99]
- **Goal:** Develop near-optimal policies in policy space Π

Near Delay Optimality

Policy P is near delay optimal in distribution if the delay metric (e.g. avg delay) of files starting transmission in policy P is stochastically smaller than the delay metric of files completing transmissions in any policy in Π.

- $\mathbf{V}(\pi) = [V_1(\pi), \ldots, V_n(\pi)], \mathbf{C}(\pi) = [C_1(\pi), \ldots, C_n(\pi)]$
- By definition, $\mathbf{V}(\pi) \leq \mathbf{C}(\pi)$

- **Delay optimality in distribution:**

$$\Pr[f(\mathbf{C}(P)) > x] = \min_{\pi \in \Pi} \Pr[f(\mathbf{C}(\pi)) > x], \forall\ x \geq 0.$$

- **Near delay optimality in distribution:**

$$\Pr[f(\mathbf{V}(P)) > x] \leq \min_{\pi \in \Pi} \Pr[f(\mathbf{C}(\pi)) > x], \forall\ x \geq 0.$$

That is, Policy P provides a lower bound by replacing $\mathbf{C}(P)$ with $\mathbf{V}(P)$

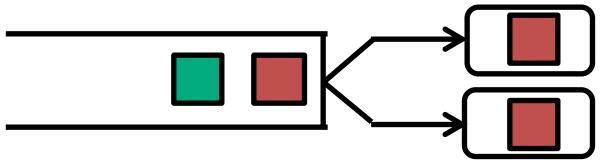
Challenges of (Near) Delay Optimality Without Coding

THE OHIO STATE UNIVERSITY

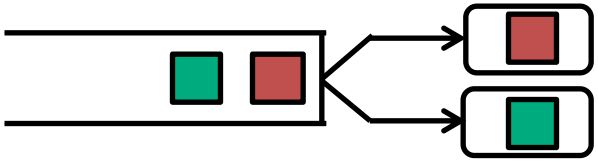
- Single-server queueing systems: Optimality is achievable
  - Deterministic scheduling (known CSI/packet completion time):
    - Average delay: SRPT [Schrage'68]
  - Stochastic scheduling:
    - Weighted average delay: c- μ rule [Smith'56], Gittins index [Gittins'79], Klimov's model [Klimov'74]
- Multi-server queueing systems: Optimality is notoriously difficult!!
  - Deterministic scheduling:
    - Average delay: NP hard [Leonardi, D. Raz'97]
  - Stochastic scheduling:
    - Average delay: O(m) sub-optimality gap [Weiss'92'95], [Dacre, Glazebrook, Nino-Mora'99]
    - Order optimal: Large system limits, heavy traffic limits, …
  - Delay optimality outside of such limiting regimes is open since 1960s
    - No limits in our formulation to capture short time horizons important for military.

Further Challenges with Coding

- Repetition code: complete the red packet faster but delay green packet

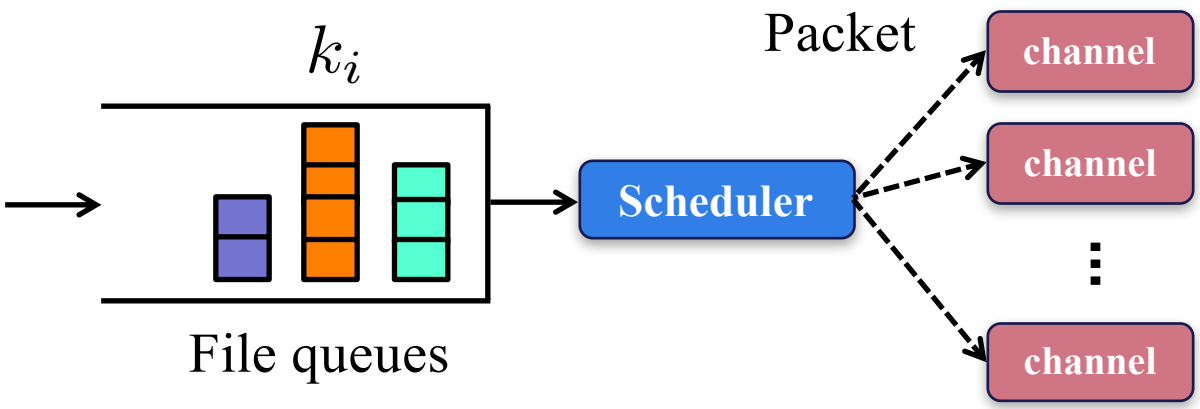


- No coding: complete the green packet earlier but delay red packet



- A fundamental dilemma

Our Contributions



- Independent Channels:

  For arbitrary number of files, file sizes, and arrival times, we have developed simple control policies that are near optimal in distribution for minimizing several classes of delay metrics among all causal policies.

  - **Note:** Our policies are also maximize the throughput

- Proof approach:

  sample-path ordering + coupling

  majorization

Key Design Principles

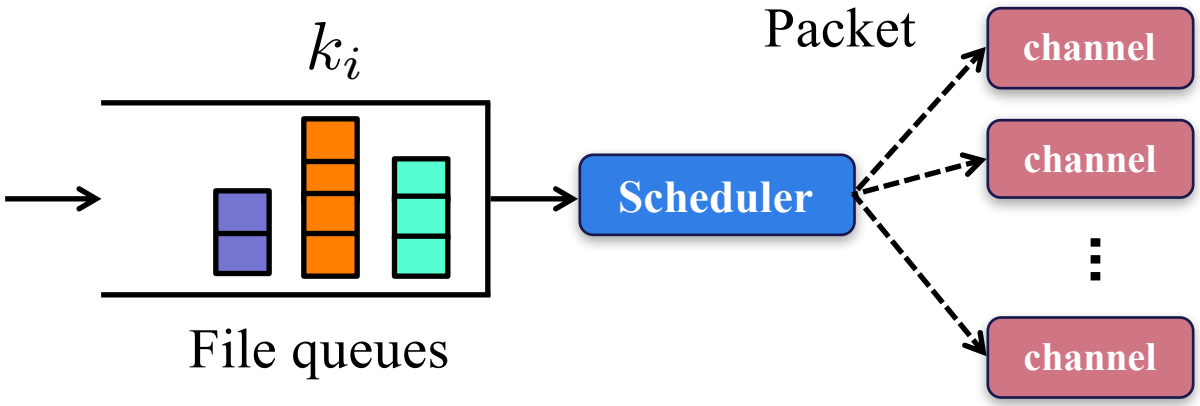
Whether to code or not and which code to use?

Repetition code

No code

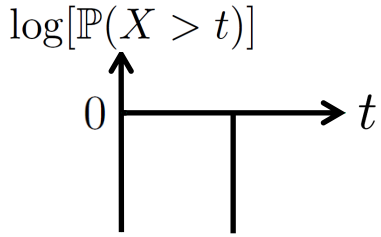
Which file to serve next?

- Small file first, First-Come First-Served, …



Intuition: When not to Code?

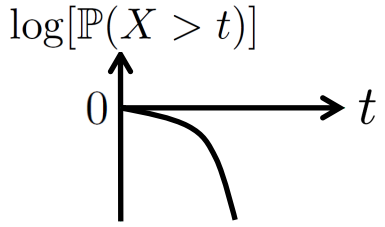
- Example 1: deterministic service time
  - Perfect CSI → no coding



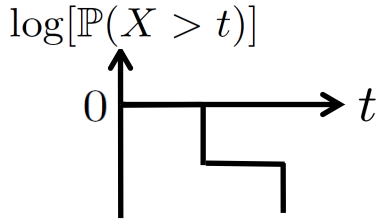
- Extended to New-Better-than-Used (NBU) distributions

$$\mathbb{P}(X > t) \geq \mathbb{P}(X > t + \tau | X > \tau) \text{ for all } t, \tau \geq 0$$

P{New packet takes more than t sec to send} $\geq$ P{Old packet takes more than another t sec to send}



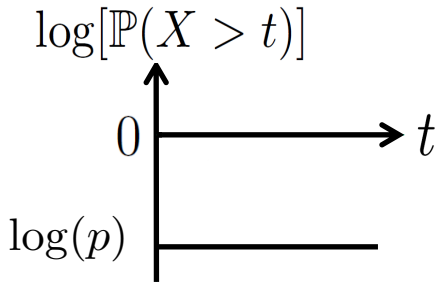
  - The remaining transmission time of an old packet is (probabilistically) shorter than the transmission time of a new packet
    - Ex. Using ARQ over binary erasure channels
  - **NBU:** Do not code across channels



geometric distribution

Intuition: When to Code?

- Example 2: instant, but fails with prob. p
  - mmWave Transmissions…
  - Replicate a packet to all channels

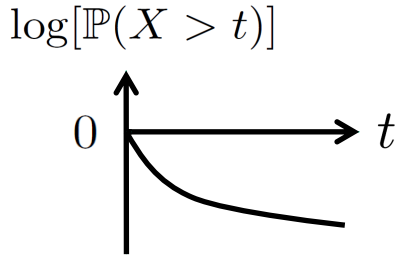


- Extended to New-Worse-than-Used (NWU) distributions

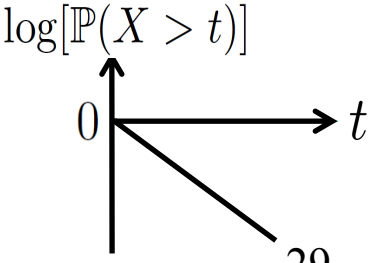
$$\mathbb{P}(X > t) \leq \mathbb{P}(X > t + \tau | X > \tau) \text{ for all } t, \tau \geq 0$$

P{New packet takes more than t sec to send} $\leq$ P{Old packet takes more than another t sec to send}

  - E.g., if a channel is jammed, it is unlikely to recover soon
  - NWU: Replicate a packet over all channels



- Exponential distribution is both NBU & NWU



Intuition: Which File to Serve?

Average delay: $$D_{\text{avg}}(\mathbf{C}(\pi)) = \frac{1}{n}\sum_{i=1}^{n}[C_i(\pi) - a_i]$$

Large file first: Ave. Delay: 10.5

Small file first: Ave. Delay: 6

- Fewest Unassigned Packets/Tasks (FUT) First

Maximum lateness: $$L_{\max}(\mathbf{C}(\pi)) = \max_{i=1,\ldots,n}[C_i(\pi) - d_i]$$

- Earliest Due Date (EDD) First

Maximum Delay: $$D_{\max}(\mathbf{C}(\pi)) = \max_{i=1,\ldots,n}[C_i(\pi) - a_i]$$

- First-Come, First-Served (FCFS)

Policies and Theorems

THE OHIO STATE UNIVERSITY

Thm	size	Arrival	Due time	Delay metric	Service time distributions	Policy	Optimality
1	Any	Any	Any	Avg delay	*Independent* NBU	FUT-NC	Near optimal
2	Any	Any	Any	Avg delay	*Independent* Exp	FUT-RC	Near optimal
3	$k_i = k$	Any	Any	Avg delay	*Independent* NWU	FUT-RC	Optimal
4	Any	Any	Any	Max lateness	*Independent* NBU	EDD-NC	Near optimal
5	Any	Any	Any	Max lateness	*Independent* Exp	EDD-RC	Near optimal
6	Any	Any	$d_i \leq d_{i+1}$	Max lateness	*Independent* NWU	EDD-RC	Optimal
7	Any	Any	Any	Max delay	*Independent* NBU	FCFS-NC	Near optimal
8	Any	Any	Any	Max delay	*Independent* Exp	FCFS-RC	Optimal
9	Any	Any	Any	Max delay	*Independent* NWU	FCFS-RC	Optimal

- Following these simple design principles for coding and scheduling:
  - We can show delay optimality/near-delay optimality
- Theorems shown in tabulated form

Policies and Theorems

THE OHIO STATE UNIVERSITY

Thm	size	Arrival	Due time	Delay metric	Service time distributions	Policy	Optimality
1	Any	Any	Any	Avg delay	*Independent* NBU	FUT-NC	Near optimal
2	Any	Any	Any	Avg delay	*Independent* Exp	FUT-RC	Near optimal
3	$k_i = k$	Any	Any	Avg delay	*Independent* NWU	FUT-RC	Optimal
4	Any	Any	Any	Max lateness	*Independent* NBU	EDD-NC	Near optimal
5	Any	Any	Any	Max lateness	*Independent* Exp	EDD-RC	Near optimal
6	Any	Any	$d_i \leq d_{i+1}$	Max lateness	*Independent* NWU	EDD-RC	Optimal
7	Any	Any	Any	Max delay	*Independent* NBU	FCFS-NC	Near optimal
8	Any	Any	Any	Max delay	*Independent* Exp	FCFS-RC	Optimal
9	Any	Any	Any	Max delay	*Independent* NWU	FCFS-RC	Optimal

- NBU distributions → No Coding (NC)
- Exp, NWU distributions → Replication coding (RC)

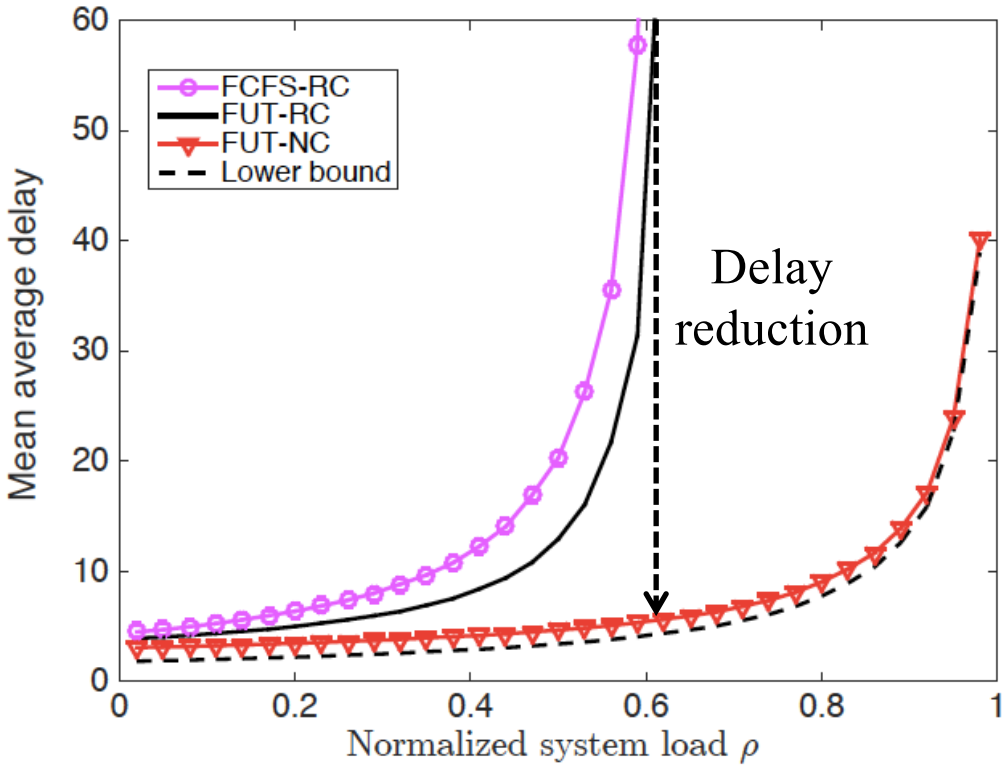
Policies and Theorems

THE OHIO STATE UNIVERSITY

Thm	size	Arrival	Due time	Delay metric	Service time distributions	Policy	Optimality
1	Any	Any	Any	Avg delay	*Independent* NBU	FUT-NC	Near optimal
2	Any	Any	Any	Avg delay	*Independent* Exp	FUT-RC	Near optimal
3	$k_i = k$	Any	Any	Avg delay	*Independent* NWU	FUT-RC	Optimal
4	Any	Any	Any	Max lateness	*Independent* NBU	EDD-NC	Near optimal
5	Any	Any	Any	Max lateness	*Independent* Exp	EDD-RC	Near optimal
6	Any	Any	$d_i \leq d_{i+1}$	Max lateness	*Independent* NWU	EDD-RC	Optimal
7	Any	Any	Any	Max delay	*Independent* NBU	FCFS-NC	Near optimal
8	Any	Any	Any	Max delay	*Independent* Exp	FCFS-RC	Optimal
9	Any	Any	Any	Max delay	*Independent* NWU	FCFS-RC	Optimal

- Average delay → Fewest Unassigned Packets/Tasks (FUT)
- Maximum lateness → Earliest Due Date (EDD)
- Maximum delay → First-Come, First-Served (FCFS)

Delay Performance: Independent Channels

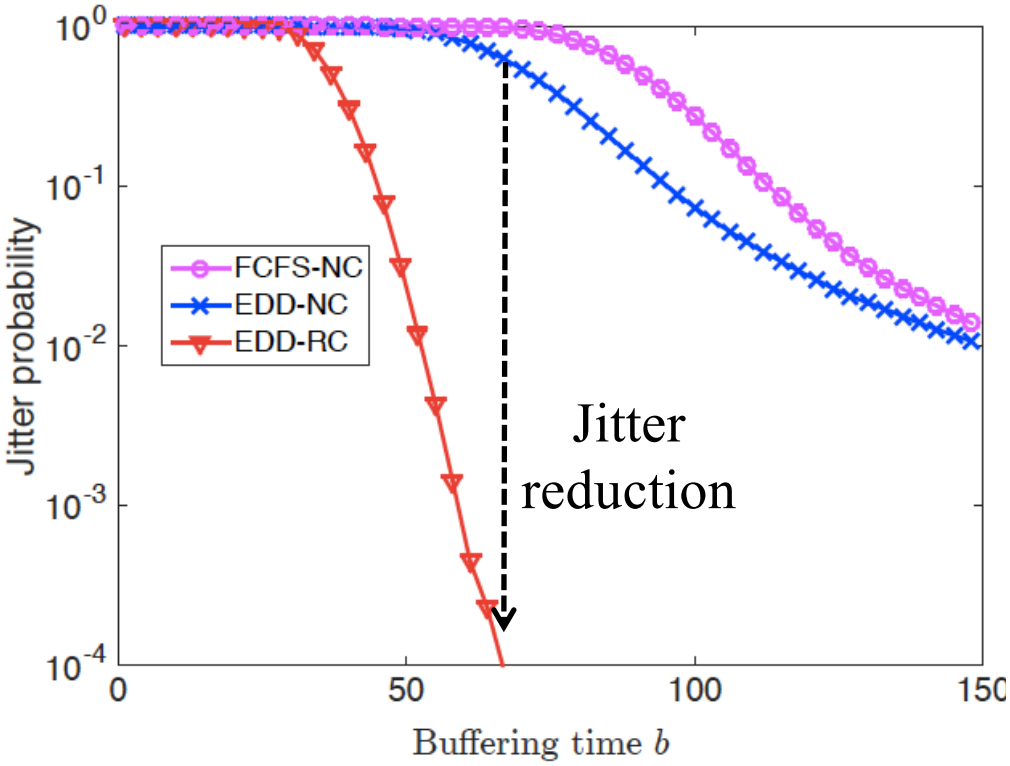


Mean average delay

Constant + Exp distribution (NBU)

Proposed policy: FUT-NC

Close to delay lower bound!
10x reduction in moderate loads!



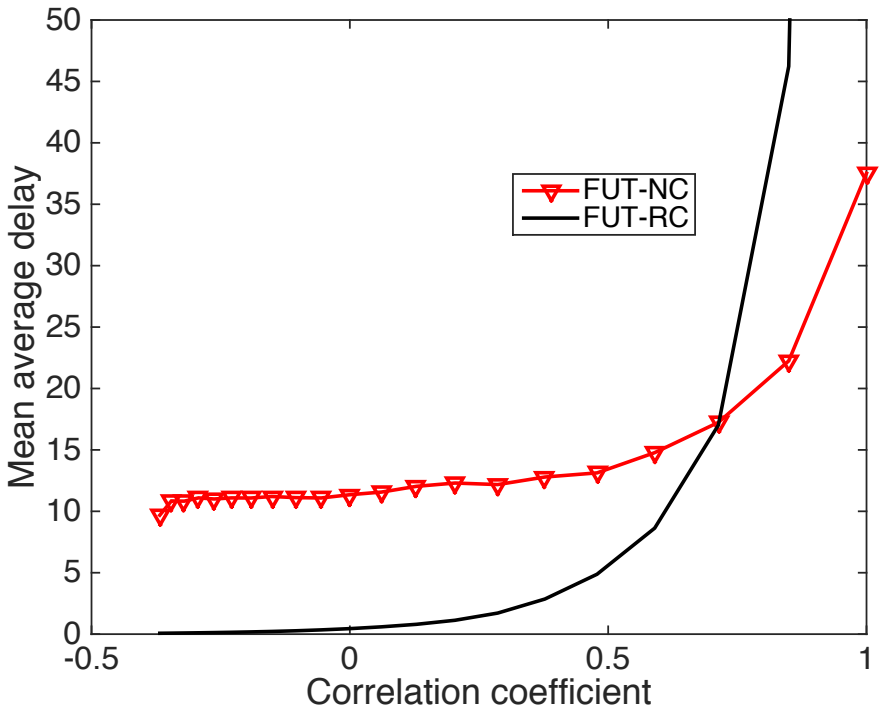
Tail probability of maximum lateness

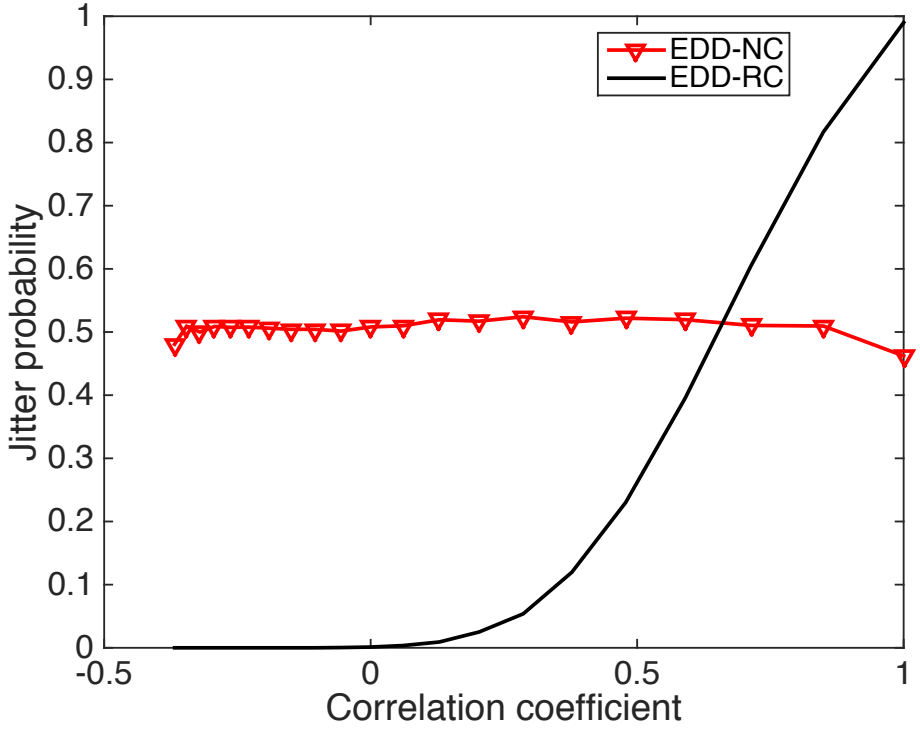
Pareto Type-II distribution (NWU)

Proposed policy: EDD-RC

EDD-RC is delay-optimal!
1000x reduction in jitter prob!

Delay Performance: Correlated Channels





Mean average delay

Log-normal distribution（heavy tail）

Normalized system load: 0.6

Tail probability of maximum lateness

Log-normal distribution (heavy tail)

Buffering time: 6

Coding is good for independent and negatively correlated channels,
Non-coding is good for strongly correlated channels.
Optimal policy to be studied…

Summary

- Delay-optimal control is notoriously difficult!
  - Optimal policies obtained in single-server/channel queueing systems.
  - Difficult problem for multi-server queues
  - Coding across channels further exacerbates this difficulty…
- Our design principles:
  - NWU servers: Replication Coding (RC)
  - NBU servers: No Coding (FC)
  - Average delay: Fewest Unassigned Packets/Tasks (FUT)
  - Maximum Lateness: Earliest Due Date (EDD)
  - Maximum Delay: First-Come First-Served (FCFS)

Designed efficient policies by using these principles

Proven near delay optimality

Yin Sun, C. Emre Koksal, and Ness Shroff, "On Delay-Optimal Scheduling in Queueing Systems with Replications," http://arxiv.org/abs/1603.07322, 2016.

Future Work

- Partial execution of computing jobs:
  - Partial execution has lower delay than full completion
- General service time distributions
  - Not necessarily NWU or NBU
- Correlated channels/servers
- Delay vs Cost
  - How many servers/channels do we need?
- Multi-resource allocation: Data, CPU, Memory, Network
- Other Systems
  - Coded Machine Learning, Coded Shuffling,…

Publications

THE OHIO STATE UNIVERSITY

1. Y. Kim, I. Koprulu and N. B. Shroff, "First Exit Time of a Levy Flight from a Bounded Region," **Advances in Applied Probability,** Vol. 52, No. 2, **2015**.
2. Y. Zheng, P. Sinha, and N. B. Shroff, "A New Analytical Technique for Designing Provably Efficient MapReduce Schedulers," **IEEE INFOCOM'13**, April 2013, Turin, Italy.
3. H. Cai, I. Koprulu, and N. B. Shroff "Exploiting Double Opportunities for Deadline Based Content Propagation in Wireless Networks," **IEEE INFOCOM'13,** April 2013, Turin, Italy (extended version submitted for journal publication).
4. Y. Kim, K. Lee, and N. B. Shroff, "An Analytical Framework to Characterize the Efficiency and Delay in a Mobile Data Offloading System," **ACM Mobihoc'14**, Philadelphia, PA, August 2014.
5. S. Buccapatnam, A. Eryilmaz, N. B. Shroff, "Stochastic Bandits with Side Observations on Networks," **ACM SIGMETRICS'14,** June 2014, Austin, Texas.
6. Y. Sun, Z. Zhang, E. Koksal, K-H. Lee, N. B. Shroff "Provably Delay Efficient Data Retrieving in Storage Clouds," **IEEE INFOCOM'15,** April 2015, Hong Kong.
7. Y. Zheng, N. B. Shroff, **R. Srikant,** and P. Sinha, "Exploiting Large System Dynamics for Designing Simple Data Center Schedulers," **IEEE INFOCOM'15,** April 2015, Hong Kong.
8. J. Lee, K. Lee, J. Jo, U. Jeong, and N. B. Shroff, "On the Benefit of Context-Aware Application Unloading in Mobile Systems", **ACM Ubicomp 2016,** Heidelberg, Germany, Sept. 12-16, 2016.
9. Y. Sun, C. E. Koksal, and N. B. Shroff, "On Delay-Optimal Scheduling in Queueing Systems with Replications," http://arxiv.org/abs/1603.07322, 2016.

Publications

THE OHIO STATE UNIVERSITY

10. Y. Kim, K. Lee, and N. B. Shroff, "On Stochastic Confidence of Information Spread in Opportunistic Networks," **IEEE Trans. on Mobile Computing (TWC)**, vol. 15, no. 4, pp. 909-923, Apr. 2016.
11. H. Cai, I. Koprulu, N. B. Shroff, "Exploiting Double Opportunities for Latency-Constrained Content Propagation in Wireless Networks," IEEE/ACM Trans. on Networking (ToN), vol. 24, no. 2, pp. 1025-1037, April 2016.
12. S. Kwon, Y. Kim, and N. B. Shroff, "Analysis of Connectivity and Capacity in One-Dimensional Vehicle-to-Vehicle Networks," **IEEE Trans. on Wireless Communications (TWC),** accepted for publication.
13. Z. Zheng and N. B. Shroff, "Online Multi-Resource Allocation for Deadline Sensitive Jobs with Partial Values in the Cloud," **IEEE INFOCOM'16**, San Francisco, CA, Apr. 2016.
14. I. Koprulu, Y. Kim, and N. B. Shroff, "Battle of Opinions over Evolving Social Networks," 50th **Conference on Information Sciences and Systems (CISS),** Princeton, NJ. Mar. 2016.

Thank You

Backup Slides

Simulations Model for Correlated Servers

- Two correlated servers
- Bivariate Correlated Gaussian Distributions:

$$X_2 = \eta X_1 + \sqrt{1-\eta^2} W$$

- Correlated Log-normal Distributions:

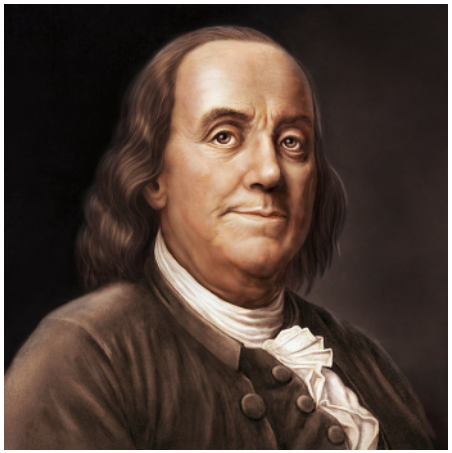
$$Y_i = e^{\sigma X_i} / \mathbb{E}[e^{\sigma X_i}]$$

$$\sigma = 2$$

- Correlation Coefficient:

$$\rho = \frac{e^{\eta} - 1}{e - 1} \in \left[\frac{e^{-1} - 1}{e - 1}, 1 \right]$$

Time is Money!

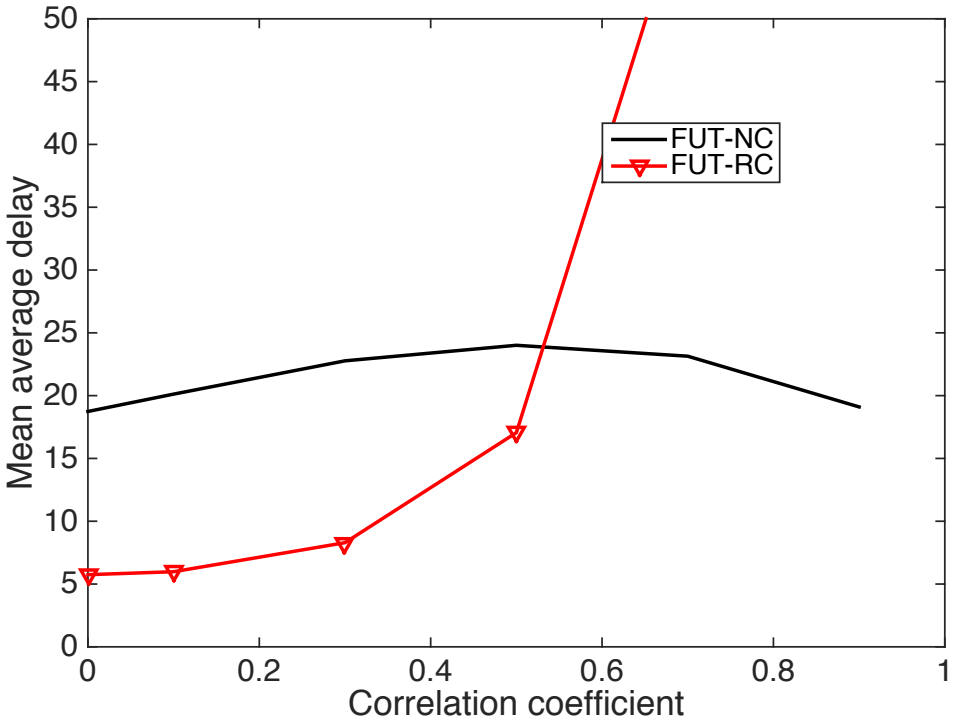


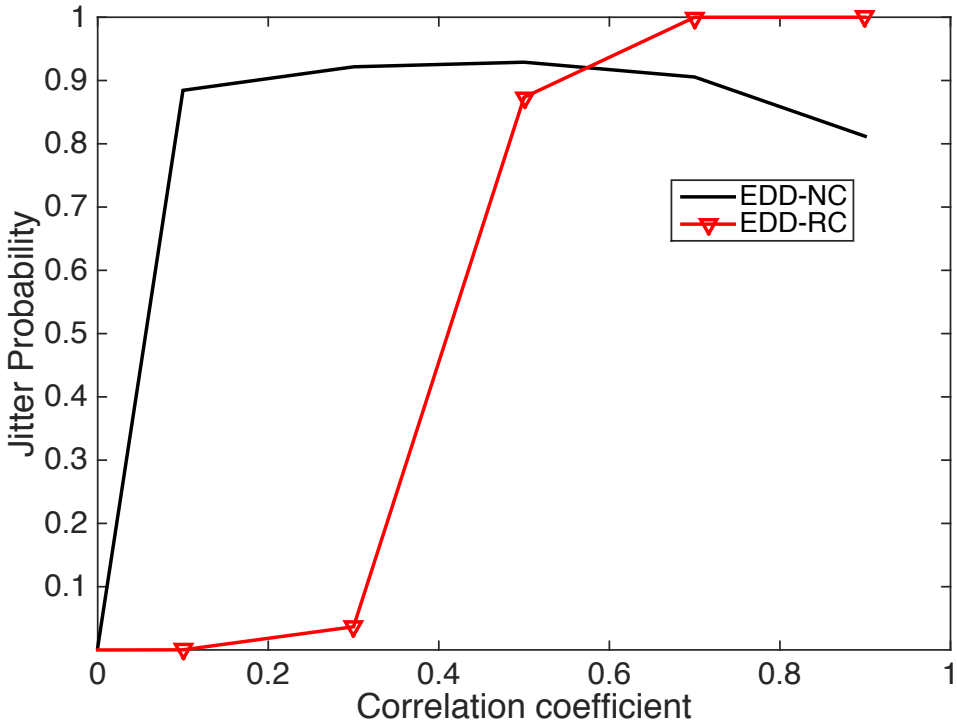
Benjamin Franklin

(1706-1790)

So, how do we reduce delay in clouds and wireless networks?

Delay Performance: Correlated Channels





Mean average delay

Pareto Type-II distribution (NWU)

Normalized system load: 0.3

Tail probability of maximum lateness

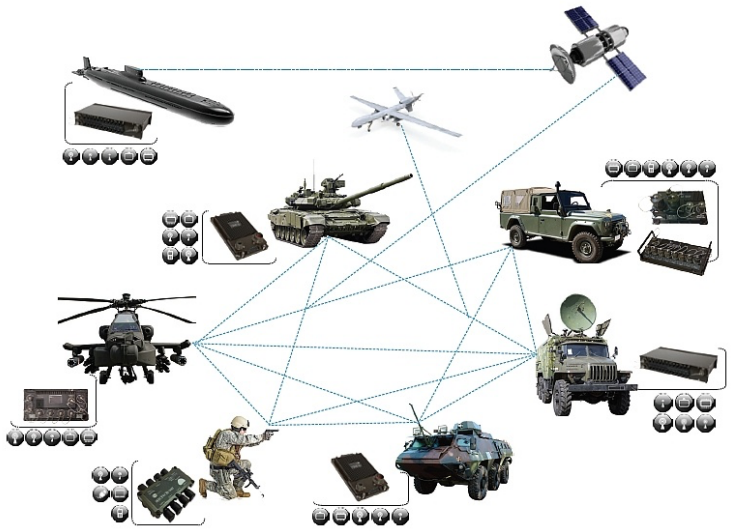
Pareto Type-II distribution (NWU)

Buffering time: 150

Coding is good for independent channels,
Non-coding is good for strongly correlated channels.

Summary of Impact

- Deeper Understanding of Mobility Patterns
  - Statistical understanding of how entities move in battlefields
- Reliable and Faster Communication Systems
  - Fast data retrieval and computation from cl
  - High throughput and low delay in heterogeneous communication networks
- Resource-efficient Mobile Devices
  - Enhancing the lifetime of mobile devices
  - Reducing the latency of tasks



*Leads to more **intelligent**, more **reliable**, and more **durable** military from central infrastructure to battlefields*

Importance of Low Latency

- **Web Searching**
  - Google: Display 10 to 30 results, delay grows from 400ms to 900ms, user traffic and ad revenue drops by 20%! [1]
  - Hence, low latency > additional results!!
- **Autonomous Vehicles/Drones:**
  - **Vehicles:** Maximum latencies between vehicles and objects ~ 10msec
  - **Aircraft:** Even lower latencies when coordinating between drones
- **E-Commerce**
  - Annual US online sales ~ $342B in 2015 [2]
  - amazon.com "Every 100 ms of latency costs 1% in sales" [3]

[1] http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html
[2] US Commerce Department
[3] Grag Linden, Make data useful, Stanford CS345 Talk, 2006.
[4] http://www.informationweek.com/wall-streets-quest-to-process-data-at-the-speed-of-light/d/d-id/1054287?