# Supporting Delay-Sensitive Applications on Next-Generation Wireless Networks


Xiaojun Lin

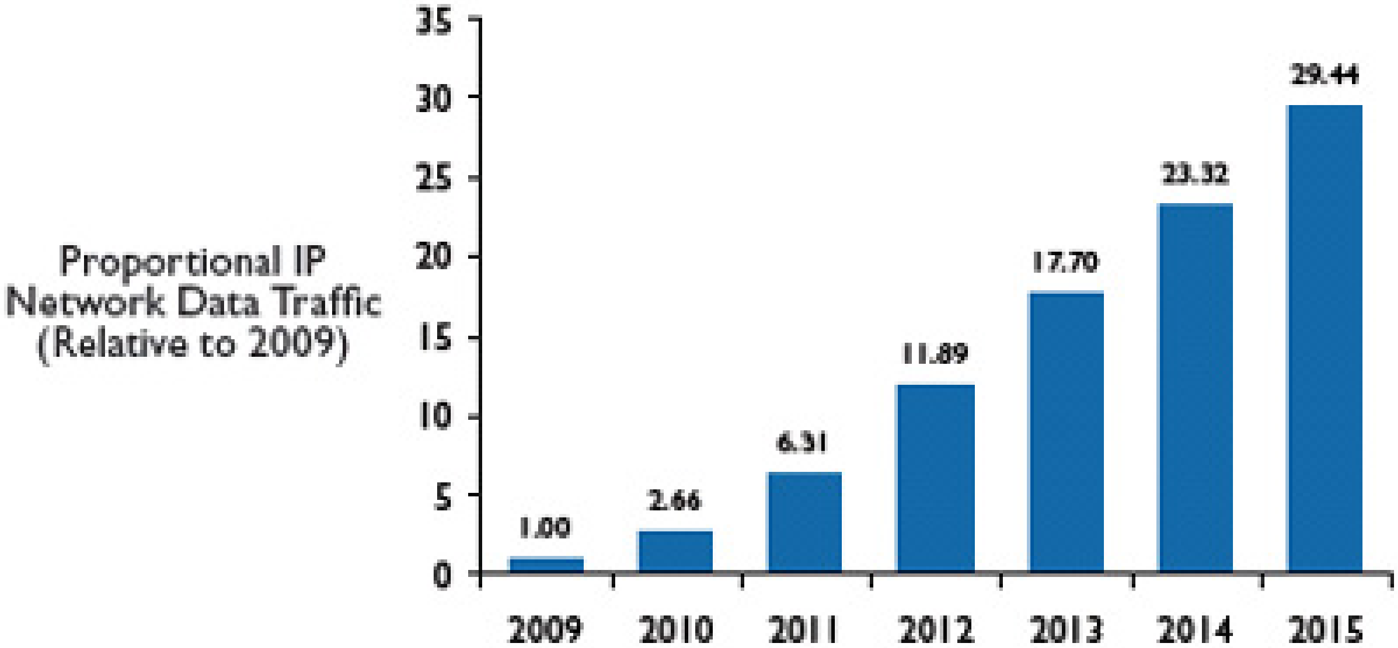
School of Electrical and Computer Engineering

Purdue University, West Lafayette

http://min.ecn.purdue.edu/~linx

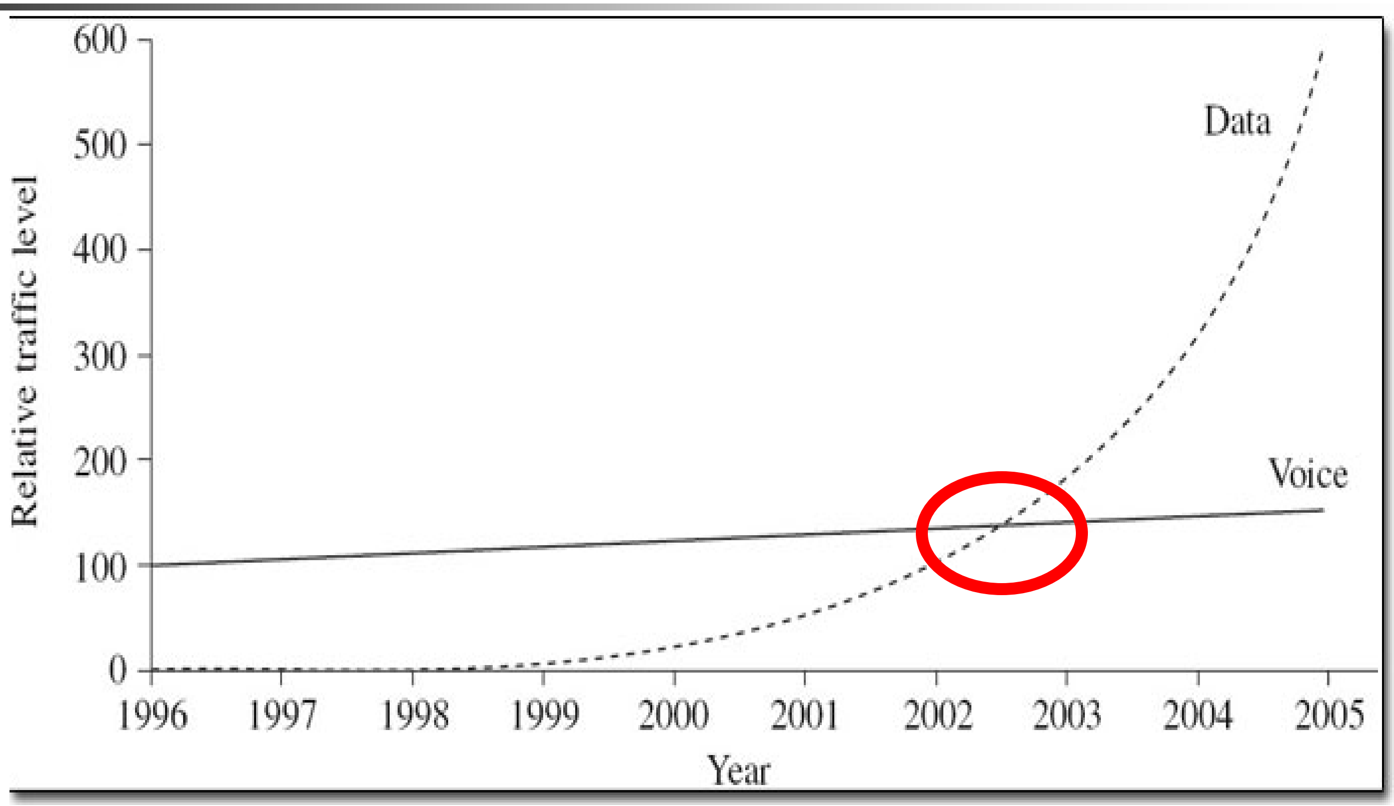

Joint work with Venkataramanan VJ

# A Time of Change




- “Mobile data to outstrip voice traffic by 2011” (Nokia-Siemens, July 2009)

# Data Exceeds Voice in Internet




- Data traffic exceeds voice traffic during 2002 (Coffman and Odlyzko, 1998)

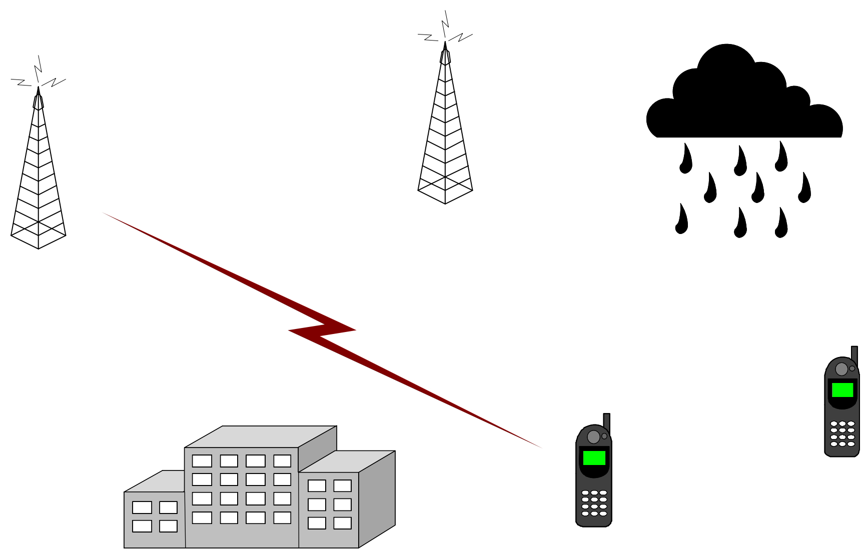
# Convergence to A Single Packet-Based Network

- Internet becomes a single IP-based network for all voice/data/video services
  - The telephone network becomes a part of IP Internet: VoIP
  - Cost reduction, ease of management
  - Enabling new applications: e.g., Internet TV
- Will the same trend occur in mobile wireless networks?
  - To move towards fully packet-based networks: from 3G and WiFi to LTE and WiMax
  - To support both data applications and delay-sensitive applications

# Convergence to A Single Packet-Based Network

- Challenges:
  - Wider radio spectrum: cognitive spectrum reuse
  - Faster bit-pipe
  - Efficient management of resources (spectrum, power, etc)
  - ***Need to provide stringent delay-guarantees in an efficient manner***

# Difficulty in Providing Delay-Guarantees in Wireless Networks



- Interference
- Time-varying channel condition due to mobility and fading
- Radio spectrum is scarce

# Difficulty in Providing Delay-Guarantees in Wireless Networks

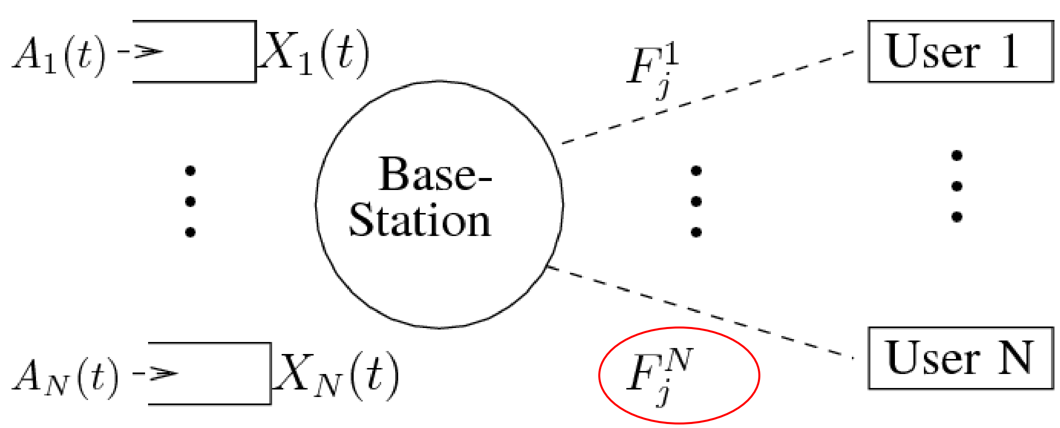
- Protocol Level:
  - Service differentiation: e.g. priority
  - Admission control
  - ***Not enough!***
- Delay Analysis and Design:
  - How do interference and channel variations affect the delay performance?
  - What is the delay performance of existing control algorithms?
  - How to design delay-optimal algorithms?

# Outline

- *System Model*
- Capacity Maximizing Algorithms without Considering Delay
- Delay Performance: Main results
- Practice: Delay-Optimal Control Algorithms
- Key Idea of Analysis: Large Deviations + Lyapunov Stability
- Conclusion

# System Model: Downlink of a Single Cell

- $N$ users. Time is slotted.
- Only one user can be served at a given time.




$C(t)$: channel state at time slot $t$

(*i.i.d.* across time)

$$p_j = \mathbf{P}[C(t) = j],$$
$$j = 1, 2, ..., M$$

- $F_j^i$ : is the rate to user $i$ if it is selected for service and the channel state $C(t)=j$

# Channel Variations

There are two ways to deal with channel variations

- To ***mitigate*** channel variations: increase transmission power when the channel is poor.
  - **Poor** users gets more resources (e.g., power)
  - Used in 2G cellular systems to maintain a constant rate to each user

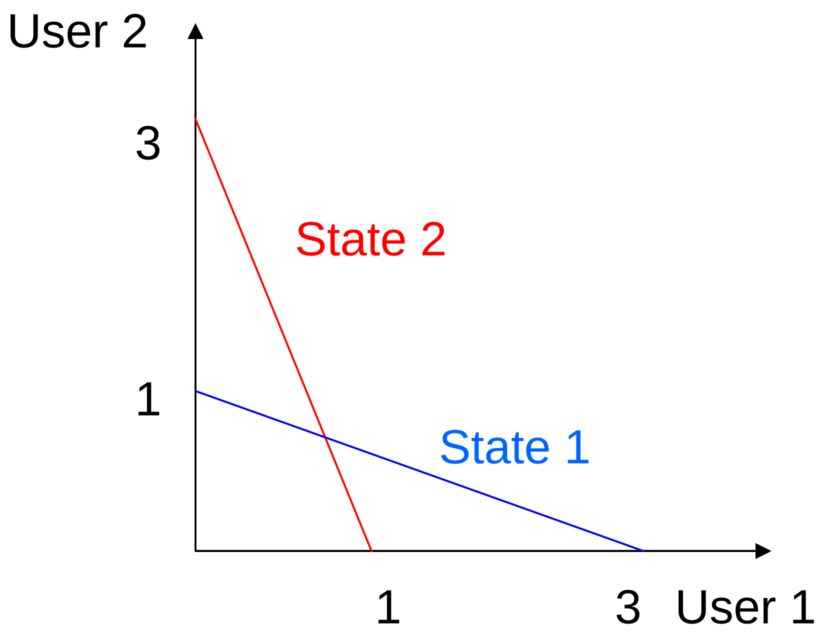
- To ***exploit*** channel variations: serve users when their channels are good
  - **Good** users get more resources
  - Can significantly increase system capacity

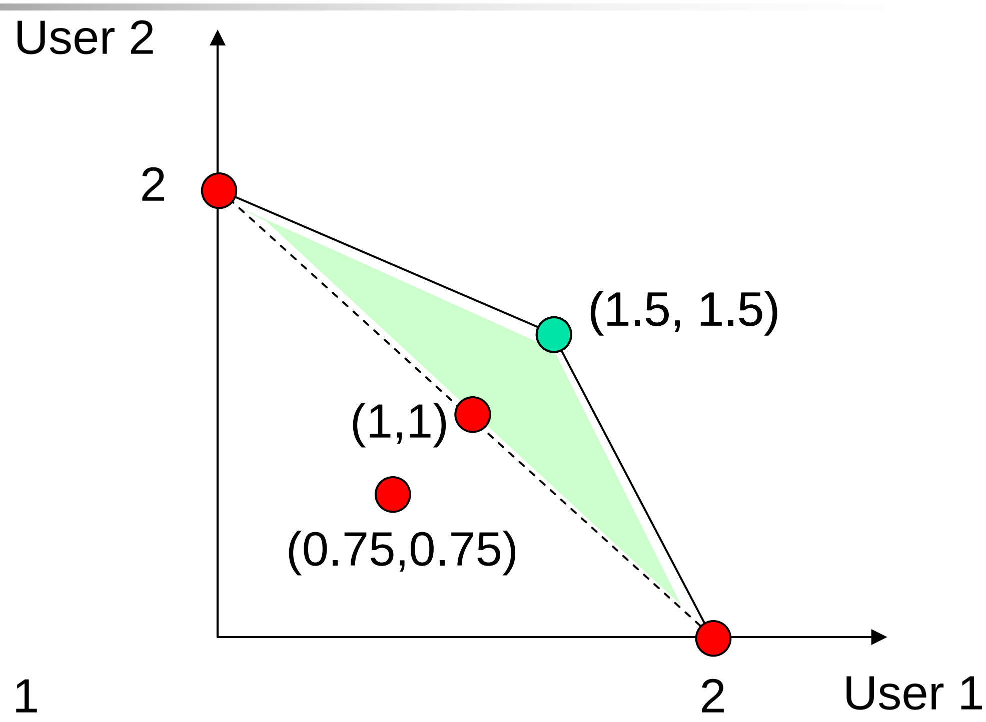
# Exploiting Channel Variations

| | Probability | Rate for user 1 | Rate for user 2 |
|---|---|---|---|
| State 1 | 1/2 | 3 | 1 |
| State 2 | 1/2 | 1 | 3 |

- If we want to ensure each user receives a constant rate at all states
  - At each state the basestation transmits to the good user ¼ of the time, and to the bad user 3/4 of time
  - Each of them will get a constant rate of 0.75.
- If we select the user to serve at its best time-slots
  - Each of them will get an average rate of 1.5!

# Exploiting Channel Variations: Tradeoff Between Capacity Gain and Delay



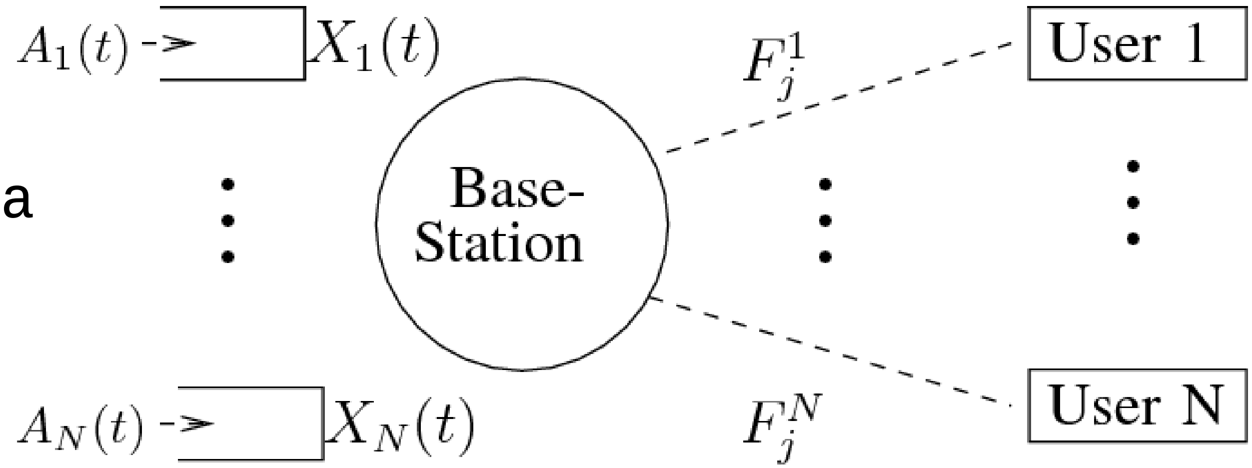




- ***Capacity*** gain (the green region) versus increasing ***delay***

# Outline

- System Model
- *Capacity Maximizing Algorithms without Considering Delay*
- Delay Performance: Main results
- Practice: Delay-Optimal Control Algorithms
- Key Idea of Analysis: Large Deviations + Lyapunov Stability
- Conclusion

# Maximizing Capacity Without Delay Considerations




$A_i(t)$: amount of data that arrive for user $i$

$X_i(t)$: queue for user $i$

$$X_i(t+1) = \left[X_i(t) + A_i(t) - \sum_{j=1}^{\mathcal{S}} F_j^i \mathbf{1}_{\{C(t)=j, U(t)=i\}}\right]^+$$

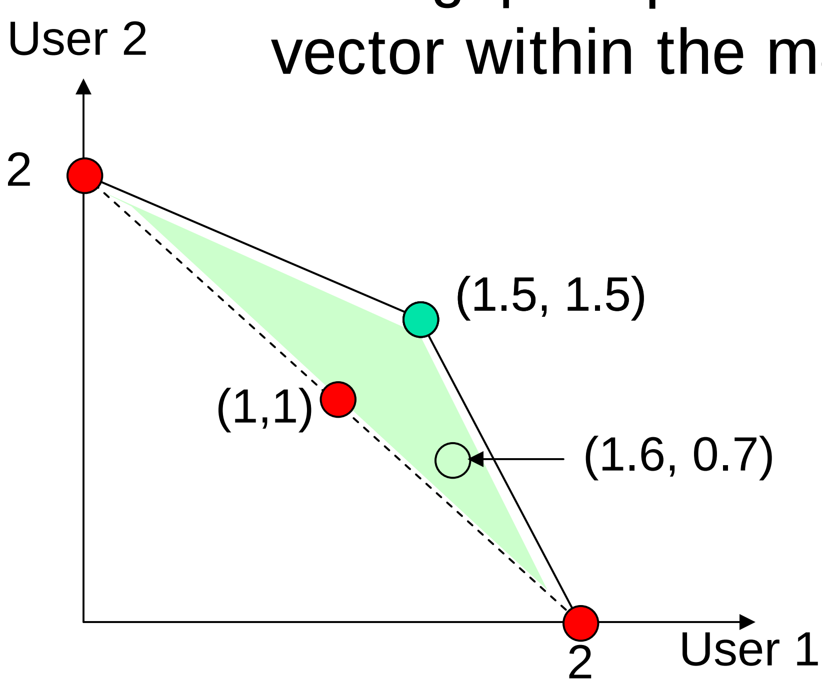
user $i$ is served at time $t$

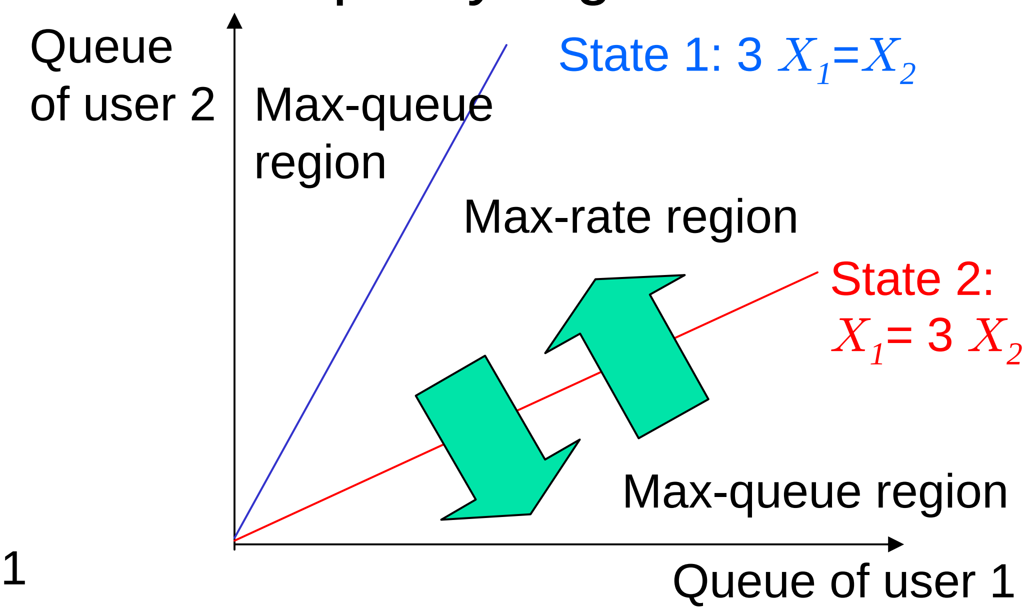
# Maximum-Weight Algorithms

- Choose the user that maximizes the queue-weighted rate, i.e., when $C(t)=j$

$$U(t) = \operatorname*{argmax}_i F_j^i X_i(t)$$

- Throughput optimal: can support any offered load vector within the maximum capacity region

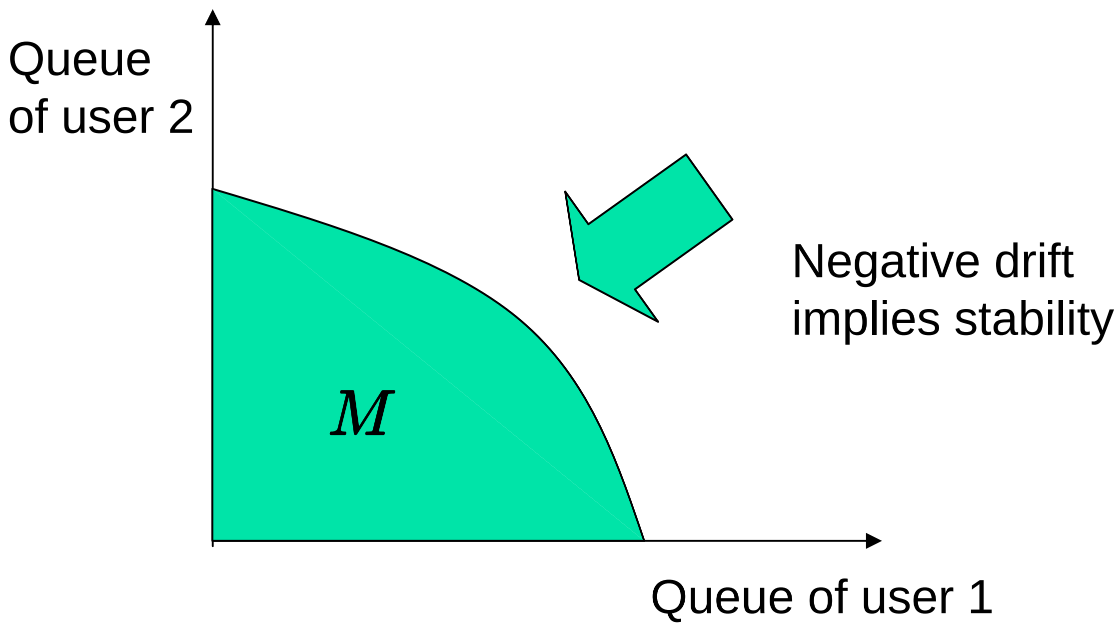


# Lyapunov Function $V(\vec{X})$

- $V(\vec{X}) \geq 0$. $V(\vec{X}) \to +\infty$ as $||\vec{X}|| \to \infty$.
- Negative drift: Except in a bounded set $M$,

$$\mathbf{E}[V(\vec{X}(t+1)) - V(\vec{X}(t))|\vec{X}(t)] < 0$$



# Throughput-Optimality of the Maximum-Weight Policy

- Use the Laypunov function: $V(\vec{X}) = \sum_i X_i^2$
- Derive the drift

$$\mathbf{E}[V(\vec{X}(t+1)) - V(\vec{X}(t))|\vec{X}(t)]$$

$$\approx \mathbf{E}[\sum_i X_i(t)(A_i(t) - D_i(t))|\vec{X}(t)]$$

$$= \boxed{\sum_i X_i(t)\lambda_i(t)} - \mathbf{E}\boxed{[\sum_{i=1}^{N} X_i(t)D_i(t)]}.$$

Choose the Service Vector that Maximize This term

When $D_i(t) = \sum_{j=1}^{S} F_j^i \mathbf{1}_{\{C(t)=j, U(t)=i\}}$,

$\max \sum_i X_i(t) D_i(t)$ is equivalent to MW policy.

# Drift-Minimizing Policies

- The maximum-weight policy in fact **minimizes the drift of the Lyapunov function**!
- If we choose a different Lyapunov function

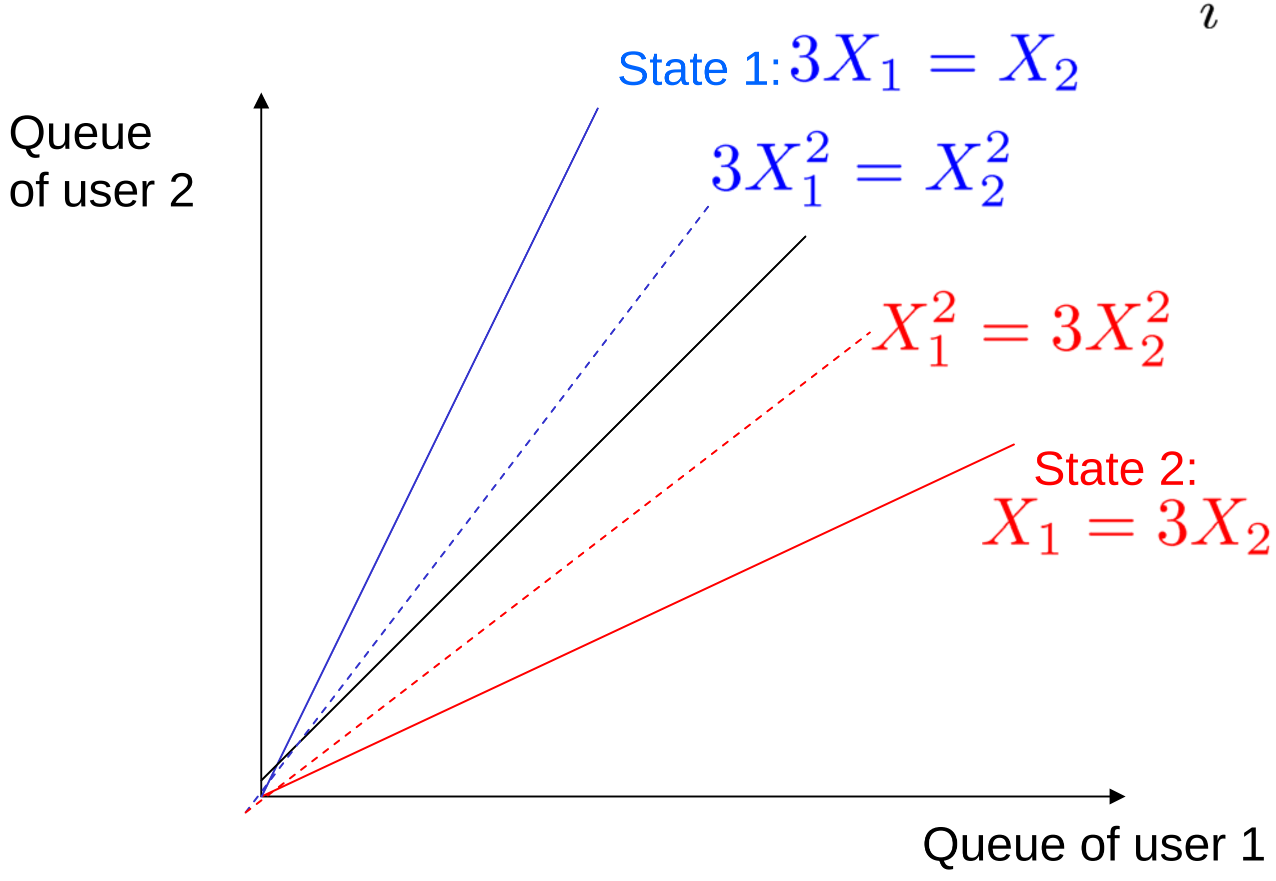
$$V(\vec{X}) = \sum_{i=1}^{N} X_i^{1+\alpha}$$

- The drift is

$$\begin{aligned}
& 1/(1+\alpha)\mathbf{E}[V(\vec{X}(t+1)) - V(\vec{X}(t)|\vec{X}(t)] \\
\approx \quad & \mathbf{E}[\sum_i X_i^{\alpha}(t)(A_i(t) - D_i(t))|\vec{X}(t)] \\
= \quad & \sum_i X_i^{\alpha}(t)\lambda_i(t) - \mathbf{E}[\sum_{i=1}^{N} X_i^{\alpha}(t)D_i(t)].
\end{aligned}$$

# All MW-α Policies are Throughput-Optimal

- MW-α Policy: $U(t) = \operatorname*{argmax}_i F_j^i X_i^\alpha(t)$




**As α increases, these lines become closer to the diagonal line**

# Outline

- System Model
- Capacity Maximizing Algorithms Without Considering Delay
- *Delay Performance: Main Results*
- Practice: Delay-Optimal Control Algorithms
- Key Idea of Analysis: Large Deviations + Lyapunov Stability
- Conclusion

# Delay-Performance

- Algorithms like MW-$\alpha$ have been the basis for many cross-layer wireless control algorithms.
- ***Open Question***: Which of these policies will have good delay-performance?

- Delay objective can be mapped to a suitable objective function of the queue length
  - What is the probability $\mathbf{P}[\max_i X_i \geq B]$?
    - maximum delay among all users
  - What is the probability $\mathbf{P}[\sum_i X_i \geq B]$?
    - delay averaged over all users.

# Large-Buffer Asymptotes

- Unfortunately, the exact overflow probability is in general difficult to study due to the correlation of the service rates among queues.
- One can use large-deviations theory and instead study the following asymptotic decay rate

$$I = -\lim_{B \to \infty} \frac{1}{B} \log \mathbf{P}[D(\vec{X}(t)) \geq B]$$

- A larger decay-rate corresponds to a smaller queue-overflow probability.

$$\mathbf{P}[D(\vec{X}(t)) \geq B] \approx Ce^{-IB}$$

# Our Main Result

- If an algorithm minimizes the drift of a Lyapunov function $V(\vec{X})$ at every time (in the fluid limit),

- Then the algorithm is optimal in the sense that it maximizes the asymptotic decay rate of the probability that the Lyapunov function value $V(\vec{X})$ exceeds a large threshold

- In other words, it maximizes the decay rate

$$-\lim_{B\to\infty}\frac{1}{B}\log\mathbf{P}[(V(\vec{X})\geq f(B)]$$

# Consequences

- **Analysis** – Cellular Downlink
  - MW-$\alpha$ minimizes drift of the Lyapunov function $V(\vec{x}) = \sum_{i=1}^{N} X_i^{\alpha+1}$
    - Or equivalently $V(\vec{X}) = (\sum_{i=1}^{N} X_i^{\alpha+1})^{\frac{1}{\alpha+1}}$
  - By our result, MW-$\alpha$ is optimal in maximizing the decay rate of

$$\mathbf{P}[(\sum_i X_i^{\alpha+1})^{\frac{1}{\alpha+1}} \geq B] = \mathbf{P}[\sum_i X_i^{\alpha+1} \geq B^{\alpha+1}]$$

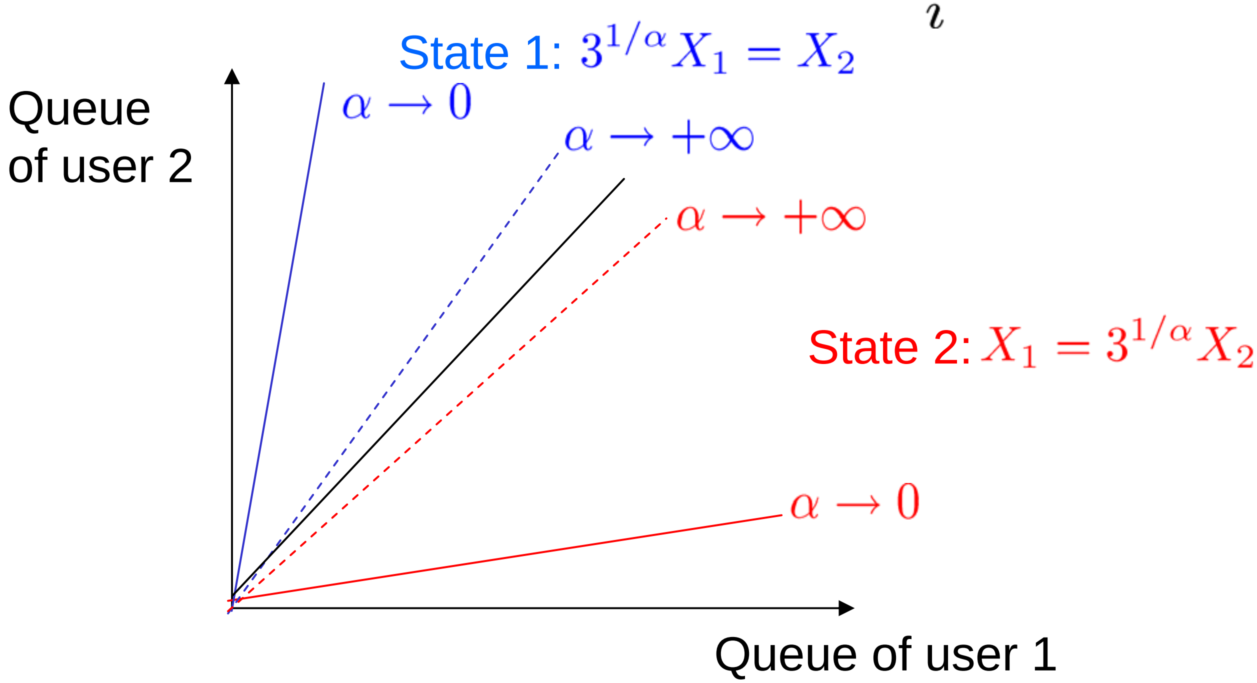
# Consequences

- **Design:**
  - Note that $\lim_{\alpha\to\infty}(\sum_i X_i^{\alpha+1})^{\frac{1}{\alpha+1}} = \max_i X_i$
  - As $\alpha \to \infty$, MW-α asymptotically maximizes decay rate of $\mathbf{P}[\max_i X_i \geq B]$

  - Also $\lim_{\alpha\to 0}(\sum_i X_i^{\alpha+1})^{\frac{1}{\alpha+1}} = \sum_i X_i$
  - As $\alpha \to 0$, MW-α asymptotically maximizes decay rate of $\mathbf{P}[\sum_i X_i \geq B]$

# Delay-Optimal Control Algorithms

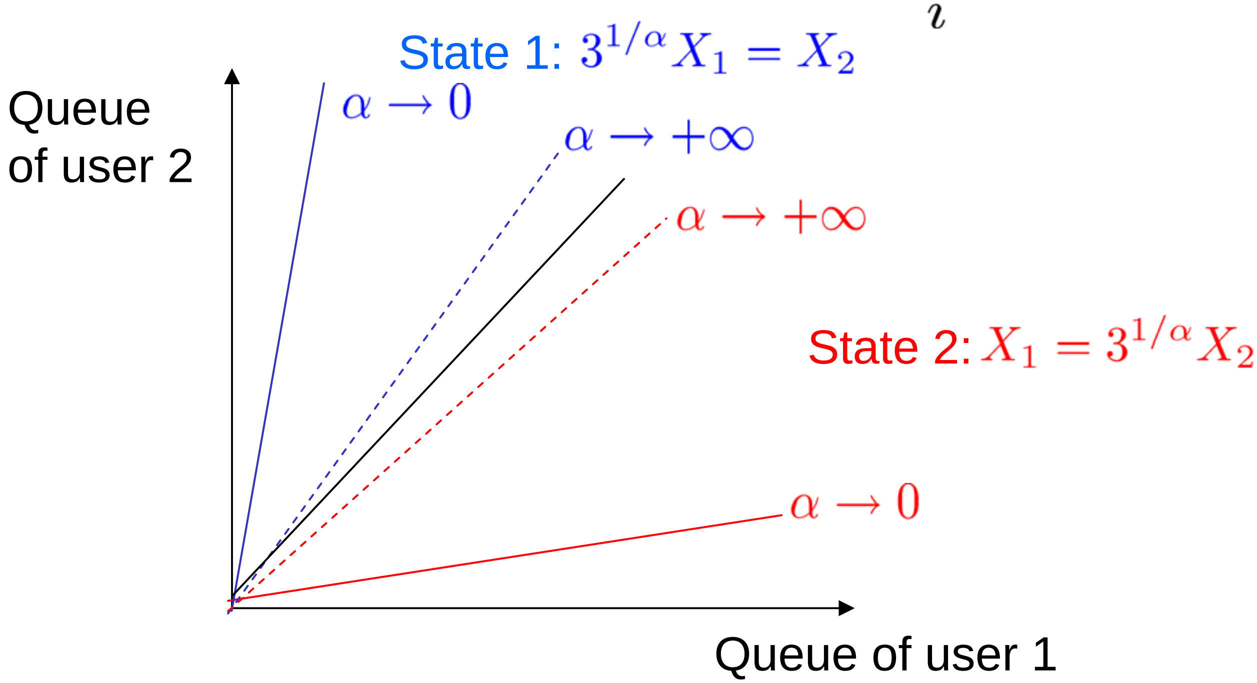
- MW-α Policy: $U(t) = \arg\max_i F_j^i X_i^{\alpha}(t)$




- $\alpha \longrightarrow +\infty$ : place more emphasis on serving the longest queue (good for max-queue)

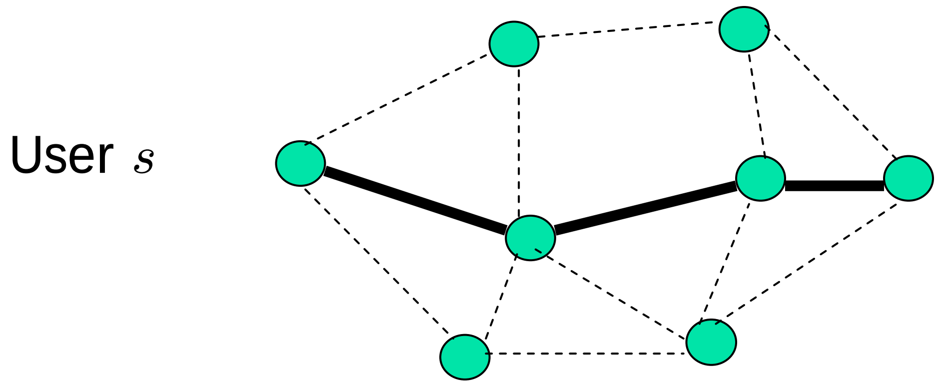
# Delay-Optimal Control Algorithms

- MW-α Policy: $U(t) = \arg\max_i F_j^i X_i^{\alpha}(t)$




- $\alpha \to 0$ : place more emphasis on serving the largest rate (good for sum-queue)

# Multi-hop Wireless Networks




- Back-pressure algorithm (Tassiulas & Ephremides '92)
  - $X_i^d$ : the queue at node $i$ for flow $d$
  - Each link ($i$,$j$) serves the flow $\hat{d}$ with the largest differential backlog $\hat{d}_{ij} = \mathrm{argmax}(X_i^d - X_j^d)$
  - The weight of each link is $w_{ij}^d = (X_i^{\hat{d}} - X_j^{\hat{d}})$
  - The links are scheduled to maximize the sum of the weighted-rate $\sum_{ij} w_{ij} r_{ij}$

# Optimality of the Back-Pressure Algorithm

- The back-pressure algorithm is known to minimize the drift of the Lyapunov function

$$V(\vec{X}) = \sum_{i,d} (X_i^d)^2$$

- By our result, it is optimal in maximizing the decay rate of

$$\mathbf{P}[\sum_{i,d} (X_i^d)^2 \geq B^2]$$

# Generalized Back-Pressure Algorithm: BP-$\alpha$

- Instead, we can take $V(\vec{X}) = \sum_{i,d} (X_i^d)^{\alpha+1}$
- Generalized back-pressure algorithm (BP-$\alpha$ )
  - Each link $(i,j)$ serves the flow $\hat{d}$ with the largest differential backlog $\boldsymbol{\hat{d}_{ij} = \mathrm{argmax}_d[(X_i^d)^\alpha - (X_j^d)^\alpha]}$
  - The weight of each link is $w_{ij} = [(X_i^{\hat{d}})^\alpha - (X_j^{\hat{d}})^\alpha]$
  - The links are scheduled to maximize the sum of the weighted-rate $\sum_{ij} w_{ij} r_{ij}$
- Each BP-$\alpha$ policy is optimal in maximizing the decay-rate of $\mathbf{P}[\sum_{i,d} (X_i^d)^{\alpha+1} \geq B^{\alpha+1}]$

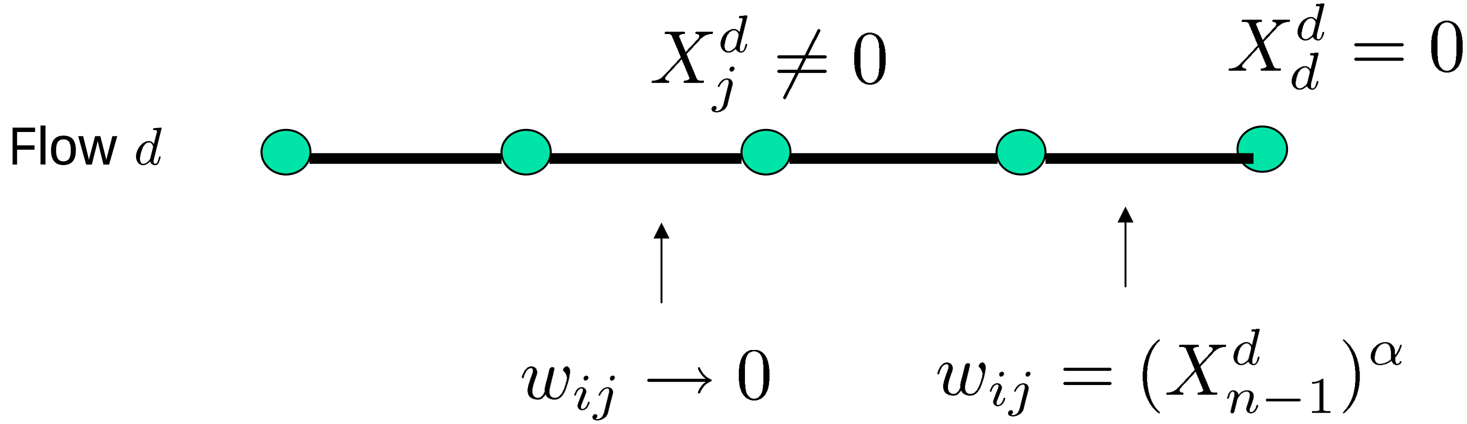
# Minimizing the Sum-Queue

- Suppose we want to minimize the sum-queue (correspondingly, the overall end-to-end delay of all flows)

- Note that $\lim_{\alpha \to 0} (\sum_{i,d} (X_i^d)^{\alpha+1})^{\frac{1}{\alpha+1}} = \sum_{i,d} X_i^d$

- As $\alpha \to 0,$ BP-$\alpha$ asymptotically maximizes decay rate of $\mathbf{P}[\sum_{i,d} X_i^d \geq B]$

# Minimizing the End-to-End Delay




- Note that the weight of each link is

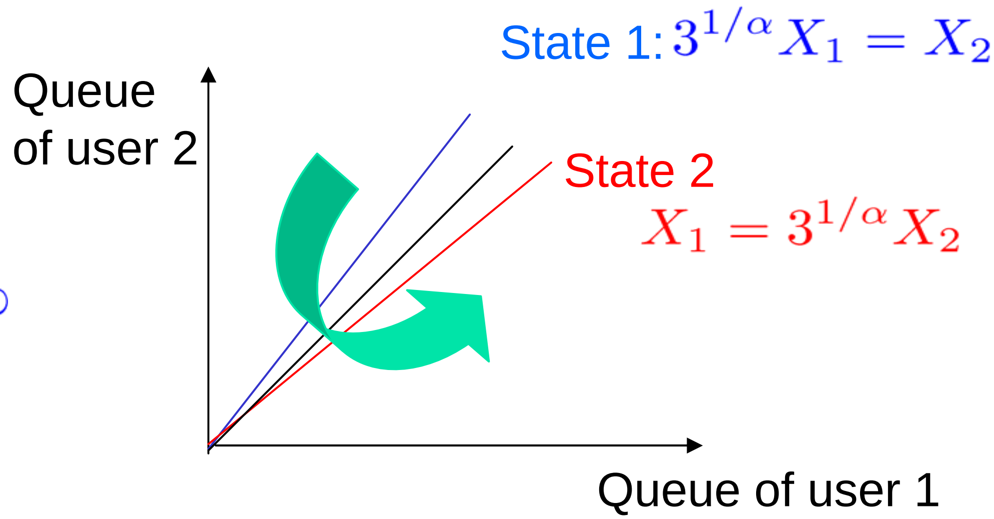
$$w_{ij} = [(X_i^{\hat{d}})^\alpha - (X_j^{\hat{d}})^\alpha]$$

- $\alpha \to 0$ : place higher priority on serving the links closer to the destination (which generalizes the result of [Tassiulas & Ephremides ‘94])

# Outline

- System Model
- Capacity Maximizing Algorithms
- Delay Performance: Main Results
- *Practice: Delay-Optimal Control Algorithms*
- Key Idea of Analysis: Large Deviations + Lyapunov Stability
- Conclusion

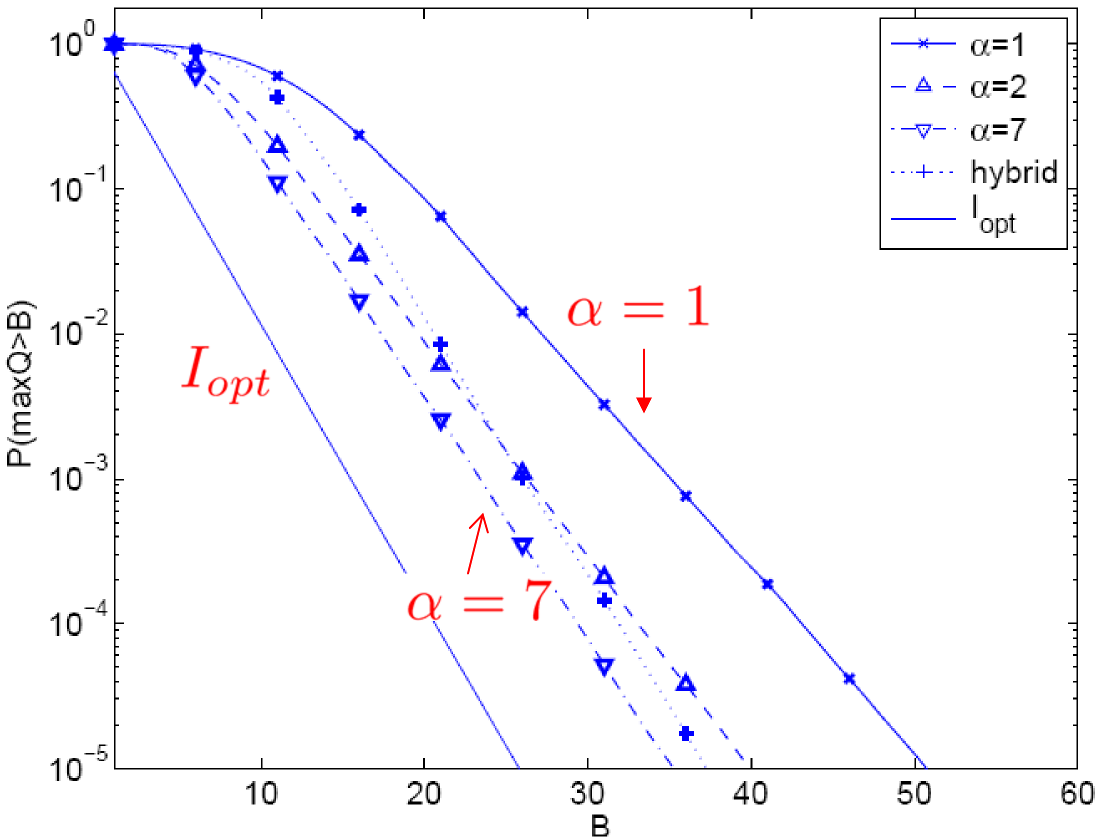
# Practice: Minimize the Max-Queue in Cellular Downlink




$\alpha \to +\infty$

- As α increases, the max-rate region shrinks,
  - The queue state might jump from one max-queue region to the other max-queue region, without entering the max-rate region
  - The queue will grow until the max-rate region opens up
  - Although the decay rate is larger, the overflow prob. decays later

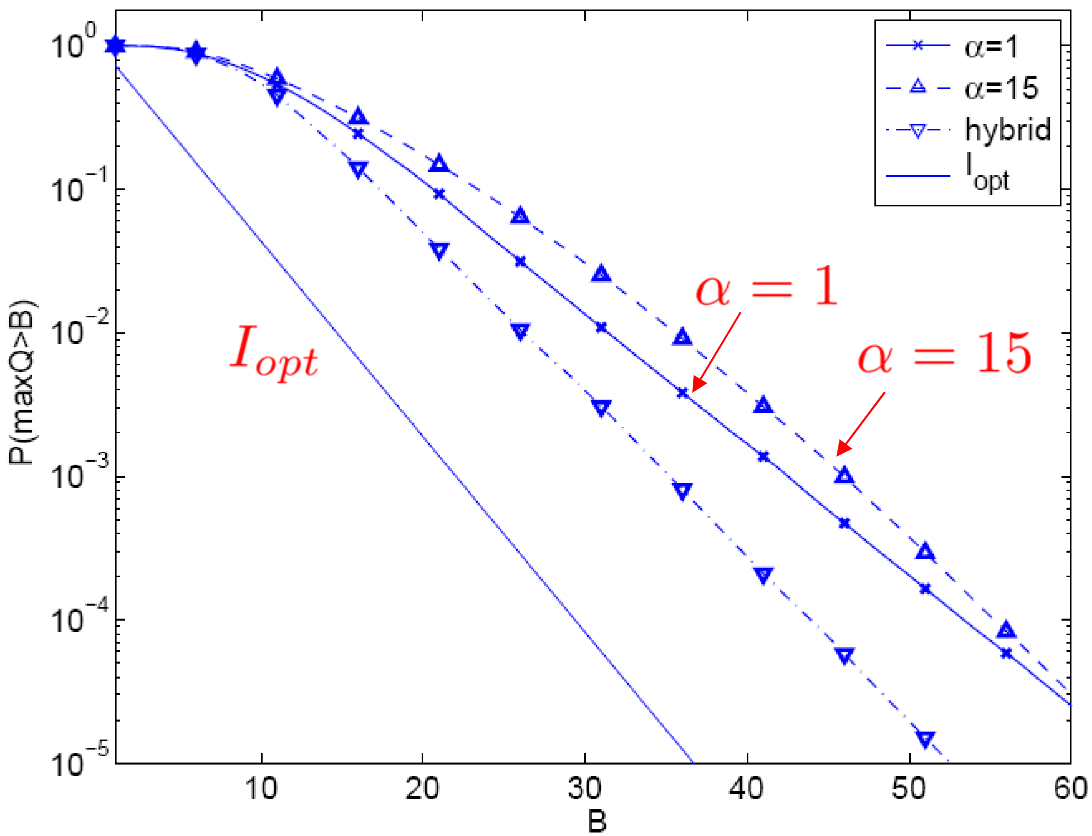
# Simulation Results: “Good Case”




Case 1: Plot of $\mathbf{P}[\max_{1\leq i\leq N} Q_i \geq B]$ vs $B$ for the $\alpha$-algorithms.

4-user downlink with three channel states.

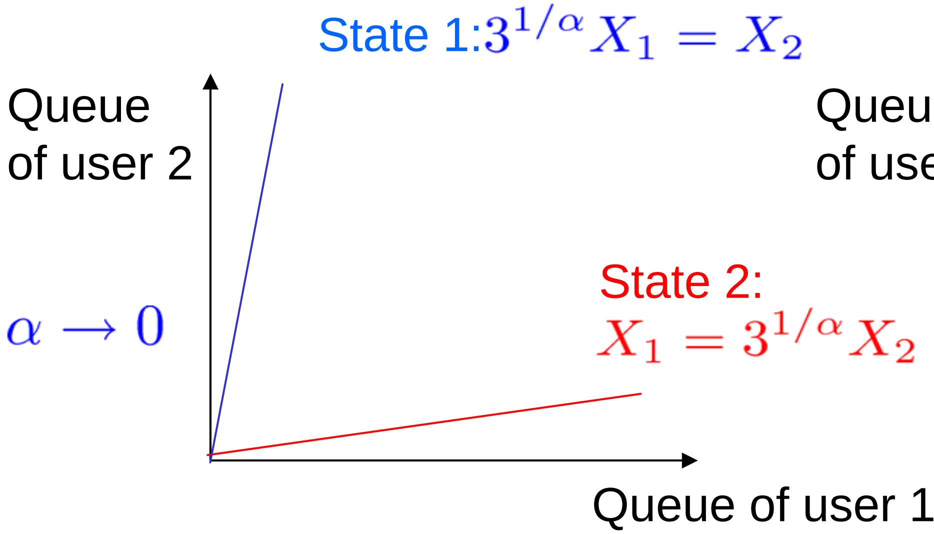
# Simulation Results: "Bad Case"

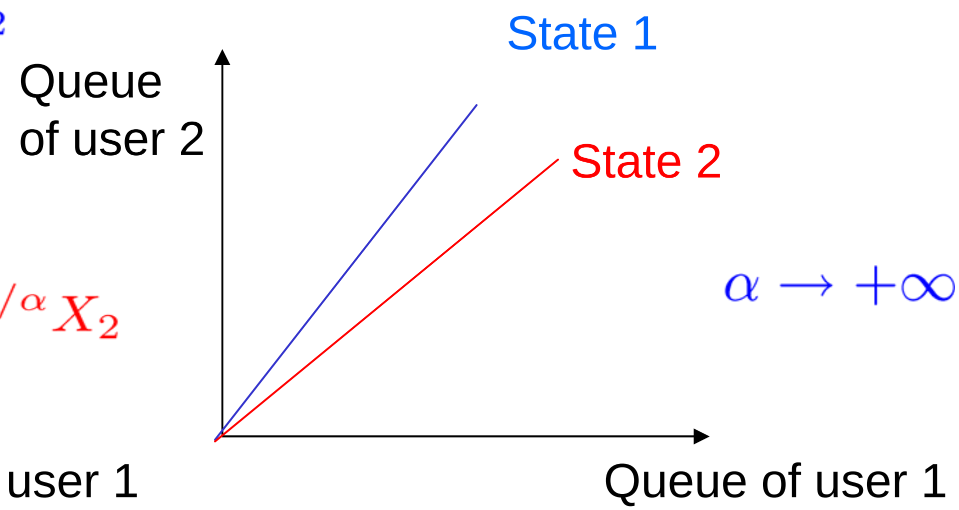



Case 2: Plot of $\mathbf{P}[\max_{1\le i\le N} Q_i \ge B]$ vs $B$ for the $\alpha$-algorithm.

A higher value of α may result in poorer performance for practical range of queue length.

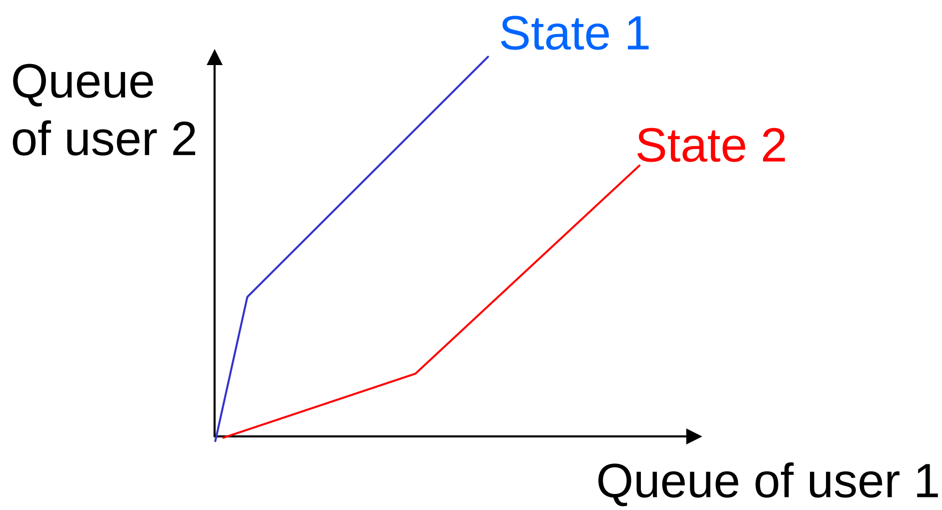
# Combining Large and Small $\alpha$



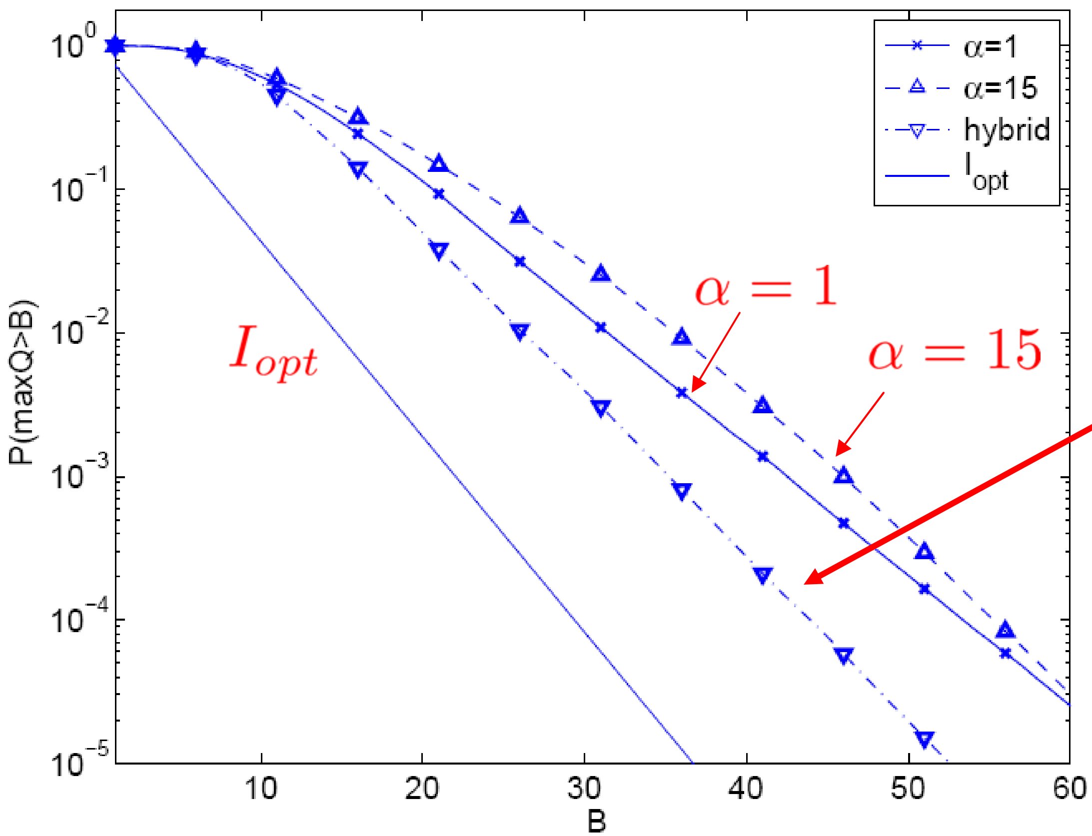

- We need an algorithm that
  - Has 45 degree boundary lines (property of large $\alpha$).
    - Leads to balanced queues.
  - Has *wide* max-rate region (property of small $\alpha$ ).

# Hybrid Algorithms




- Hybrid algorithm serves user with largest value of $w_i(\vec{X})F_j^i$ where
$$w_i(\vec{X}) = X_i + \left(\left[X_i - K(\vec{X})\right]^+\right)^{15}$$
  - Combines properties of $a$=1 and $a$=15
  - $K(\vec{X})$ is chosen to ensure that the decision boundary is smooth and the transition occurs at the right point.

# The Hybrid Algorithm Performs Well Even in the "Bad Cases"



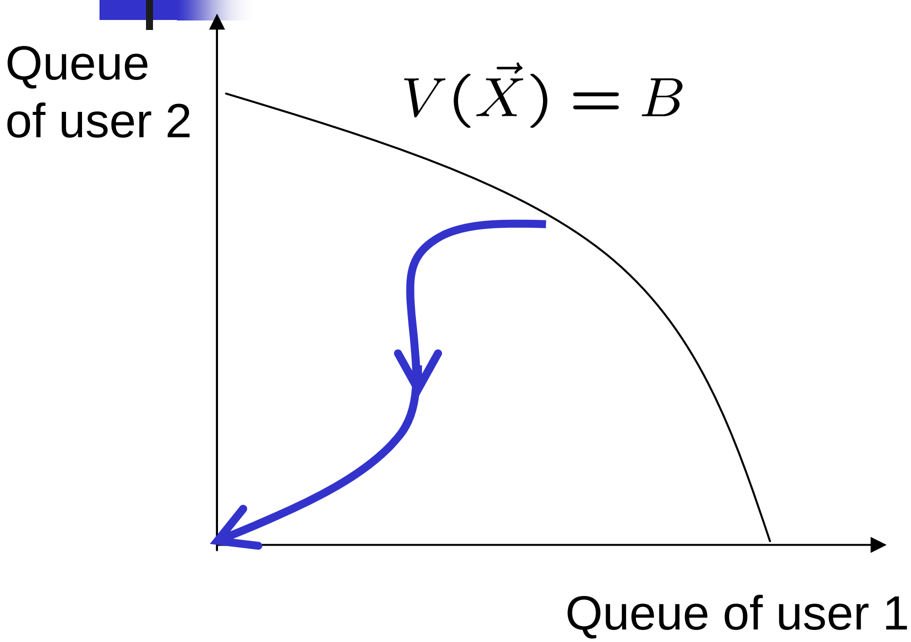

Case 2: Plot of $\mathbf{P}[\max_{1 \leq i \leq N} Q_i \geq B]$ vs $B$ for the $\alpha$-algorithm.

**Hybrid algorithm ($\alpha$ from 1 to 15)**

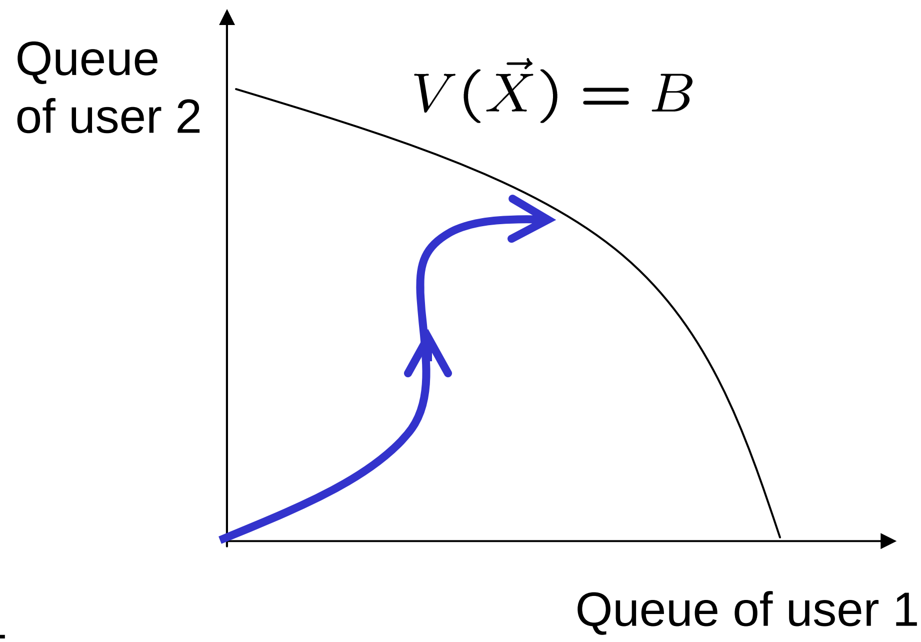
# Outline

- System Model
- Capacity Maximizing Algorithms
- Delay Performance: Main Results
- Practice: Delay-Optimal Control Algorithms
- *Key Idea of Analysis: Large Deviations + Lyapunov Stability*
- Conclusion

# Sample-Path Large Deviations




**Average Behavior**




**Large Deviations Behavior**

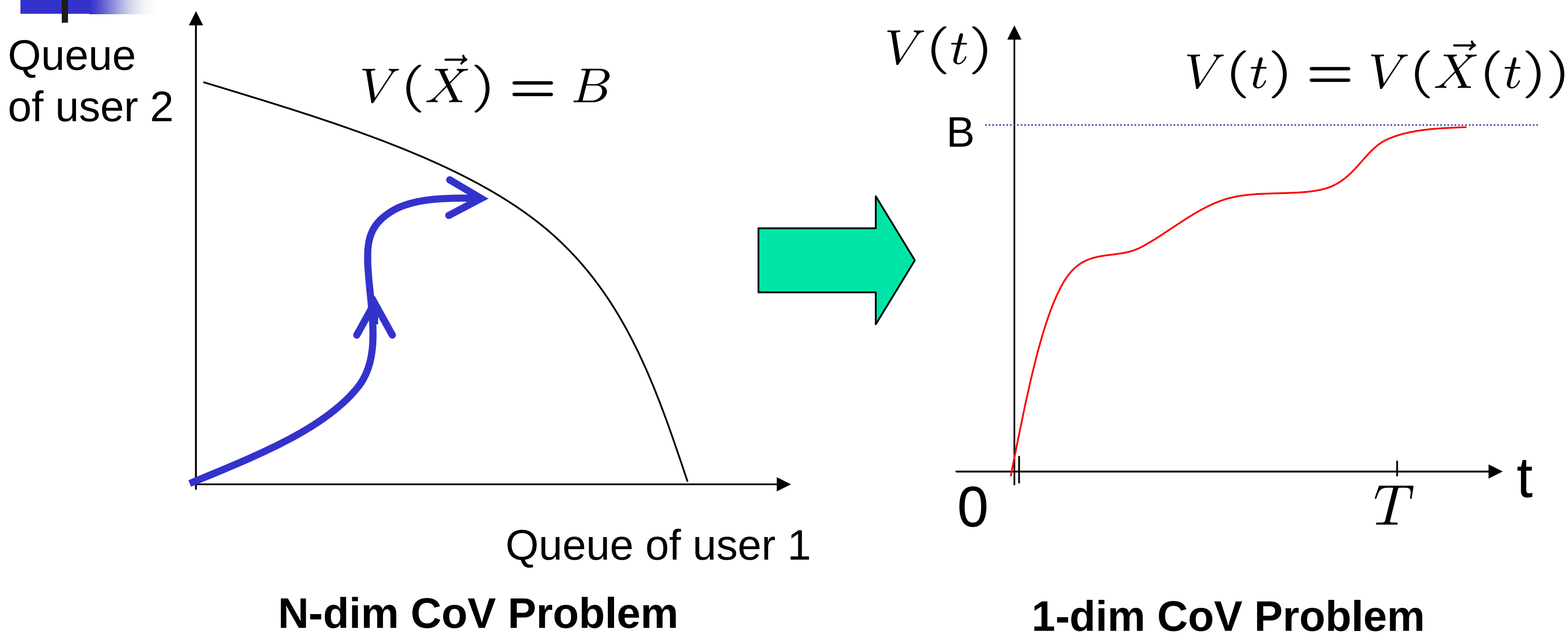
- Each “positive drift” has a non-negative cost $l(\vec{X}, \frac{d}{dt}\vec{X})$
- The decay-rate corresponds to the path with the minimum cost ⇨ “**most likely path to overflow**”

# Finding the Most-Likely Path to Overflow

- A multi-dimensional "Calculus of Variations" (CoV) problem:
  - e.g., what is the shape of soap bubbles?

- Even more difficult when the decision rule is discontinuous
  - e.g., MW-$\alpha$ Policy
  - Existing results restricted to small networks [Shakkottai04, Bertsimas et al 98], or restrictive symmetric case [Ying et al 05].

# Combining Large Deviations with Lyapunov Functions




- We can calculate the cost for V(t) to grow $l_V(V, \frac{dV}{dt})$
- The corresponding 1-dim CoV problem is much easier to solve.

# Outline

- System Model
- Capacity Maximizing Algorithms
- Delay Performance: Main Results
- Practice: Delay-Optimal Control Algorithms
- Key Idea of Analysis: Large Deviations + Lyapunov Stability
- *Conclusion*

# Conclusion

- We have developed a new unified theory for delay-analysis that combines large-deviations with Lyapunov stability
- This new theory can be easily applied to cellular and multi-hop wireless networks
- Practical algorithms with good delay performance can be developed using this approach.

# Related Work

- Large-deviations in wireline networks [Elwalid & Mitra 93, Kesidis et al 93]
- Large-deviations for queue-unaware algorithms [Wu & Negi 03, Eryilmaz& Srikant 04 ]

- Large-deviations for queue-length based algorithms
  - Two users [Shakkottai04, Bertsimas et al 98]
  - Symmetric setting [Ying et al 05]
  - Exponential rule [Stolyar 08]
  - Log rule [Sadiq & De Veciana 08]

- Heavy traffic asymptotes: [Stolyar 04]
  - Usually require complete resource pooling conditions, except [Srikant 09]
- Mean delay analysis [Neely, Gupta & Shroff, Koushik & Saswati]
  - Provides upper and lower bounds
- Sample-path analysis [Tassiulas & Ephremides 94]

# Future Work

- A theory for small delays
  - The hybrid algorithm can be viewed as a way of improving the pre-factor
  - Other largeness regimes
    - Many-channel asymptotes [Bodas et al 09]
    - Heavy-traffic asymptotes [Ji et al 09]
- Delay in multi-hop wireless networks with dynamic routing:
  - Small queue does not mean small delay (due to non-work-conserving)
- Algorithms that are not max-weight, or back-pressure based.

# Thank you!


- V. J. Venkataramanan and X. Lin, "On Wireless Scheduling Algorithms for Minimizing the Queue-Overflow Probability," *IEEE/ACM Transactions on Networking*, to appear.

- V. J. Venkataramanan and X. Lin, "On the Queue-Overflow Probability of Wireless Systems: A New Approach Combining Large Deviations with Lyapunov Functions," submitted to *IEEE Transactions on Information Theory*, 2009.

- V. J. Venkataramanan and X. Lin, "Structural Properties of LDP for Queue-Length Based Wireless Scheduling Algorithms," in *45th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, September 2007

- C. Zhao and X. Lin, "On the Queue-Overflow Probabilities of Distributed Scheduling Algorithms," to appear in *IEEE CDC*, 2009

- Xiaojun Lin and V. J. Venkataramanan, "On the Large-Deviations Optimality of Scheduling Policies Minimizing the Drift of a Lyapunov Function," in *47th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, September 2009