Opportunistic Routing with Congestion Diversity in Multi-hop Wireless Networks

Tara Javidi
Electrical and Computer Engineering
University of California, San Diego

Joint Work with: Parul Gupta, M. Naghshvar, and H. Zhuang

(Acknowledgment: D. Teneketzis, C. Lott)
Opportunism: Multi-hop Wireless Routing
Opportunism: Multi-hop Wireless Routing
Opportunism: Multi-hop Wireless Routing
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
  - path diversity is available in many settings
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
  - path diversity is available in many settings
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
  - path diversity is available in many settings
Opportunism: Multi-hop Wireless Routing

• Wireless provides opportunities and spatial diversity
  • a long hop might be rare but it does happen
  • path diversity is available in many settings
Opportunism: Multi-hop Wireless Routing

- Wireless provides opportunities and spatial diversity
  - a long hop might be rare but it does happen
  - path diversity is available in many settings
- Opportunistically avoiding routing decisions a priori
Opportunistic Routing: Model
Opportunistic Routing: Model

**Model (M1)**

single tx-type, single commodity, with orthogonal channels

[LottTeneketzis, CDC’00], [LottJTeneketzis, SN’02], [Neely, CISS’06]
Opportunistic Routing: Model

**Model (M1)**

single tx-type, single commodity, with orthogonal channels

[LottTeneketzis, CDC’00], [LottJTeneketzis, SN’02], [Neely, CISS’06]

- Network consists of nodes: \( \{1, 2, \ldots, d\} \)
- Packets are destined for \( d \)
  - \( A_t(i) \): # of packets originating at node \( i \) at time \( t \)
  - Bounded and mixing random process with rate \( \lambda_i \)
Opportunistic Routing: Model

**Model (M1)**

single tx-type, single commodity, with orthogonal channels

[LottTenekeetzis, CDC’00], [LottJTenekeetzis, SN’02], [Neely, CISS’06]

- Network consists of nodes: \{1, 2, \ldots, d\}
- Packets are destined for d
  - \(A_t(i)\): # of packets originating at node \(i\) at time \(t\)
  - Bounded and mixing random process with rate \(\lambda_i\)
- Slotted time: Node \(i\) can transmit one packet during a time slot
Opportunistic Routing: Model

Model (M1)

single tx-type, single commodity, with orthogonal channels
[LottTeneketzis, CDC’00], [LottJTeneketzis, SN’02], [Neely, CISS’06]

• Network consists of nodes: \{1, 2, \ldots, d\}
• Packets are destined for \(d\)
  • \(A_t(i)\): # of packets originating at node \(i\) at time \(t\)
  • Bounded and mixing random process with rate \(\lambda_i\)
• Slotted time: Node \(i\) can transmit one packet during a time slot
• Node \(i\)’s tx successfully rcved and acked by subset \(S\) of neighbors
  with probability \(P(S|\hat{i})\) independent of other tx (orthogonal tx)
Opportunistic Routing: Model
Opportunistic Routing: Model

- Opportunistic routing decisions
  - The node responsible, $i$, transmits (locally broadcasts)
  - Nodes $S_t$ successfully decode & acknowledge reception
  - The next action is to 1) choose a neighbor in $S_t$ as the next relay, or 2) retransmit
Opportunistic Routing: Model

• Opportunistic routing decisions
  • The node responsible, $i$, transmits (locally broadcasts)
  • Nodes $S_t$ successfully decode & acknowledge reception
  • The next action is to 1) choose a neighbor in $S_t$ as the next relay, or 2) retransmit

• Distributed: “routing token” + three way hand-shake:
  • Node with the token transmits; upon Ack reception, routing token passed to the next (best) relay while others drop the packet
Opportunistic Routing: Control Objective
Opportunistic Routing: Control Objective

- Routing determines packet departures from $i$ to $j$
Opportunistic Routing: Control Objective

• Routing determines packet departures from $i$ to $j$
  
  • $D_t(i,j)$: # of packet departures from node $i$ to $j$ at time $t$
Opportunistic Routing: Control Objective

- Routing determines packet departures from $i$ to $j$
  - $D_t(i,j)$: # of packet departures from node $i$ to $j$ at time $t$
  - $D_t(i,j) \in \{0,1\}$ and $\sum_j D_t(i,j) \leq 1$
Opportunistic Routing: Control Objective

- Routing determines packet departures from node $i$ to node $j$
  
  - $\mathbb{D}_t(i,j)$: number of packet departures from node $i$ to $j$ at time $t$
  
  - $\mathbb{D}_t(i,j) \in \{0,1\}$ and $\sum_j \mathbb{D}_t(i,j) \leq 1$
  
- Vector of queue backlogs: a stochastic process in $\mathbb{R}^d$

$$q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)$$
Opportunistic Routing: Control Objective

- Routing determines packet departures from \( i \) to \( j \)
  - \( D_t(i,j) \): # of packet departures from node \( i \) to \( j \) at time \( t \)
  - \( D_t(i,j) \in \{0,1\} \) and \( \sum_j D_t(i,j) \leq 1 \)

- Vector of queue backlogs: a stochastic process in \( \mathbb{R}^d \)
  \[
  q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
  \]

- Routing policy controls transitions of this process
Opportunistic Routing: Control Objective

- Routing determines packet departures from node \(i\) to \(j\)
  - \(D_t(i,j)\): number of packet departures from node \(i\) to \(j\) at time \(t\)
  - \(D_t(i,j) \in \{0,1\}\) and \(\sum_j D_t(i,j) \leq 1\)

- Vector of queue backlogs: a stochastic process in \(\mathbb{R}^d\)
  \[
  q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
  \]

- Routing policy controls transitions of this process
  - Markov under a Markov policy \(\pi : \vec{q}_t \times S_t \rightarrow \vec{D}_t\)
Opportunistic Routing: Control Objective

- Routing determines packet departures from \( i \) to \( j \)
  - \( D_t(i,j) \): \# of packet departures from node \( i \) to \( j \) at time \( t \)
  - \( D_t(i,j) \in \{0,1\} \) and \( \sum_j D_t(i,j) \leq 1 \)

- Vector of queue backlogs: a stochastic process in \( \mathbb{R}^d \)
  \[
  q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
  \]

- Routing policy controls transitions of this process
  - Markov under a Markov policy \( \pi : \vec{q}_t \times S_t \to \vec{D}_t \)

Objective:
Opportunistic Routing: Control Objective

- Routing determines packet departures from $i$ to $j$
  - $D_t(i, j)$: # of packet departures from node $i$ to $j$ at time $t$
  - $D_t(i, j) \in \{0, 1\}$ and $\sum_j D_t(i, j) \leq 1$

- Vector of queue backlogs: a stochastic process in $\mathbb{R}^d$
  \[
  q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i, j) \right]^+ + A_t(i) + \sum_j D_t(j, i)
  \]

- Routing policy controls transitions of this process
  - Markov under a Markov policy $\pi : \bar{q}_t \times S_t \rightarrow \bar{D}_t$

Objective:
Find policy with small mean delay, i.e. small $E\left\{ \sum_i q_{t+1}(i) \right\}$
Opportunistic Routing: What is known?
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
  - Unbounded delay in a system with high traffic demand
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
  - Unbounded delay in a system with high traffic demand
- Diversity Backpressure Routing [N’06]
Opportunistic Routing: What is known?

• Opportunistic Shortest Path Routing [LT’00]
  • *Delay optimal* if a max of one packet at a time
  • *Unbounded delay* in a system with high traffic demand

• Diversity Backpressure Routing [N’06]
  • *Throughput optimal* (bounded delay, all admissible traffic)
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - **Delay optimal** if a max of one packet at a time
  - **Unbounded delay** in a system with **high** traffic demand

- Diversity Backpressure Routing [N’06]
  - **Throughput optimal** (bounded delay, **all** admissible traffic)
  - Very large delay in **low to medium** traffic conditions
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
  - Unbounded delay in a system with high traffic demand
- Diversity Backpressure Routing [N’06]
  - Throughput optimal (bounded delay, all admissible traffic)
  - Very large delay in low to medium traffic conditions

... and some “unsuccessful” heuristics doing both [N’07] [YSR’09]
Our Contributions (Outline of the Talk)
Our Contributions (Outline of the Talk)

- Integrate backlog states along short paths
Our Contributions (Outline of the Talk)

- Integrate backlog states along short paths
  - Review of shortest path and backpressure routing algorithms
  - Introducing opportunistic routing with congestion diversity (ORCD)
Our Contributions (Outline of the Talk)

- Integrate backlog states along short paths
  - Review of shortest path and backpressure routing algorithms
  - Introducing opportunistic routing with congestion diversity (ORCD)
- Our contributions
Our Contributions (Outline of the Talk)

• Integrate backlog states along short paths
  • Review of shortest path and backpressure routing algorithms
  • Introducing opportunistic routing with congestion diversity (ORCD)

• Our contributions
  • Significant delay improvements (in simulations)
Our Contributions (Outline of the Talk)

• Integrate backlog states along short paths
  • Review of shortest path and backpressure routing algorithms
  • Introducing opportunistic routing with congestion diversity (ORCD)

• Our contributions
  • Significant delay improvements (in simulations)
  • Throughput optimal (bounded delay under all traffic)
Our Contributions (Outline of the Talk)

- Integrate backlog states along short paths
  - Review of shortest path and backpressure routing algorithms
  - Introducing opportunistic routing with congestion diversity (ORCD)
- Our contributions
  - Significant delay improvements (in simulations)
  - Throughput optimal (bounded delay under all traffic)
  - Proof results in characterizing a general class of policies
Our Contributions (Outline of the Talk)

- Integrate backlog states along short paths
  - Review of shortest path and backpressure routing algorithms
  - Introducing opportunistic routing with congestion diversity (ORCD)

- Our contributions
  - Significant delay improvements (in simulations)
  - Throughput optimal (bounded delay under all traffic)
  - Proof results in characterizing a general class of policies

- Ongoing and future work
  - Practical implementations of ORCD
  - Delay optimality
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
  - Unbounded delay in a system with high traffic demand
- Diversity Backpressure Routing [N’06]
  - Throughput optimal (bounded delay, all admissible traffic)
  - Very large delay in low to medium traffic conditions

... and some “unsuccessful” heuristics doing both [N’07] [YSR’09]
Opportunistic Shortest Path Routing

- Add to (M1) node i’s transmission cost of $c_i$
Opportunistic Shortest Path Routing

- Add to (M1) node \( i \)'s transmission cost of \( c_i \)
- Expected per packet cost, form node \( i \), under \( \pi \)
Opportunistic Shortest Path Routing

• Add to (M1) node $i$’s transmission cost of $c_i$
• Expected per packet cost, form node $i$, under $\pi$

$$J^\pi(i) = E \left\{ \sum_{t=1}^{\tau} c_{i(t)} \right\},$$

where $\tau$ is the termination time; $i(t)$ is node with token at time $t$
Opportunistic Shortest Path Routing

- Add to (M1) node $i$’s transmission cost of $c_i$
- Expected per packet cost, form node $i$, under $\pi$

$$J^{\pi}(i) = E \left\{ \sum_{t=1}^{\tau} c_{i(t)} \right\},$$

where $\tau$ is the termination time; $i(t)$ is node with token at time $t$

**Objective:**
Find a policy that minimizes the expected cost
Opportunistic Shortest Path Routing

- Add to (M1) node \( i \)'s transmission cost of \( c_i \)
- Expected per packet cost, form node \( i \), under \( \pi \)

\[
J^\pi(i) = E\left\{ \sum_{t=1}^{\tau} c_{i(t)} \right\},
\]

where \( \tau \) is the termination time; \( i(t) \) is node with token at time \( t \)

Objective:
Find a policy that minimizes the expected cost

\( c_i = 1 \) recovers the shortest/fastest path routing also known as Extremely Opportunistic Routing (ExOR)
Structural Property of the Optimal Policy

[LottTeneketzis'06]
Structural Property of the Optimal Policy

[LottTeneketzis'06]

- Policies of interest equivalent to total orders of nodes

\[ i \text{ ranks higher than } j \ (j \leq_{\pi^*} i) \iff \pi^*(S_t = \{i, j\}) = i \]
Structural Property of the Optimal Policy

[Lott Teneketzis'06]

• Policies of interest equivalent to total orders of nodes

\[ i \text{ ranks higher than } j \ (j \leq_{\pi^*} i) \iff \pi^* (S_t = \{i, j\}) = i \]

• Under optimal policy, \( \pi^* \), the expected cost to transmit the message from node \( i \) solves:

\[
J_{\pi^*} (i) = C_i + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} J_{\pi^*} (j)
\]

\[
J_{\pi^*} (d) = 0
\]
Structural Property of the Optimal Policy

[LottTeneketzis'06]

- Policies of interest equivalent to total orders of nodes

  \[ i \text{ ranks higher than } j \ (j \leq_{\pi^*} i) \iff \pi^* (S_t = \{i, j\}) = i \]

- Under optimal policy, \( \pi^* \), the expected cost to transmit the message from node \( i \) solves:

  \[
  J^{\pi^*} (i) = C_i + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} J^{\pi^*} (j)
  \]

  \[
  J^{\pi^*} (d) = 0
  \]

- The optimal node ordering and policy is based on optimal expected cost
Structural Property of the Optimal Policy
[LottTeneketzis'06]

• Policies of interest equivalent to total orders of nodes

\[ i \text{ ranks higher than } j \ (j \leq_{\pi^*} i) \iff \pi^* (S_t = \{i, j\}) = i \]

• Under optimal policy, \( \pi^* \), the expected cost to transmit the message from node \( i \) solves:

\[
J_{\pi^*} (i) = C_i + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} J_{\pi^*} (j)
\]

\[
J_{\pi^*} (d) = 0
\]

• The optimal node ordering and policy is based on optimal expected cost

\[
J_{\pi^*} (j) \geq J_{\pi^*} (i) \iff i \text{ ranks higher than } j \ (j \leq_{\pi^*} i)
\]
More than one packet?
More than one packet?

- In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes
More than one packet?

• In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes

• Intuitively, it might not be efficient and/or feasible for all packets to “opportunistically” take the “shortest” path
More than one packet?

• In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes

• Intuitively, it might not be efficient and/or feasible for all packets to “opportunistically” take the “shortest” path
More than one packet?

- In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes.
- Intuitively, it might not be efficient and/or feasible for all packets to “opportunistically” take the “shortest” path.

Example: Let $\lambda_1 = .25$, $\lambda_2 = .3$
More than one packet?

- In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes.
- Intuitively, it might not be efficient and/or feasible for all packets to “opportunistically” take the “shortest” path.

Example: Let $\lambda_1 = 0.25$, $\lambda_2 = 0.3$

Node 1 “opportunistically” routes additional traffic to 2 at a rate of $0.63 \times 0.25 \approx 0.16$. 
More than one packet?

- In a system with high traffic intensity, much of the delay is due to queueing delay at intermediate nodes.
- Intuitively, it might not be efficient and/or feasible for all packets to “opportunistically” take the “shortest” path.

Example: Let $\lambda_1 = .25$, $\lambda_2 = .3$

Node 1 “opportunistically” routes additional traffic to 2 at a rate of .63*.25 ≈ .16

⇒ Unbounded Delay (.3 + .16 > .4)
Opportunistic Routing: What is known?

- Opportunistic Shortest Path Routing [LT’00]
  - Delay optimal if a max of one packet at a time
  - Unbounded delay in a system with high traffic demand
- Diversity Backpressure Routing [N’06]
  - Throughput optimal (bounded delay, all admissible traffic)
  - Very large delay in low to medium traffic conditions

... and some “unsuccessful” heuristics doing both [N’07] [YSR’09]
Opportunistic Backpressure (DIVBAR)  
[TE’93, Neely’06]
Opportunistic Backpressure (DIVBAR) [TE'93, Neely’06]

• Recall: queue backlogs form stochastic process in $\mathbb{R}^d$

$$q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)$$
Opportunistic Backpressure (DIVBAR)

[TE'93, Neely’06]

• Recall: queue backlogs form stochastic process in $\mathbb{R}^d$

$$q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)$$

• Routing policies $\pi: \tilde{q}_t \times \tilde{S}_t \rightarrow \tilde{D}_t$ (rank orderings $j \succ^\pi_k$)
Opportunistic Backpressure (DIVBAR)

[TE’93, Neely’06]

• Recall: queue backlogs form stochastic process in $\mathbb{R}^d$

$$q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)$$

• Routing policies $\pi: \mathbb{Q}_t \times \mathbb{S}_t \rightarrow \mathbb{D}_t$ (rank orderings $j \succ^t \pi k$)

Objective:

Find a policy ensuring queue stability under all admissible traffic
Opportunistic Backpressure (DIVBAR) 
[TE’93, Neely’06]

• Recall: queue backlogs form stochastic process in \( \mathbb{R}^d \)

\[
q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
\]

• Routing policies \( \pi : \bar{q}_t \times \bar{S}_t \rightarrow \bar{D}_t \) (rank orderings \( j \succ_{\pi}^t k \) )

Objective:
Find a policy ensuring queue stability under all admissible traffic

• Queue stability \( \Leftrightarrow \) positive recurrence of \( \bar{o} \), the empty state (or a compact neighborhood of it) \( \Leftrightarrow \) finite \( E\left\{ \sum_i q_{t+1}(i) \right\} \)
Opportunistic Backpressure (DIVBAR)

[TE’93, Neely’06]

• Recall: queue backlogs form stochastic process in $\mathbb{R}^d$

$$q_{t+1}(i) = \left[q_t(i) - \sum_j D_t(i,j)\right]^+ + A_t(i) + \sum_j D_t(j,i)$$

• Routing policies $\pi : \tilde{q}_t \times \tilde{S}_t \rightarrow \tilde{D}_t$ (rank orderings $j >^t \pi k$)

Objective:

Find a policy ensuring queue stability under all admissible traffic

• Queue stability $\iff$ positive recurrence of $\tilde{o}$, the empty state (or a compact neighborhood of it) $\iff$ finite $E\left\{\sum_i q_{t+1}(i)\right\}$

• DIVBAR, $\pi_b^*$, rank-orders nodes based on backlogs

$$q_t(j) < q_t(k) \iff j >^t \pi^* k$$
Opportunistic Backpressure (DIVBAR)

[TE'93, Neely’06]

- Recall: queue backlogs form stochastic process in $\mathbb{R}^d$
  \[ q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i) \]
  - Routing policies $\pi : \tilde{q}_t \times \tilde{S}_t \rightarrow \tilde{D}_t$ (rank orderings $j \succ^\pi k$)

Objective:
Find a policy ensuring queue stability under all admissible traffic

- Queue stability $\Leftrightarrow$ positive recurrence of $\tilde{0}$, the empty state (or a compact neighborhood of it) $\Leftrightarrow$ finite $E\left\{ \sum_i q_{t+1}(i) \right\}$
- DIVBAR, $\pi_b^*$, rank-orders nodes based on backlogs
  \[ q_t(j) < q_t(k) \Leftrightarrow j \succ^*_{\pi_b} k \]
  - Stabilizes queues under all admissible traffic
Delay Performance of DIVBAR
Delay Performance of DIVBAR

- Delay performance of backpressure policy in low to medium traffic can be quite poor!
Delay Performance of DIVBAR

• Delay performance of backpressure policy in low to medium traffic can be quite poor!
• Lack of topology information
Delay Performance of DIVBAR

• Delay performance of backpressure policy in low to medium traffic can be quite poor!

• Lack of topology information
  • pressure away from d
Delay Performance of DIVBAR

- Delay performance of backpressure policy in low to medium traffic can be quite poor!
- Lack of topology information
  - pressure away from d

![Diagram](image)

add the hop count [N’07]
(Enhanced-DIVBAR)
Delay Performance of DIVBAR

• Delay performance of backpressure policy in low to medium traffic can be quite poor!

• Lack of topology information
  • pressure away from d

add the hop count [N’07]
(Enhanced-DIVBAR)
constrained hop count [YSR’07]
Delay Performance of DIVBAR

- Delay performance of backpressure policy in low to medium traffic can be quite poor!
- Lack of topology information
  - pressure away from d

add the hop count [N’07]
(Enhanced-DIVBAR)
constrained hop count [YSR’07]
Delay Performance of DIVBAR

- Delay performance of backpressure policy in low to medium traffic can be quite poor!

- Lack of topology information
  - pressure away from d

add the hop count [N’07]
(Enhanced-DIVBAR)
constrained hop count [YSR’07]

- “hole”-effect
Delay Performance of DIVBAR

• Delay performance of backpressure policy in low to medium traffic can be quite poor!

• Lack of topology information
  • pressure away from d

add the hop count [N’07]
(Enhanced-DIVBAR)
constrained hop count [YSR’07]

• “hole”-effect
• ignoring congestion diversity
Opportunism w/ Congestion Diversity
Opportunism w/ Congestion Diversity

• Delay optimality requires identifying the shortest AND least congested path!
Opportunism w/ Congestion Diversity

• Delay optimality requires identifying the shortest AND least congested path!

Opportunistic Routing w/ Congestion Diversity (ORCD): is a priority policy based on the indices of the nodes

\[ V_t(j) < V_t(k) \iff j >_{\pi^*_c}^t k \]
Opportunism w/ Congestion Diversity

- Delay optimality requires identifying the shortest AND least congested path!

**Opportunistic Routing w/ Congestion Diversity (ORCD):** is a priority policy based on the indices of the nodes

\[ V_t(j) < V_t(k) \iff j >_{\pi_c^*} k \]

where

\[ V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S | i) \min_{j \in S} V_t(j) \]

\[ V_t(d) = 0 \]
Opportunism w/ Congestion Diversity

• Delay optimality requires identifying the shortest AND least congested path!

Opportunistic Routing w/ Congestion Diversity (ORCD): is a priority policy based on the indices of the nodes

\[ V_t(j) < V_t(k) \iff j >_{\pi_c}^t k \]

where

\[ V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j) \]

\[ V_t(d) = 0 \]
Opportunism w/ Congestion Diversity

- Delay optimality requires identifying the shortest AND least congested path!

**Opportunistic Routing w/ Congestion Diversity (ORCD):** is a priority policy based on the indices of the nodes

\[
V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j)
\]

where

\[
V_t(d) = 0
\]

- \( V \approx \) minimum expected drain time assuming time-invariant queues

\[
V_t(j) < V_t(k) \iff j >^*_{\pi_c} k
\]
Simulations: Delay Performance of ORCD
Simulations: Delay Performance of ORCD
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure $\Rightarrow$ large delay

- Three parameters
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay

- Three parameters
  - Hole size, N
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay

- Three parameters
  - Hole size, $N$
  - 1-step diversity, $M$
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay

- Three parameters
  - Hole size, $N$
  - 1-step diversity, $M$
  - 2-step diversity, $K$
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay

- Three parameters
  - Hole size, $N$
  - 1-step diversity, $M$
  - 2-step diversity, $K$

Table 1: Mean Delay from the source node $S$ to the destination node $D$ for different policies. $N=5$, $K=3$. iid traffic with $\lambda_S=0.1$, $\lambda_C=0.8$, $\lambda_{Bij}=0.5$, and $\lambda_A=\lambda_B=\lambda_{Bi}=0$

<table>
<thead>
<tr>
<th>$M$</th>
<th>DIVBAR</th>
<th>E-DIVBAR</th>
<th>ORCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>40.19</td>
<td>107.72</td>
<td>6.21</td>
</tr>
<tr>
<td>3</td>
<td>36.65</td>
<td>138.32</td>
<td>5.69</td>
</tr>
<tr>
<td>5</td>
<td>37.04</td>
<td>182.09</td>
<td>5.45</td>
</tr>
</tbody>
</table>
Simulations: Delay Performance of ORCD

- Routing affects traffic at S
  - locally reducing back pressure ⇒ large delay

- Three parameters
  - Hole size, \( N \)
  - 1-step diversity, \( M \)
  - 2-step diversity, \( K \)

<table>
<thead>
<tr>
<th>( \lambda_S=0.1, \lambda_C=0.8, \lambda_{Bij}=0.5 ) and ( \lambda_A=\lambda_B=\lambda_{Bi}=0 )</th>
<th>DIVBAR</th>
<th>E-DIVBAR</th>
<th>ORCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=2</td>
<td>40.19</td>
<td>107.72</td>
<td>6.21</td>
</tr>
<tr>
<td>M=3</td>
<td>36.65</td>
<td>138.32</td>
<td>5.69</td>
</tr>
<tr>
<td>M=5</td>
<td>37.04</td>
<td>182.09</td>
<td>5.45</td>
</tr>
</tbody>
</table>

Table 1: Mean Delay from the source node S to the destination node D for different policies. \( N=5, K=3 \). iid traffic with
Throughput Optimality of ORCD
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$:

$$V_t(j) < V_t(k) \iff j >^t_{\pi_c^*} k$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$:

\[ V_t(j) < V_t(k) \iff j \succ^*_\pi \ k \]

Policy $\pi_c^*$ (ORCD) is throughput optimal.

NOTE:

• Communication overhead is identical to DIVBAR
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$:

$$V_t(j) < V_t(k) \iff j >^t_{\pi_c^*} k$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.

NOTE:

- Communication overhead is identical to DIVBAR
- Computational complexity associated with solving fixed point

\[
V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j)
\]
Our Contributions (Outline of the Talk)

• Integrate backlog states along short paths
  • Review of shortest path and backpressure routing algorithms
  • Introducing opportunistic routing with congestion diversity (ORCD)

• Our contributions
  • Significant delay improvements (in simulations)
  • Throughput optimal (bounded delay under all traffic)
  • Proof results in characterizing a general class of policies

• Ongoing and future work
  • Practical implementations of ORCD
  • Delay optimality
Routing and Congestion: Revisited

\[ q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i, j) \right]^+ + A_t(i) + \sum_j D_t(j, i) \]
Routing and Congestion: Revisited

- Recall: $d$ dimensional Markov chain of queue backlogs

\[
q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
\]

where

\[
\tilde{D}_t = \pi(\tilde{q}_t \times \tilde{S}_t)
\]
Routing and Congestion: Revisited

- Recall: \(d\) dimensional Markov chain of queue backlogs

\[
q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i,j) \right]^+ + A_t(i) + \sum_j D_t(j,i)
\]

where

\[
\tilde{D}_t = \pi(\tilde{q}_t \times \tilde{S}_t)
\]

Question: What to ensure about this Markov Chain?
Routing and Congestion: Revisited

• Recall: $d$ dimensional Markov chain of queue backlogs

$$q_{t+1}(i) = \left[ q_t(i) - \sum_j D_t(i, j) \right]^+ + A_t(i) + \sum_j D_t(j, i)$$

where

$$\tilde{D}_t = \pi(\tilde{q}_t \times \tilde{S}_t)$$

Question: What to ensure about this Markov Chain?

- The $E\left\{ \sum_{i=1}^d q_t(i) \right\}$ is small $\equiv$ small average delay
Recall: $d$ dimensional Markov chain of queue backlogs

$$q_{t+1}(i) = \left[ q_t(i) - \sum_{j} D_t(i, j) \right]^{+} + A_t(i) + \sum_{j} D_t(j, i)$$

where

$$\tilde{D}_t = \pi(\bar{q}_t \times \bar{S}_t)$$

Question: What to ensure about this Markov Chain?

- The $E\left\{ \sum_{i=1}^{d} q_t(i) \right\}$ is small $\equiv$ small average delay
- Markov chain positive recurrent $\equiv$ queues infinitely often empty $\equiv$ throughput optimality of $\pi^* \equiv$ finite $E\left\{ \sum_{i=1}^{d} q_t(i) \right\}$
Consequence of Foster–Lyapunov Thm
Consequence of Foster–Lyapunov Thm

- Positive recurrence of MC guaranteed if there exists a Lyapunov function $L : \bar{q} \to \mathbb{R}^+$ with expected negative drift
Consequence of Foster–Lyapunov Thm

• Positive recurrence of MC guaranteed if there exists a Lyapunov function \( L : \bar{q} \rightarrow \mathbb{R}^+ \) with expected negative drift

• Given a routing policy, find a Lyapunov function with negative drift
Consequence of Foster–Lyapunov Thm

• Positive recurrence of MC guaranteed if there exists a Lyapunov function $L : \bar{q} \rightarrow \mathbb{R}^+$ with expected negative drift
  • Given a routing policy, find a Lyapunov function with negative drift
  • Reverse engineering throughput optimal schemes
Consequence of Foster–Lyapunov Thm

• Positive recurrence of MC guaranteed if there exists a Lyapunov function $L : \vec{q} \rightarrow \mathbb{R}^+$ with expected negative drift

  • Given a routing policy, find a Lyapunov function with negative drift
  
  • Reverse engineering throughput optimal schemes

  • Quadratic Lyapunov function $L_1(\vec{q}_t) := \sum_{i=1}^{d-1} q_t^2(i)$
Consequence of Foster–Lyapunov Thm

• Positive recurrence of MC guaranteed if there exists a Lyapunov function \( L : \bar{q} \rightarrow \mathbb{R}^+ \) with expected negative drift

  • Given a routing policy, find a Lyapunov function with negative drift

  • Reverse engineering throughput optimal schemes

    • Quadratic Lyapunov function \( L_1(q_t) := \sum_{i=1}^{d-1} q_t^2(i) \)

    • Backpressure maximizes expected negative drift of \( L_1 \)

  • Often, maximizing the negative drift in \( L_1 \) is what causes delay!!
Consequence of Foster–Lyapunov Thm

- Positive recurrence of MC guaranteed if there exists a Lyapunov function $L : \vec{q} \rightarrow \mathbb{R}^+$ with expected negative drift
  - Given a routing policy, find a Lyapunov function with negative drift
  - Reverse engineering throughput optimal schemes
    - Quadratic Lyapunov function $L_1(\vec{q}_t) := \sum_{i=1}^{d-1} q_t^2(i)$
    - Backpressure maximizes expected negative drift of $L_1$
    - Often, maximizing the negative drift in $L_1$ is what causes delay!!
A Class of Lyapunov Functions
A Class of Lyapunov Functions

- What happens if $L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
What happens if \( L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i) \)?

When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in \( L_2 \)!
A Class of Lyapunov Functions

- What happens if \( L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i) \)

- When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in \( L_2 \)!
A Class of Lyapunov Functions

• What happens if $L_2(\bar{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  - When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  - Not possible to ensure negative drift if a queue is empty!

\[ L_2(\bar{q}_t) = \sum_{i=1}^{d-1} q_t(i) \]
A Class of Lyapunov Functions

- What happens if $L_2(\bar{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  - When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  - Not possible to ensure negative drift if a queue is empty!

- Goal: search for a piecewise quadratic/linear function
A Class of Lyapunov Functions

• What happens if $L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  • When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  • Not possible to ensure negative drift if a queue is empty!

• Goal: search for a piecewise quadratic/linear function
  • A good candidate (2D): paste together $L_1$ and $L_2$
A Class of Lyapunov Functions

• What happens if $L_2(\bar{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  • When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  • Not possible to ensure negative drift if a queue is empty!

• Goal: search for a piecewise quadratic/linear function
  • A good candidate (2D): paste together $L_1$ and $L_2$
A Class of Lyapunov Functions

- What happens if $L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  - When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  - Not possible to ensure negative drift if a queue is empty!

- Goal: search for a piecewise quadratic/linear function
  - A good candidate (2D): paste together $L_1$ and $(L_2)^2$
A Class of Lyapunov Functions

• What happens if $L_2(\vec{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  • When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!

• Not possible to ensure negative drift if a queue is empty!

$L_f(\vec{q}) = \sum_i L_i(\vec{q})1_{\{\vec{q} \in K_i\}}$

• Goal: search for a piecewise quadratic/linear function
  • A good candidate (2D): paste together $L_1$ and $(L_2)^2$
A Class of Lyapunov Functions

• What happens if $L_2(\bar{q}_t) := \sum_{i=1}^{d-1} q_t(i)$
  
• When all queues sufficiently backlogged, any non-idling policy ensures a negative drift in $L_2$!
  
• Not possible to ensure negative drift if a queue is empty!

\[ L_f(\bar{q}) = \sum_i L_i(\bar{q}) \mathbf{1}_{\{\bar{q} \in K_i\}} \]

• Goal: search for a piecewise quadratic/linear function

  • A good candidate (2D): paste together $L_1$ and $(L_2)^2$ in a smooth manner
P–W Quadratic Lyapunov Functions

• Generalize to higher dimension
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[
\begin{align*}
Q_1 & \\
Q_2 & \\
Q_3 &
\end{align*}
\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[(Q_1 + Q_2 + Q_3)^2\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[(Q_1 + Q_2 + Q_3)^2\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[ (Q_1 + Q_2 + Q_3)^2 \]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[
(Q_1 + Q_2 + Q_3)^2
\]

\[
Q_2^2 + (Q_1 + Q_3)^2
\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[(Q_1 + Q_2 + Q_3)^2\]

\[Q_2^2 + (Q_1 + Q_3)^2\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[(Q_1 + Q_2 + Q_3)^2\]

\[Q_2^2 + (Q_1 + Q_3)^2\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension

\[(Q_1 + Q_2 + Q_3)^2\]

\[(Q_2 + Q_3)^2 + Q_1^2\]

\[Q_1^2 + Q_2^2 + Q_3^2\]

\[Q_1^2 + Q_2^2 + Q_3^2\]

\[Q_2^2 + (Q_1 + Q_3)^2\]
P–W Quadratic Lyapunov Functions

- Generalize to higher dimension
P–W Quadratic Lyapunov Functions

• Generalize to higher dimension

Piece-wise quadratic
Lyapunov function

\[ L_f(\tilde{q}) = \sum L_i(\tilde{q}) 1_{\{\tilde{q} \in K_i\}} \]
P–W Quadratic Lyapunov Functions

- Need to guarantee continuity and differentiability

Piece-wise quadratic smooth Lyapunov function

\[ L_f(\bar{q}) = \sum_i L_i(\bar{q})1_{\{\bar{q} \in K_i\}} \]
P–W Quadratic Lyapunov Functions

- Need to guarantee continuity and differentiability

Piece-wise quadratic smooth Lyapunov function

\[ L_f(\vec{q}) = \sum_i L_i(\vec{q})1_{\{\vec{q} \in K_i\}} \]

\[ f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2 \]

\[ f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2 \]

\[ f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2 \]

\[ f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2 \]

\[ f(0,1)Q_2^2 + f(1,2)(Q_1 + Q_3)^2 \]

\[ f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2 \]

\[ f(0,1)Q_1^2 + f(0,2)(Q_1 + Q_3)^2 + f(1,1)Q_3^2 \]
P–W Quadratic Lyapunov Functions

- Need to guarantee continuity and differentiability carefully pick coefficients \( f(.,.) \)

Piece-wise quadratic smooth Lyapunov function

\[
L_f(q) = \sum_i L_i(q)1_{\{q \in K_i\}}
\]
P–W Quadratic Lyapunov Functions

- Sufficient conditions to ensure smoothness:

Piece-wise quadratic smooth Lyapunov function

\[ L_f(\vec{q}) = \sum_i L_i(\vec{q}) \mathbf{1}_{\{\vec{q} \in K_i\}} \]

\[ f(m,n_1) \geq f(m+n_1,n_2) \]

\[ \frac{1}{f(m,n_1 + n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m+n_1,n_2)} \]

\[ f(0,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2 \]

\[ f(0,1)Q_1^2 + f(2,1)Q_2^2 \]

\[ f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2 \]

\[ f(0,1)Q_1^2 + f(2,1)Q_2^2 \]

\[ f(0,3)(Q_1 + Q_2 + Q_3)^2 \]

\[ f(0,1)Q_1^2 + f(2,1)(Q_1 + Q_2)^2 \]

\[ f(2,1)Q_1^2 + f(1,2)(Q_1 + Q_2)^2 \]

\[ f(0,1)Q_1^2 + f(2,1)Q_2^2 \]

\[ f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2 \]

\[ f(0,1)Q_1^2 + f(1,2)(Q_1 + Q_3)^2 \]

\[ f(1,1)Q_1^2 + f(0,1)Q_3^2 \]
P–W Quadratic Lyapunov Functions

- Sufficient conditions to ensure smoothness:

Piece-wise quadratic smooth Lyapunov function

\[ L_f(\tilde{q}) = \sum_i L_i(\tilde{q}) \textbf{1}_{\{\tilde{q} \in K_i\}} \]

\[ f(m, n_1) \geq f(m + n_1, n_2) \]

\[ \frac{1}{f(m, n_1 + n_2)} = \frac{1}{f(m, n_1)} + \frac{1}{f(m + n_1, n_2)} \]

\[ f(0, 2)(Q_1 + Q_3)^2 + f(2, 1)Q_2^2 \]

\[ f(0, 1)Q_1^2 + f(2, 1)Q_2^2 \]

\[ f(0, 1)Q_1^2 + f(0, 1)Q_3^2 \]

\[ f(2, 1)Q_1^2 + f(1, 2)(Q_1 + Q_2)^2 \]

\[ f(0, 2)(Q_2 + Q_3)^2 + f(2, 1)Q_1^2 \]

\[ f(0, 3)(Q_1 + Q_2 + Q_3)^2 \]

\[ f(0, 1)Q_1^2 + f(1, 2)(Q_1 + Q_3)^2 \]
Minimizing the Expected Drift of $L_f(\vec{q})$

$$L_f(\vec{q}) = \sum_i L_i(\vec{q}) 1_{\{\vec{q} \in K_i\}}$$
Minimizing the Expected Drift of $L_f(\bar{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\bar{q}) = \sum_i L_i(\bar{q}) 1_{\{\bar{q} \in K_i\}}$$

where

$$L_f(q) = f(0,1)Q_1^2 + f(1,2)Q_1Q_2 + f(2,1)Q_2^2$$

and

$$L_f(\bar{q}) = f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2$$

and

$$L_f(q) = f(0,1)Q_1^2 + f(1,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\bar{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\bar{q}) = \sum_i L_i(\bar{q})1_{\{\bar{q} \in K_i\}}$$

$$f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2$$

$$f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2$$

$$f(2,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2$$

$$f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_3)^2$$

$$f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2$$

$$f(0,1)Q_1^2 + f(1,2)(Q_2 + Q_3)^2$$

$$f(0,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(0,2)(Q_1 + Q_2)^2 + f(2,1)Q_3^2$$

$$f(0,1)Q_2^2 + f(1,2)(Q_1 + Q_3)^2$$

$$f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$

$$f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\vec{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\vec{q}) = \sum_i L_i(\vec{q}) 1_{\{\vec{q} \in K_i\}}$$

$$= f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$+ f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2$$

$$+ f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2$$

$$+ f(2,1)Q_1^2 + f(1,1)Q_2^2$$

$$+ f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2 + f(1,1)Q_3^2$$

$$+ f(0,1)Q_1^2 + f(1,2)(Q_2 + Q_3)^2$$

$$+ f(0,2)(Q_1 + Q_2)^2 + f(2,1)Q_3^2$$

$$+ f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$

$$+ f(0,1)Q_2^2 + f(1,2)(Q_1 + Q_3)^2$$

$$+ f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\tilde{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\tilde{q}) = \sum_i L_i(\tilde{q}) 1_{\{\tilde{q} \in K_i\}}$$
Minimizing the Expected Drift of $L_f(\vec{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\vec{q}) = \sum_i L_i(\vec{q})\mathbf{1}_{\{\vec{q} \in K_i\}}$$

$$f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$(\{1,2,3\})$$
Minimizing the Expected Drift of $L_f(\tilde{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\tilde{q}) = \sum_i L_i(\tilde{q}) \mathbf{1}_{\{\tilde{q} \in K_i\}}$$

$$f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2$$

$$f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2$$

$$f(2,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(0,1)Q_2^2 + f(1,2)(Q_1 + Q_3)^2$$

$$(\{1,2,3\})$$

$$f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2$$

$$f(0,1)Q_1^2 + f(2,1)Q_2^2 + f(1,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,2)(Q_2 + Q_3)^2$$

$$f(0,2)(Q_1 + Q_2)^2 + f(2,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,2)Q_1 + Q_3^2$$

$$f(0,1)Q_2^2 + f(1,1)Q_2^2$$

$$f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2$$

$$f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\bar{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\bar{q}) = \sum_i L_i(\bar{q})1_{\{\bar{q} \in K_i\}}$$

$$f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2$$

$$f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2$$

$$f(2,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,2)(Q_1 + Q_3)^2$$

$$f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2$$

$$f(0,1)Q_1^2 + f(1,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(0,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\bar{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\bar{q}) = \sum_i L_i(\bar{q})\mathbb{1}_{\{\bar{q} \in K_i\}}$$

$$f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2$$

$$f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2$$

$$f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2$$

$$f(2,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2$$

$$(\{1,2,3\})$$

$$(\{2\}, \{1,3\})$$

$$f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2$$

$$f(0,1)Q_1^2 + f(2,1)Q_2^2 + f(1,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,2)(Q_2 + Q_3)^2$$

$$f(0,2)(Q_1 + Q_2)^2 + f(2,1)Q_3^2$$

$$f(0,1)Q_1^2 + f(1,1)Q_2^2$$

$$f(1,1)Q_1^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2$$
Minimizing the Expected Drift of $L_f(\tilde{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\tilde{q}) = \sum_i L_i(\tilde{q}) 1_{\{\tilde{q} \in K_i\}}$$

\[ f(1,1)Q_1^2 + f(2,1)Q_2^2 + f(0,1)Q_3^2 \]

\[ f(0,1)Q_3^2 + f(1,2)(Q_1 + Q_2)^2 \]

\[ f(0,2)(Q_2 + Q_3)^2 + f(2,1)Q_1^2 \]

\[ f(2,1)Q_1^2 + f(1,1)Q_2^2 \]

\[ f(2,1)Q_1^2 + f(0,1)Q_2^2 + f(1,1)Q_3^2 \]

\[ f(0,2)(Q_1 + Q_3)^2 + f(2,1)Q_2^2 \]

\[ f(0,1)Q_1^2 + f(2,1)Q_2^2 + f(1,1)Q_3^2 \]

\[ f(0,1)Q_1^2 + f(1,2)(Q_2 + Q_3)^2 \]

\[ f(0,2)(Q_1 + Q_2)^2 + f(2,1)Q_3^2 \]

\[ f(0,1)Q_1^2 + f(1,1)Q_2^2 \]

\[ f(0,1)Q_1^2 + f(1,1)Q_2^2 \]

\[ f(0,1)Q_2^2 + f(0,1)Q_2^2 + f(2,1)Q_3^2 \]

\[ f(1,1)Q_1^2 + f(0,1)Q_3^2 + f(1,1)Q_3^2 \]
Minimizing the Expected Drift of $L_f(\bar{q})$

By construction, one can identify a rank ordering minimizing the drift in

$$L_f(\bar{q}) = \sum_i L_i(\bar{q})\mathbf{1}_{\{\bar{q} \in K_i\}}$$
Theorem [ZNJ ’09] Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that continuous

(i) $f(m,n_1) \geq f(m + n_1,n_2)$

(ii) \[
\frac{1}{f(m,n_1 + n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m + n_1,n_2)}
\]

(iii) Lyapunov function $L_f(\cdot)$ has a negative expected drift.

Then policy $\pi$ is throughput optimal.
A Class of Throughput Optimal Policies

**Theorem [ZNJ '09]** Consider ANY routing policy, \( \pi \), for which there exists a function \( f \) such that continuous

(i) \( f(m, n_1) \geq f(m + n_1, n_2) \)

(ii) \[ \frac{1}{f(m, n_1 + n_2)} = \frac{1}{f(m, n_1)} + \frac{1}{f(m + n_1, n_2)} \] \( \Rightarrow \) \( L_f(\cdot) \) continuous

(iii) Lyapunov function \( L_f(\cdot) \) has a negative expected drift.

Then policy \( \pi \) is throughput optimal.
A Class of Throughput Optimal Policies

**Theorem [ZNJ '09]** Consider ANY routing policy, \( \pi \), for which there exists a function \( f \) such that continuous

(i) \( f(m,n_1) \geq f(m + n_1, n_2) \)

(ii) \[
\frac{1}{f(m,n_1 + n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m + n_1,n_2)}
\]

\( \Rightarrow L_f(\cdot) \) continuous

(iii) Policy \( \pi \) consistent with given rank ordering.

Then policy \( \pi \) is throughput optimal.
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that continuous

(i) $f(m,n_1) \geq f(m+n_1,n_2)$

(ii) $\frac{1}{f(m,n_1+n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m+n_1,n_2)}$ \Rightarrow \text{$L_f(\cdot)$ continuous}$

(iii) *Policy $\pi$ consistent with given rank ordering.*

Then policy $\pi$ is throughput optimal.

Note:

- Size of the cones depend on $f$
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, \( \pi \), for which there exists a function \( f \) such that continuous

(i) (ii) (iii) Policy \( \pi \) consistent with given rank ordering.

Then policy \( \pi \) is throughput optimal.

Note:

- Size of the cones depend on \( f \)

Example:

\[
f(m,n) = \frac{1}{K^{m+n} - K^m}
\]

\( K = 3 \) \( K = 5 \)
A Class of Throughput Optimal Policies

Theorem [ZNJ '09] Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that continuous

$\begin{align*}
\text{(i)} & \quad f(m, n) = \frac{1}{K^{m+n} - K^m} \\
\text{(ii)} & \quad \text{Policy } \pi \text{ consistent with given rank ordering.}
\end{align*}$

$\begin{align*}
\text{Note:} & \quad \text{Size of the cones depend on } f \\
& \quad \text{In much of the state space, non-idling is sufficient}
\end{align*}$

Example:

$K = 3$

$K = 5$
Path-Connected Routing
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
  - When 2 has small backlog, to minimize drift, routing decisions create disconnected network!

![Graph with nodes 0, 1, 2, 3 and edges with weights 0.6 and 0.8]
Path–Connected Routing

• Negative drift in $L_f$ still causes unnecessary delay!!
  
  • When 2 has small backlog, to minimize drift, routing decisions create disconnected network!

```
+-----+-----+-----+
|  2  |  1  |  0  |
+-----+-----+-----+
|  3  |     |     |
+-----+-----+-----+
```

```
+-----+-----+-----+
|  2  |  1  |  0  |
+-----+-----+-----+
|  3  |     |     |
+-----+-----+-----+
```

$Q_1$, $Q_2$, $Q_3$
Path-Connected Routing

• Negative drift in $L_f$ still causes unnecessary delay!!
  • When 2 has small backlog, to minimize drift, routing decisions create disconnected network!

\[
f(0,1)q_2^2 + f(1,2)(q_1 + q_3)^2\]
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
  - When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
Positive drift in $L_f$ still causes unnecessary delay!!

- When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
Path-Connected Routing

• Negative drift in $L_f$ still causes unnecessary delay!!
  • When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
  - When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
- Fix: only cones associated w/ connected routes
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
  - When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
- Fix: only cones associated w/ connected routes
Path-Connected Routing

- Negative drift in $L_f$ still causes unnecessary delay!!
- When 2 has small backlog, to minimize drift, routing decisions create disconnected network!
- Fix: only cones associated with connected routes

\[ L^c_f(\bar{q}) = \sum_{K_i : \text{path-connected}} L_i(\bar{q})1_{\{\bar{q} \in K_i\}}q_i \]
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that continuous

(i) $f(m,n_1) \geq f(m+n_1,n_2)$

(ii) $\frac{1}{f(m,n_1+n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m+n_1,n_2)}$

(iii) Lyapunov function $L_f^c(\cdot)$ has a negative expected drift.

Then policy $\pi$ is throughput optimal.
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, \( \pi \), for which there exists a function \( f \) such that continuous

(i) \[ f(m,n_1) \geq f(m+n_1,n_2) \]

(ii) \[ \frac{1}{f(m,n_1+n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m+n_1,n_2)} \]

(iii) Lyapunov function \( L_f^c(\cdot) \) has a negative expected drift.

Then policy \( \pi \) is throughput optimal.

• For a given \( f \), reverse engineer throughput optimal policy \( \pi^*_c \)
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that: continuous

(i) $f(m,n_1) \geq f(m+n_1,n_2)$

(ii) $\frac{1}{f(m,n_1 + n_2)} = \frac{1}{f(m,n_1)} + \frac{1}{f(m+n_1,n_2)}$

(iii) Lyapunov function $L^c_f(\cdot)$ has a negative expected drift.

Then policy $\pi$ is throughput optimal.

- For a given $f$, reverse engineer throughput optimal policies $\pi^*_c$. class of
A Class of Throughput Optimal Policies

**Theorem [ZNJ ’09]** Consider ANY routing policy, $\pi$, for which there exists a function $f$ such that continuous

(i) $f(m, n_1) \geq f(m + n_1, n_2)$

(ii) $\frac{1}{f(m, n_1 + n_2)} = \frac{1}{f(m, n_1)} + \frac{1}{f(m + n_1, n_2)}$

(iii) Lyapunov function $L_f^c(\cdot)$ has a negative expected drift.

Then policy $\pi$ is throughput optimal.

- For a given $f$, reverse engineer throughput optimal policies $\pi^*_c$
- More useful: for a given policy $\pi^*_c$, find function $f$: $E\{\Delta L_f^c(\cdot)\} < 0$. 

Friday, January 8, 2010
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$, i.e.

$$V_t(j) < V_t(k) \iff j \succ^t_{\pi_c^*} k,$$

where

$$V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j)$$

$$V_t(d) = 0.$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.
Throughput Optimality of ORCD

Theorem [ZNJ '09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$, i.e.

$$V_t(j) < V_t(k) \iff j >_{\pi_c^*} k,$$

where

$$V_t(i) = q_t(i) + \sum_{S \in \Omega} P(S \mid i) \min_{j \in S} V_t(j)$$

$$V_t(d) = 0.$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.

- Sufficient to show when $f(m,n) = \frac{1}{K^{m+n} - K^m}$ and K large under $\pi_c^*$:

$$\mathbb{E}\left\{ L_f^c(\tilde{q}_{t+1}) - L_f^c(\tilde{q}_t) \mid \tilde{q}_t \right\} < 0$$
Throughput Optimality of ORCD

**Theorem [ZNJ ’09]** Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$, i.e.

$$V_t(j) < V_t(k) \iff j >^t_{\pi_c^*} k,$$

where

$$V_t(i) = q_t(i) + \sum_{S \in \Omega} P(S \mid i) \min_{j \in S} V_t(j)$$

$$V_t(d) = 0.$$  

Policy $\pi_c^*$ (ORCD) is throughput optimal.

- Sufficient to show when $f(m,n) = \frac{1}{K^{m+n} - K^m}$ and $K$ large under $\pi_c^*$:

$$E \left\{ L_f^c(\tilde{q}_{t+1}) - L_f^c(\tilde{q}_t) \mid \tilde{q}_t \right\} < 0$$

satisfied if $\pi_c^*$ consistent with the given rank ordering.
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$, i.e.

$$V_t(j) < V_t(k) \iff j >^t_{\pi_c^*} k,$$

where

$$V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j)$$

$$V_t(d) = 0.$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.

- Sufficient to show when $f(m,n) = \frac{1}{K^{m+n} - K^m}$ and K large under $\pi_c^*$:

$$E \left\{ L_f^c(\tilde{q}_{t+1}) - L_f^c(\tilde{q}_t) \mid \tilde{q}_t \right\} < 0$$

if packets are routed from heavy to light groups.
Throughput Optimality of ORCD

Theorem [ZNJ ’09] Let $\pi_c^*$ be a routing policy under which the (time-varying) priority is given by the congestion vector $V_t$, i.e.

$$V_t(j) < V_t(k) \iff j > \pi_c^* k,$$

where

$$V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} V_t(j)$$

$$V_t(d) = 0.$$

Policy $\pi_c^*$ (ORCD) is throughput optimal.

- Sufficient to show when $f(m,n) = \frac{1}{K^{m+n} - K^m}$ and $K$ large under $\pi_c^*$:

$$\mathbb{E}\left\{L_c^f(\bar{q}_{t+1}) - L_c^f(\bar{q}_t) \mid \bar{q}_t\right\} < 0$$

if packets are routed from heavy to light groups.
Our Contributions (Outline of the Talk)

• Integrate backlog states along short paths
  • Review of shortest path and backpressure routing algorithms
  • Introducing opportunistic routing with congestion diversity (ORCD)

• Our contributions
  • Significant delay improvements (in simulations)
  • Throughput optimal (bounded delay under all traffic)
  • Proof results in characterizing a general class of policies

• Ongoing and Future work
  • Practical implementations of ORCD
  • Delay optimality
ORCD: Extensions and Practical Issues

- Simple extensions: multi-rate and multi-commodity
- Interference: scheduled MAC vs. random access
- Communication Overhead
  - Ack explosion: limiting neighbor set
  - Information dissemination rate
- Computation of congestion measure
Ongoing Research: Extensions of ORCD
Ongoing Research: Extensions of ORCD

• It is possible to account and optimize other degrees of freedom
Ongoing Research: Extensions of ORCD

• It is possible to account and optimize other degrees of freedom
Ongoing Research: Extensions of ORCD

• It is possible to account and optimize other degrees of freedom
  
  • $\nu \in \mathbb{U}$ may represent choice of transmission rate and power
    • Trade-off between how far vs. how reliable
    • $\nu \in \mathbb{U}$ may represent choice of neighbor set (ack overhead)
Ongoing Research: Extensions of ORCD

• It is possible to account and optimize other degrees of freedom

\[ V_t(i) = \min_{u \in U} q_t(i)T(u) + \sum_{S \subset \Omega} P(S \mid i,u) \min_{j \in S} V_t(j) \]

• \( u \in U \) may represent choice of transmission rate and power
  • Trade-off between how far vs. how reliable
• \( u \in U \) may represent choice of neighbor set (ack overhead)
  • When \(|u| = 1\), no overhearing (\( \Rightarrow \) traditional dynamic routing)
  • Trade-off among diversity, overhead cost, congestion, and reliability

\[ V_t(k) < V_t(j) \iff k >_{\pi}^t j \]
Ongoing Research: Extensions of ORCD

• It is possible to account and optimize other degrees of freedom

\[ V_t (i) = \min_{u \in U} q_t (i) T(u) + \sum_{S \subset \Omega} P(S \mid i, u) \min_{j \in S} V_t (j) \]

• \( u \in U \) may represent choice of transmission rate and power
  • Trade-off between how far vs. how reliable
• \( u \in U \) may represent choice of neighbor set (ack overhead)
  • When \( |u| = 1 \), no overhearing \((\Rightarrow \) traditional dynamic routing\))
  • Trade-off among diversity, overhead cost, congestion, and reliability

• A node only requires ordering of its neighbors

\[ V_t (k) < V_t (j) \iff k >^{t}_\pi j \]
Ongoing Research: Extensions of ORCD
Ongoing Research: Extensions of ORCD

- Interference Issues
Ongoing Research: Extensions of ORCD

• Interference Issues

  • Scheduling can be combined with routing
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
  • Congestion-based CSMA for random access?
Ongoing Research: Extensions of ORCD

• Interference Issues

  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
  • Congestion-based CSMA for random access?
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
  • Congestion-based CSMA for random access?

• Multiple Destinations
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
  • Congestion-based CSMA for random access?

• Multiple Destinations
  • Multi-commodity version; separate queues per destination \( q^k_t (i) \)
Ongoing Research: Extensions of ORCD

• Interference Issues
  • Scheduling can be combined with routing
  • (Centralized) scheduling and routing while avoid interference
  • Tx scheduled in order to maximize the weighted congestion
  • Congestion-based CSMA for random access?

• Multiple Destinations
  • Multi-commodity version; separate queues per destination $q_t^k(i)$
  • Congestion to destination $k$
Ongoing Research: Extensions of ORCD

• Interference Issues
  - Scheduling can be combined with routing
  - (Centralized) scheduling and routing while avoid interference
  - Tx scheduled in order to maximize the weighted congestion
  - Congestion-based CSMA for random access?

• Multiple Destinations
  - Multi-commodity version; separate queues per destination $q^k_t(i)$
  - Congestion to destination $k$

$$V^k_t(i) = \min_{u \in U_k} q^k_t(i)T(u) + \sum_{S \subseteq \Omega} P(S \mid i, u) \min_{j \in S} V^k_t(j)$$
Ongoing Research: Complexity of ORCD
Ongoing Research: Complexity of ORCD

- A node only requires ordering of its neighbors

\[ V_t(k) < V_t(j) \iff k > _{\pi_i} j \]
Ongoing Research: Complexity of ORCD

- A node only requires ordering of its neighbors
  \[ V_t(k) < V_t(j) \iff k >_{\pi_i} j \]
- Congestion measure solves
  \[ V_t(i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} V_t(j) \]
Ongoing Research: Complexity of ORCD

- A node only requires ordering of its neighbors

\[ V_t(k) < V_t(j) \iff k > \pi_i j \]

- Congestion measure solves

\[ V_t(i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} V_t(j) \]

- Centralized/Iterative Computations \( V_t(j) \)
Ongoing Research: Complexity of ORCD

• A node only requires ordering of its neighbors

\[ V_t(k) < V_t(j) \iff k >_{\pi_i} j \]

• Congestion measure solves

\[ V_t(i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} V_t(j) \]

• Centralized/Iterative Computations \( V_t(j) \)
  • Stochastic variants of Dijkstra and or Bellman-Ford
Ongoing Research: Complexity of ORCD

• A node only requires ordering of its neighbors
  \[ V_t(k) < V_t(j) \iff k > \pi_i j \]

• Congestion measure solves
  \[ V_t(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S | i) \min_{j \in S} V_t(j) \]

• Centralized/Iterative Computations \( V_t(j) \)
  • Stochastic variants of Dijkstra and or Bellman-Ford
  • Infrequent updates of
    \[ \tilde{V}_t(i) = q_{\lfloor t/K \rfloor}(i) + \sum_{S \subseteq \Omega} P(S | i) \min_{j \in S} \tilde{V}_t(j) \]
Ongoing Research: Complexity of ORCD

- A node only requires ordering of its neighbors
  \[ V_t(k) < V_t(j) \iff k > \pi_i j \]

- Congestion measure solves
  \[ V_t(i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} V_t(j) \]

- Centralized/Iterative Computations \( V_t(j) \)
  - Stochastic variants of Dijkstra and or Bellman-Ford
  - Infrequent updates of
    \[ \tilde{V}_t(i) = q_{[t/K]}(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} \tilde{V}_t(j) \]
  - Throughput optimality at arbitrary low overhead

Friday, January 8, 2010
Ongoing Research: Distributed ORCD
Ongoing Research: Distributed ORCD

- A node only requires ordering of its neighbors

\[ \tilde{V}_t^i(k) < \tilde{V}_t^i(j) \iff k >_{\pi_i}^t j \]
Ongoing Research: Distributed ORCD

• A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i(k) < \tilde{V}_t^i(j) \iff k > _{\pi_i}^t j \]
Ongoing Research: Distributed ORCD

- A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i (k) < \tilde{V}_t^i (j) \iff k >_{\pi_i}^t j \]
- Distributed computation via message passing
Ongoing Research: Distributed ORCD

• A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i(k) < \tilde{V}_t^i(j) \iff k > \pi_i^t j \]

• Distributed computation via message passing
  • Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i(j) \), by all \( j \in N(i) \)
Ongoing Research: Distributed ORCD

• A node only requires ordering of its neighbors

\[ \tilde{V}_t^i (k) < \tilde{V}_t^i (j) \iff k > \pi_t^i j \]

• Distributed computation via message passing
  
  • Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i (j) \), by all \( j \in \mathbb{N} (i) \)
  
  • Node \( i \) computes, infinitely often, its estimated index

\[
\tilde{V}_{t+1}^i (i) = q_t (i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} \tilde{V}_t^i (j)
\]
Ongoing Research: Distributed ORCD

- A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i (k) < \tilde{V}_t^i (j) \iff k > _{\pi_i}^t j \]
- Distributed computation via message passing
  - Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i (j) \), by all \( j \in \mathcal{N} (i) \)
  - Node \( i \) computes, infinitely often, its estimated index
    \[ \tilde{V}_{t+1}^i (i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} \tilde{V}_t^i (j) \]
Ongoing Research: Distributed ORCD

• A node only requires ordering of its neighbors

\[ \tilde{V}_t^i (k) < \tilde{V}_t^i (j) \iff k > _{\pi_t} j \]

• Distributed computation via message passing

  • Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i (j) \), by all \( j \in \mathbb{N} \ (i) \)

  • Node \( i \) computes, infinitely often, its estimated index

\[
\tilde{V}_{t+1}^i (i) = q_t (i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_{j \in S} \tilde{V}_t^i (j)
\]
Ongoing Research: Distributed ORCD

• A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i (k) < \tilde{V}_t^i (j) \iff k > \pi_i^t j \]

• Distributed computation via message passing
  - Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i (j) \), by all \( j \in \mathcal{N} (i) \)
  - Node \( i \) computes, infinitely often, its estimated index
    \[ \tilde{V}_{t+1}^i (i) = q_t (i) + \sum_{S \subset \Omega} P(S \mid i) \min_{j \in S} \tilde{V}_t^i (j) \]
  - If \( q_t (i) \) time invariant, \( V_t (\cdot) \) converges [LT’06]
Ongoing Research: Distributed ORCD

- A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i(k) < \tilde{V}_t^i(j) \iff k > \pi_i j \]
- Distributed computation via message passing
  - Node \( i \) is updated, infinitely often, \( \tilde{V}_t^i(j) \), by all \( j \in N(i) \)
  - Node \( i \) computes, infinitely often, its estimated index
    \[
    \tilde{V}_{t+1}^i(i) = q_t(i) + \sum_{S \subset \Omega} P(S \mid i) \min\tilde{V}_t^i(j)
    \]
    - If \( q_t(i) \) time invariant, \( V_t(\cdot) \) converges [LT’06]
- Simulations show similar performance to ORCD
Ongoing Research: Distributed ORCD

- A node only requires ordering of its neighbors
  \[ \tilde{V}_t^i(k) < \tilde{V}_t^i(j) \iff k > \pi_t^i j \]
- Distributed computation via message passing
  - Node \(i\) is updated, infinitely often, \(\tilde{V}_t^i(j)\), by all \(j \in \mathbb{N}(i)\)
  - Node \(i\) computes, infinitely often, its estimated index
    \[ \tilde{V}_{t+1}^i(i) = q_t(i) + \sum_{S \subseteq \Omega} P(S \mid i) \min_j \tilde{V}_t^i(j) \]
    - If \(q_t(i)\) time invariant, \(V_t(\cdot)\) converges [LT’06]
- Simulations show similar performance to ORCD
- Loopy message passing trees: Proof remains open!
ORCD: Ongoing and Future Work
ORCD: Ongoing and Future Work

• Throughput optimality of D-ORCD?
  • Loopy computation tree and outdated information
ORCD: Ongoing and Future Work

• Throughput optimality of D-ORCD?
  • Loopy computation tree and outdated information

• Delay Optimality
  • Path-based delay optimal routing [Gallager’77]
  • Heavy traffic regime (snapshot principle)
  • Approximate value function