



Exploit Broadcast Advantage in Wireless Networks

Tao Cui

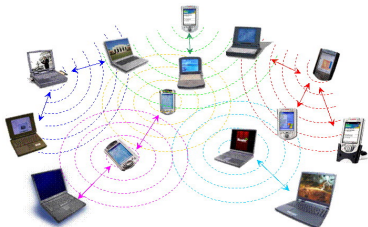
Department of Electrical Engineering
California Institute of Technology

The Lee Center for Advanced Networking Workshop
Pasadena, CA, May 2007

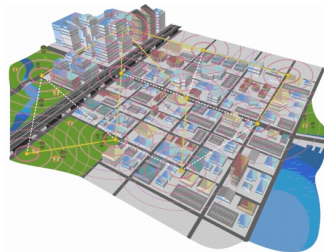
Joint work with Lijun Chen and Tracey Ho

- 1 Wireless Networks, Multicast, and Network Coding
- 2 Cross-Layer Design with Broadcast Advantage
- 3 Distributed Hypergraph Matching
- 4 Experimental Results

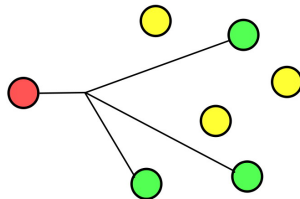
- Wireless networks have significantly impacted the world.
- Can be classified as
 - Cellular Networks
 - Wireless Sensor Networks
 - Wireless Ad-hoc Networks
- Play important roles in
 - Military communication,
 - Commercial communication,
 - Education



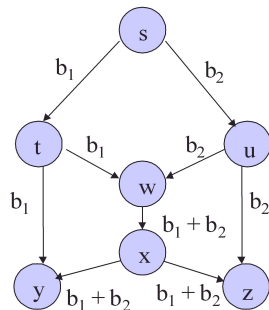
- Challenging problems:
 - Broadcast interference,
 - Distributed control ...
- Broadcast advantage:
 - Every transmission by a node can be received by all nodes that lie within its communication range.
 - Possible power saving and throughput improvement especially with broadcasting and multicasting.
- Question: *How to exploit the broadcast advantage in a distributed fashion?*



- Multicast delivers information to a group of receivers simultaneously.
- It uses each link of the network only once, and creates copies only when the links to the receivers split.
- Useful in applications such as teleconferencing and audio/video broadcasting.



- Conventional packet networks: each node's functions are limited to the forwarding or replication of received packets.
- Network coding:
 - Each node is allowed to perform algebraic operations on received packets.
 - Necessary to achieve multicast capacity in some networks.
 - Complexity benefits.



1 Wireless Networks, Multicast, and Network Coding

2 Cross-Layer Design with Broadcast Advantage

3 Distributed Hypergraph Matching

4 Experimental Results

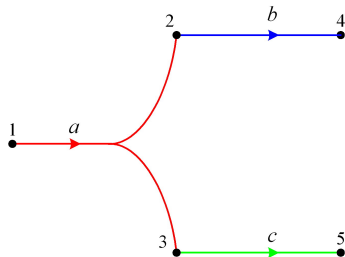
Directed Hypergraph

Hypergraph:

- The network is modelled as a **directed hypergraph** $\mathcal{H} = (\mathcal{N}, \mathcal{A})$.
- A hyperarc is a pair (i, J) , e.g., $a = (1, \{2, 3\})$.
- Each hyperarc (i, J) represents a broadcast link from node i to nodes in J .

Variables:

- g_{ij}^{mst} : virtual flow from source s to sink t over (i, J) and is intended to node $j \in J$, e.g., g_{1a2}^{14} .
- f_{ij}^m : physical flow of session m on (i, J) .
- r_{ij} : achievable rate on (i, J) .



Network resource allocation problem:

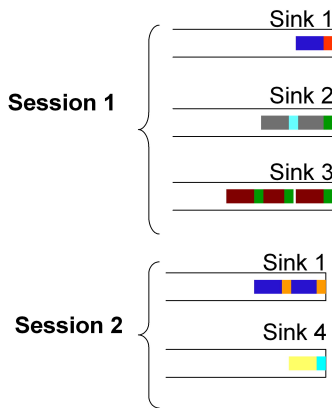
- Maximize $\sum_{m,s} U_{ms}(\mathbf{x}^{ms})$
- Subject to:
 - Flow conservation of g_{ij}^{mst} ,
 - Network coding: $\sum_{s,j} g_{ij}^{mst} \leq f_{ij}^m$,
 - Rate constraint: $\sum_m f_{ij}^m \leq r_{ij}$, $(r_{ij}) \in \text{Co}(\underline{r}(\underline{P}, \underline{S}))$.

Solution:

- Dual decomposition.
- Introduce dual variable q_i^{mst} at each node, which can be interpreted as queue length.

Each source adjusts its sending rate according to the aggregate queue lengths

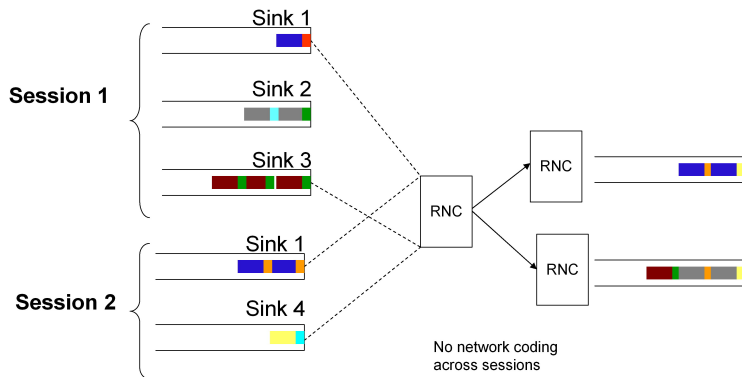
$$x^{ms}(\tau + 1) = U_{ms}^{\prime-1} \left(\sum_t q_s^{mst}(\tau) \right)$$



Session Scheduling and Network Coding



- For each (i, J) , the session with the maximum aggregate queue difference is chosen, i.e., $\hat{m} = \arg \max_m \sum_t \max_{s, j \in J} [q_i^{mst} - q_j^{mst}]^+$.
- A random linear combination of packets from sources in \hat{m} is broadcast to all nodes in J at the rate of r_{iJ} .



Link Scheduling:

- Define $w_{iJ} = \max_m \sum_t \max_{s,j \in J} [q_i^{mst} - q_j^{mst}]^+$.
- Link scheduling problem:

$$\max_{r,P} \sum_{(i,J) \in \mathcal{A}} w_{iJ} r_{iJ}, \text{ s.t. } (r_{iJ}) \in \text{Co}(\underline{r}(\underline{P}, \underline{S}))$$

Queue Length Update:

- At the end of each time slot, each node passes its queue length to all its neighbors for next time slot rate control, scheduling and network coding.



1 Wireless Networks, Multicast, and Network Coding

2 Cross-Layer Design with Broadcast Advantage

3 Distributed Hypergraph Matching

4 Experimental Results

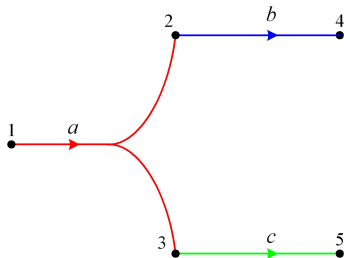
Hypergraph Matching Under the Primary Interference Model

- Primary interference model:
 - Each node is equipped with only a single transceiver. Hyperarcs that share a common node cannot be active simultaneously. CDMA or FDMA is used.

Hypergraph matching

A set of hyperarcs with no pair incident to the same node.

- Link scheduling becomes finding a maximum weighted hypergraph matching in \mathcal{H} with weight $w_{ij}r_{ij}$.



- Hypergraph matching is NP-complete [Lovasz86].
- Global greedy algorithm:
 - Adds a globally maximum weight edge into the matching.
 - Hard to be decentralized.
- Local greedy algorithm:
 - Adds a locally heaviest hyperedge into the hypergraph matching at each step.

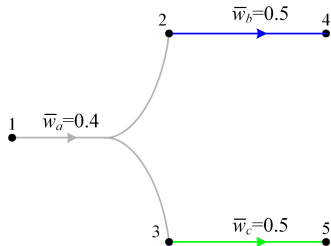
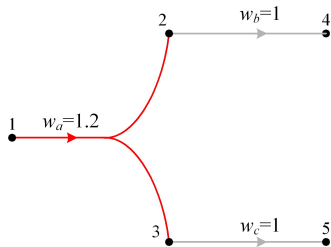
Locally heaviest hyperedge

A hyperedge is locally heaviest if its weight is at least as large as the weight of all adjacent hyperedges.

- Linear-time complexity and fully decentralized.
- Achieve an approximation ratio $\max\{\frac{1}{K}, \frac{1}{\kappa}\}$, where K is the maximum cardinality of the hyperedges and $\kappa = \max_{m \in \mathcal{M}} |\mathcal{T}_m| + 1$.

Greedy Algorithms (Cont.)

- Local greedy algorithm sometimes performs not well.
 - Some matched nodes may not contribute much to this locally heaviest hyperedge.
 - When these nodes are matched in other hyperedges, they may contribute more.
- Improved greedy algorithm
 - Use the average hyperedge weight, i.e., $\bar{w}_e = w_e/|e|$.
 - Achieve the same approximation ratio but perform better in practice.



- Randomized algorithm to find a maximal hypergraph matching.

Algorithm

- 1 Each unmatched node i attempts to transmit with probability p_i .
 - 2 If i attempts to transmit, it sends a matching request to each neighbor with probability $1/2$.
 - 3 If node i does not transmit and it receives several matching requests from its neighbors, it chooses one of them uniformly at random and sends a matched message.
 - 4 For all the nodes that decide to transmit, the nodes that reply with a matched messages are added into the hypergraph matching.
 - 5 Continue until no hyperedge can be found.
- The expected running time is $O(\log |\mathcal{E}|)$.
 - Stabilizes the system for any rate vector from $\frac{1}{K}\Lambda$.

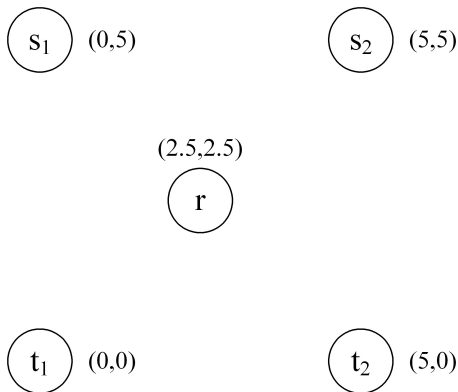


1 Wireless Networks, Multicast, and Network Coding

2 Cross-Layer Design with Broadcast Advantage

3 Distributed Hypergraph Matching

4 Experimental Results



- Path-loss exponent: 1.
- All nodes have unity signal power and identical noise power 0.1.

Rate Evolution

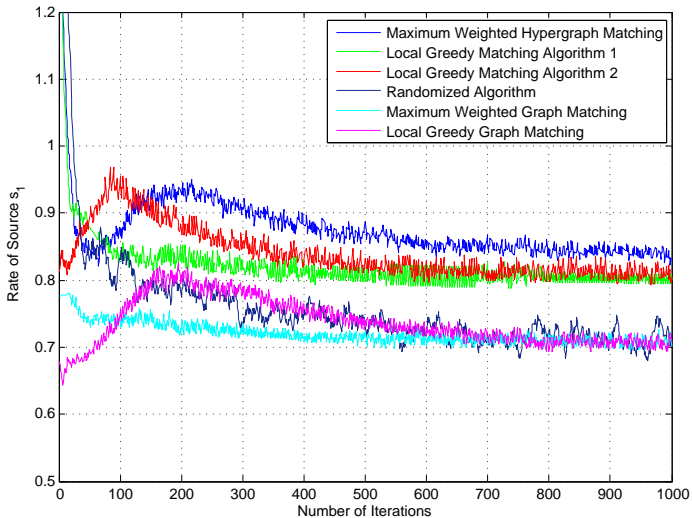


Table: Comparison of Different Algorithms in the Wireless Butterfly Network.

	HM_{opt}	HM_{grd1}	HM_{grd2}	HM_{rand}	M_{opt}	M_{lgd}
Average rate (bits/s)	0.8424	0.8030	0.8114	0.7145	0.7061	0.7054
Rate gain over M_{opt}	19.30%	13.72%	14.91%	1.19%	0%	-1%
Expected $w/w_{HM_{opt}}$	1	0.9759	0.9763	0.8385	0.7364	0.7290
Expected time-slots	-	4	3.9980	4.6400	-	5

Throughput gain increases as the number of sinks increases.

Table: Comparison of Different Algorithms in the Wireless Butterfly Network.

	HM_{opt}	HM_{grd1}	HM_{grd2}	HM_{rand}	M_{opt}	M_{lgd}
Average rate (bits/s)	0.8424	0.8030	0.8114	0.7145	0.7061	0.7054
Rate gain over M_{opt}	19.30%	13.72%	14.91%	1.19%	0%	-1%
Expected $w/w_{HM_{opt}}$	1	0.9759	0.9763	0.8385	0.7364	0.7290
Expected time-slots	-	4	3.9980	4.6400	-	5

Throughput gain increases as the number of sinks increases.

Table: Comparison of Different Algorithms in the Wireless Butterfly Network.

	HM_{opt}	HM_{grd1}	HM_{grd2}	HM_{rand}	M_{opt}	M_{lgd}
Average rate (bits/s)	0.8424	0.8030	0.8114	0.7145	0.7061	0.7054
Rate gain over M_{opt}	19.30%	13.72%	14.91%	1.19%	0%	-1%
Expected $w/w_{HM_{opt}}$	1	0.9759	0.9763	0.8385	0.7364	0.7290
Expected time-slots	-	4	3.9980	4.6400	-	5

Throughput gain increases as the number of sinks increases.

Table: Comparison of Different Algorithms in the Wireless Butterfly Network.

	HM_{opt}	HM_{grd1}	HM_{grd2}	HM_{rand}	M_{opt}	M_{lgd}
Average rate (bits/s)	0.8424	0.8030	0.8114	0.7145	0.7061	0.7054
Rate gain over M_{opt}	19.30%	13.72%	14.91%	1.19%	0%	-1%
Expected $w/w_{HM_{opt}}$	1	0.9759	0.9763	0.8385	0.7364	0.7290
Expected time-slots	-	4	3.9980	4.6400	-	5

Throughput gain increases as the number of sinks increases.

- Distributed optimization with broadcast advantage.
- Distributed low complexity hypergraph matching algorithms for link scheduling.
- Ongoing work:
 - Extension to data gathering in wireless sensor networks.
 - New hypergraph matching algorithms
 - Extension to other interference models.