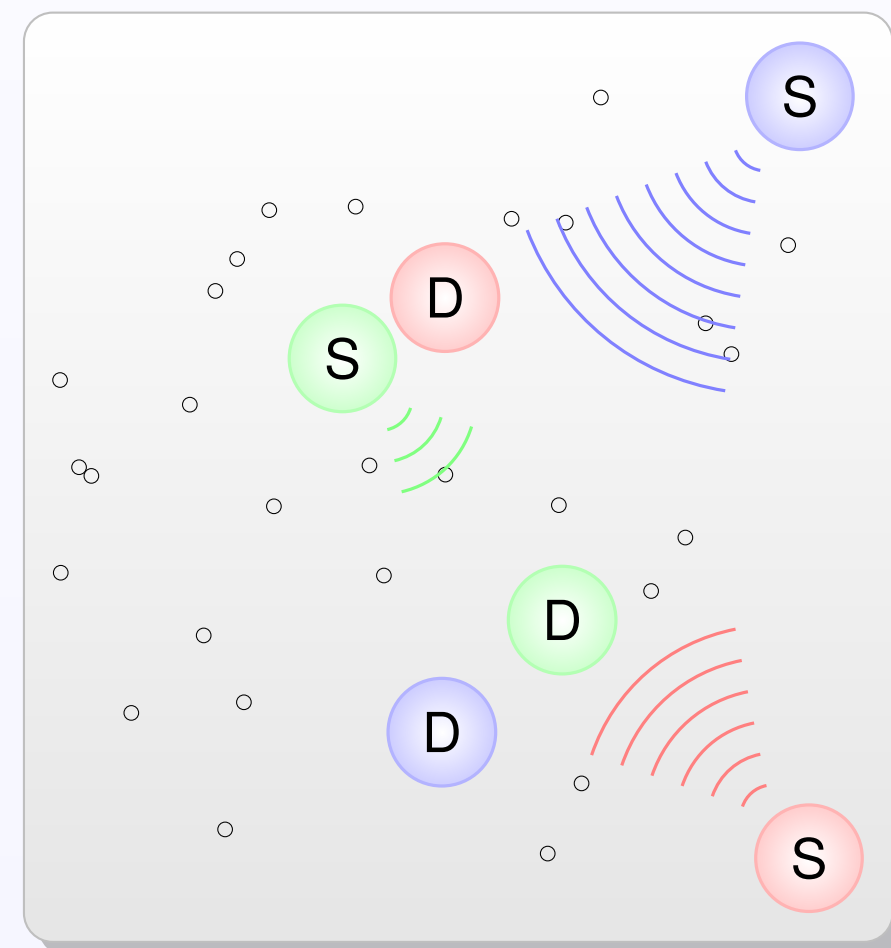


Motivation

- Today's systems are distributed: sources and destinations are no longer necessarily centralized: *mesh, ad-hoc*.
- There are usually other terminals sharing the resource which can relay or forward packets.
- Fine-grained resource management is required: schedule, power allocation, and route. *NP-Hard, Non-convex*



Is there a low-complexity scheme for optimally allocating spatial and temporal resources?

System Model

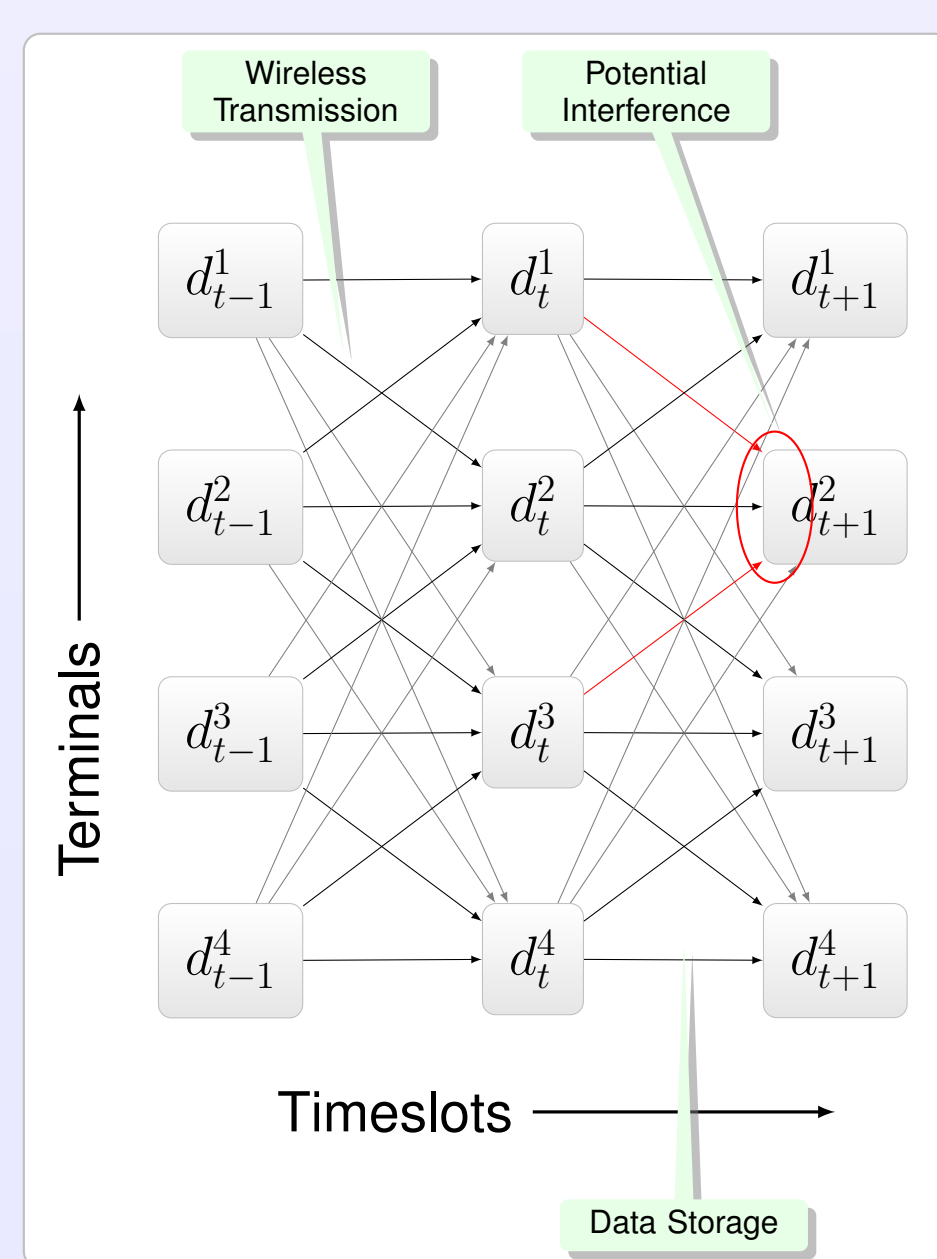
- All resources are shared, TDMA system
- Pathloss-dominant AWGN channel model
- N identical terminals are *all* half duplex and share a maximum power constraint
- **Goal:** For each timeslot, allocate resources for a set of flows to maximize the minimum throughput.

Decomposition and the NFIC

Approach: *treat each flow separately*. We must manage spatial and temporal interactions of the terminals and flows, while simultaneously addressing half-duplex, interference, and route continuity constraints.

Network-Flow Interaction Chart (NFIC)

- Well-defined data structure with values derived from network topology
- Nodes represent terminals from the network
- Edges represent possible transmission between terminals



Strengths of NFIC:

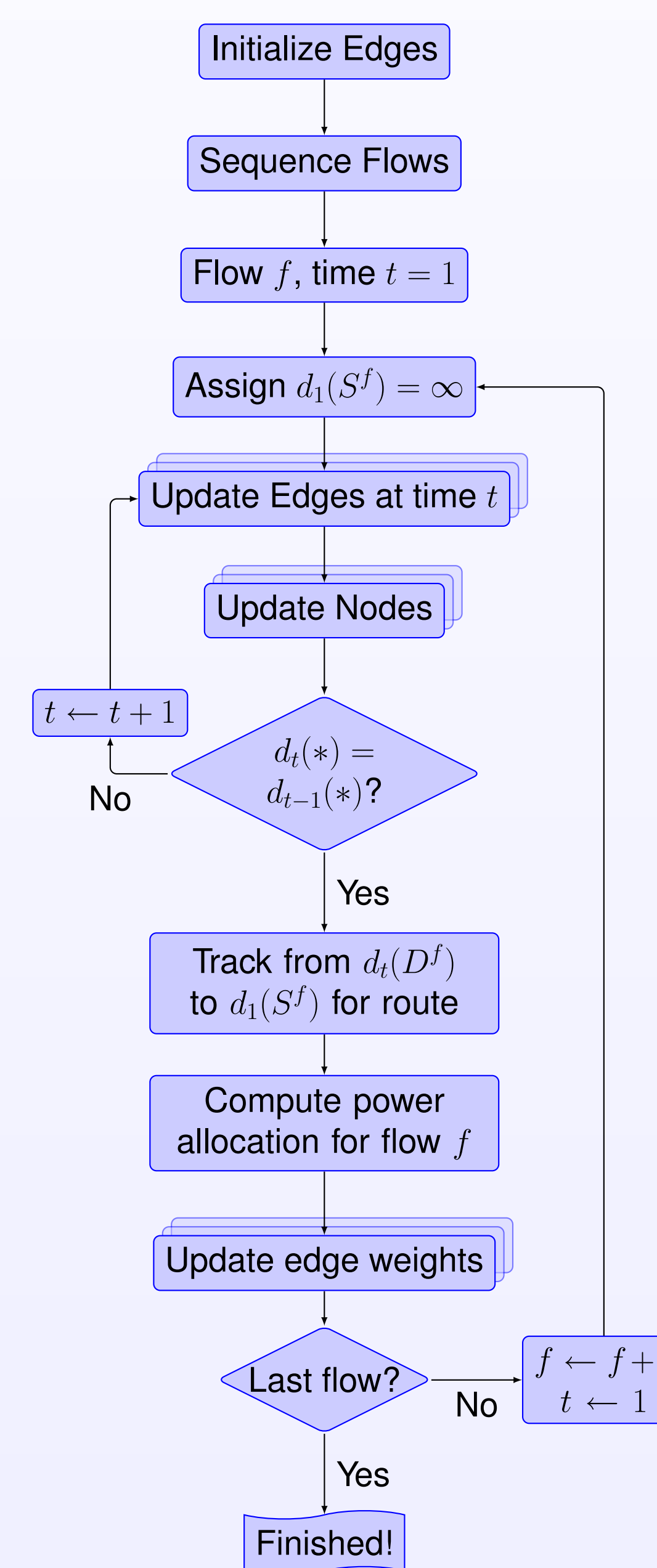
- Jointly capturing spatial and temporal interactions
- Simple rules manage interference, duplexing, etc.
- Generalizes to cooperative technology
- Enables use of fast *dynamic programming* techniques

Dynamic Programming:

- Reduces complexity from $O(N^N)$ to $O(N^3)$
- General approach, can optimize for throughput, delay, cost, etc.
- Only require monotone semiring $K = \{A, \oplus, \otimes, I_{\oplus}, I_{\otimes}\} \Rightarrow$ widespread applicability

Scheduling Algorithm

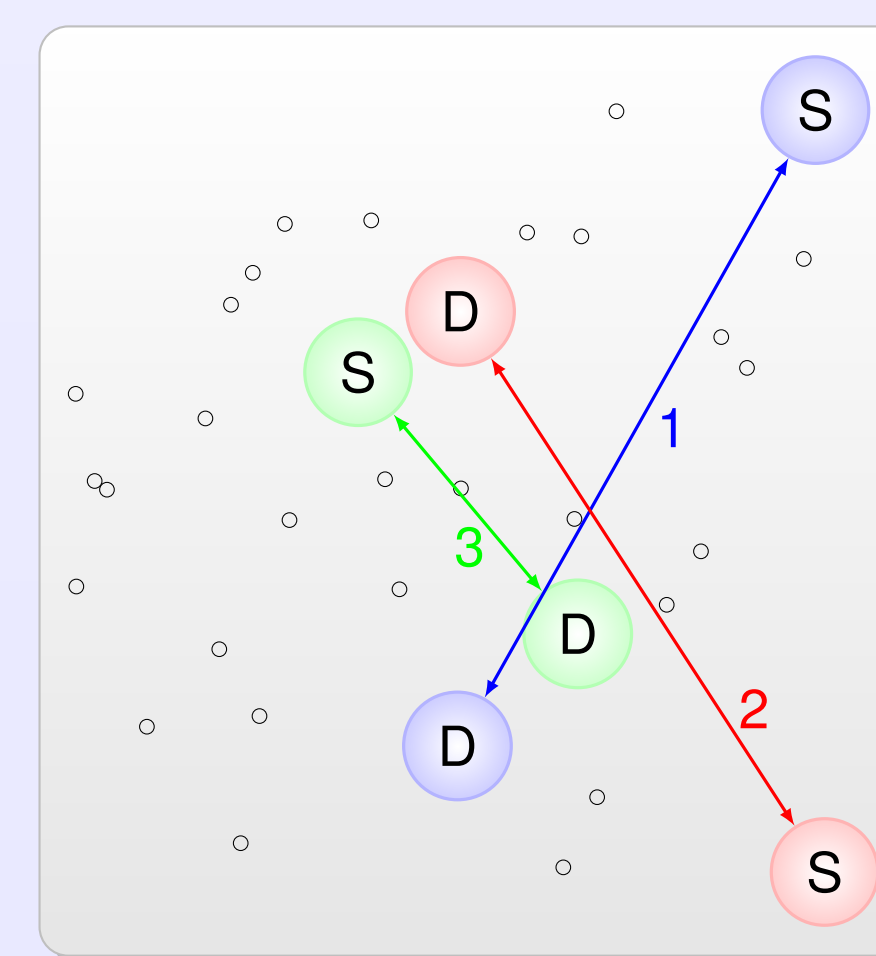
1. Edges are given weights corresponding to the capacity between terminals if transmission is isolation: $\log(1 + \text{SNR})$.
2. Flows are sequenced according to the resources they're estimated to consume (see below).
3. Only the source node of flow f has value at time $t = 1$, all others are set to zero.
4. Edges are updated with the minimum of either their destination node value or their current value.
5. Nodes are updated with the maximum of all incoming edges.
6. If no node weights have changed since the last time step, we have discovered all routes for the packets.
7. The destination is now weighted with highest throughput, follow the path from $d_t(D^f)$ back to $d_t(S^f)$ to find the route and corresponding schedule.
8. Power for each leg of the route is chosen such that all constituent links have the same rate.
9. Subtract resources utilized by flow f : Remove edges to nodes in use, limit transmission rates so as not to cause undue interference all previously routed flows.
10. Address the next flow. If this is the last flow, the schedule has been computed and we are finished.



Flow Sequencing

Scheme: Sort flows by estimated resource use: how many timeslots we expect them to require based on pathloss and distance.

- **Longest first:** The longest flow is a good estimate of the overall schedule length in terms of hops. After it has been allocated, shorter flows can be scheduled and routed opportunistically.
- **Shortest first:** Allocate flows which require fewest resources first, then schedule long flows with remaining resources.
- **Random order**

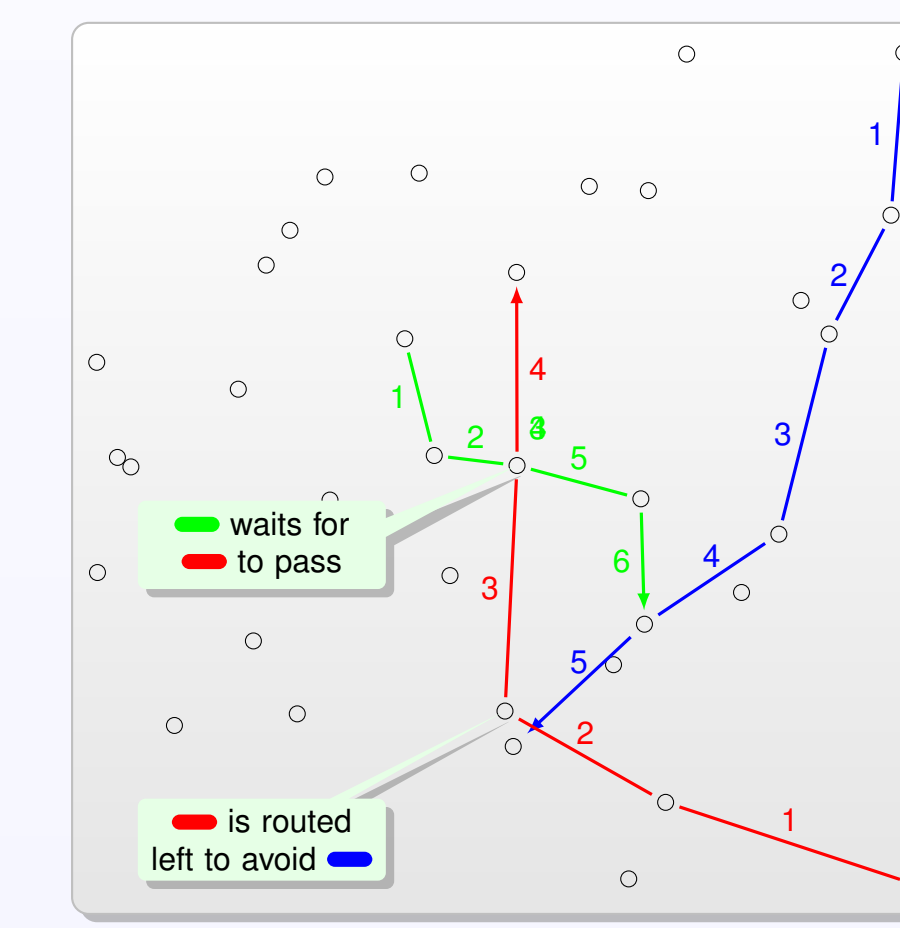


Sorting flows from longest to shortest yields best performance

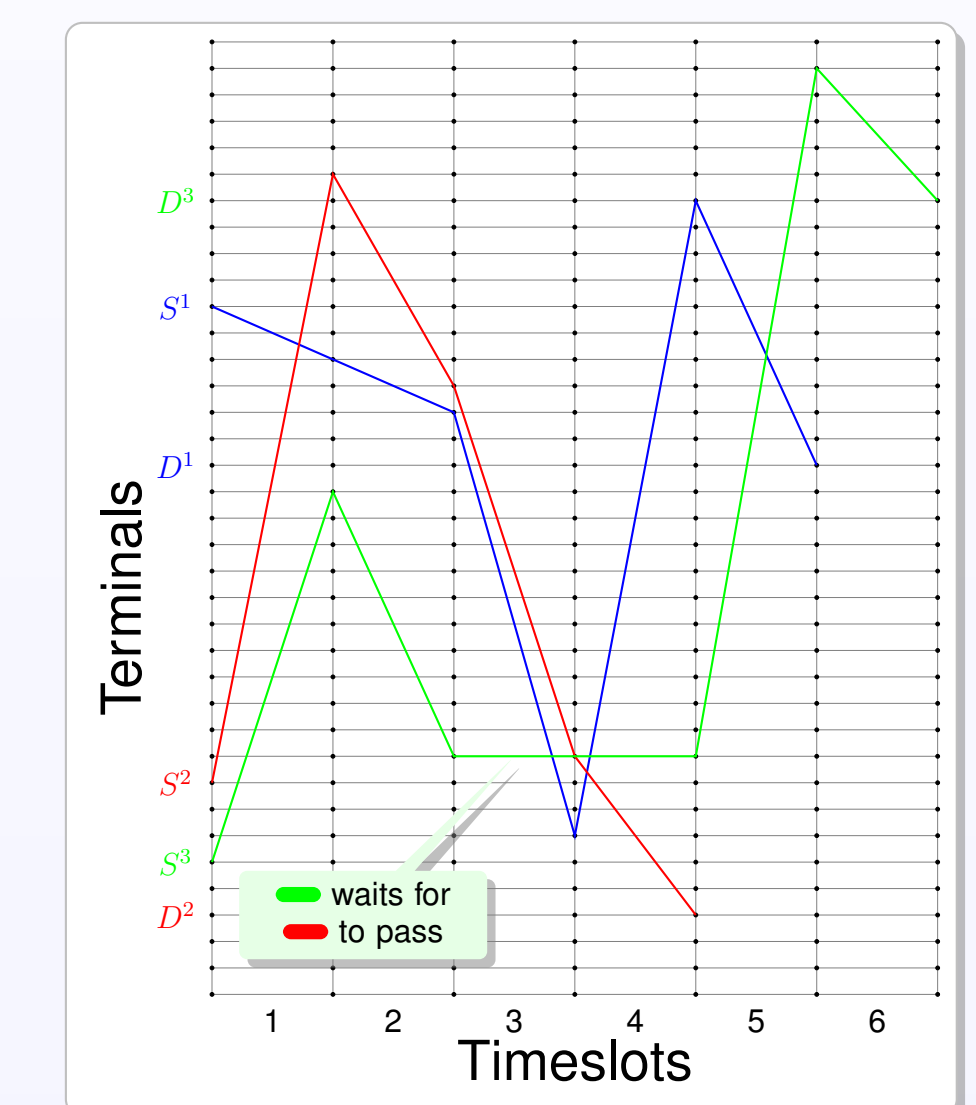
Complexity

Decomposing by flows and exploiting dynamic programming allows our algorithm to run in $O(N^3)$, exactly on par with the best-known routing algorithms.

Example: Random Network



Flows are processed in order 1, 2, 3, which corresponds to the longest flow first.

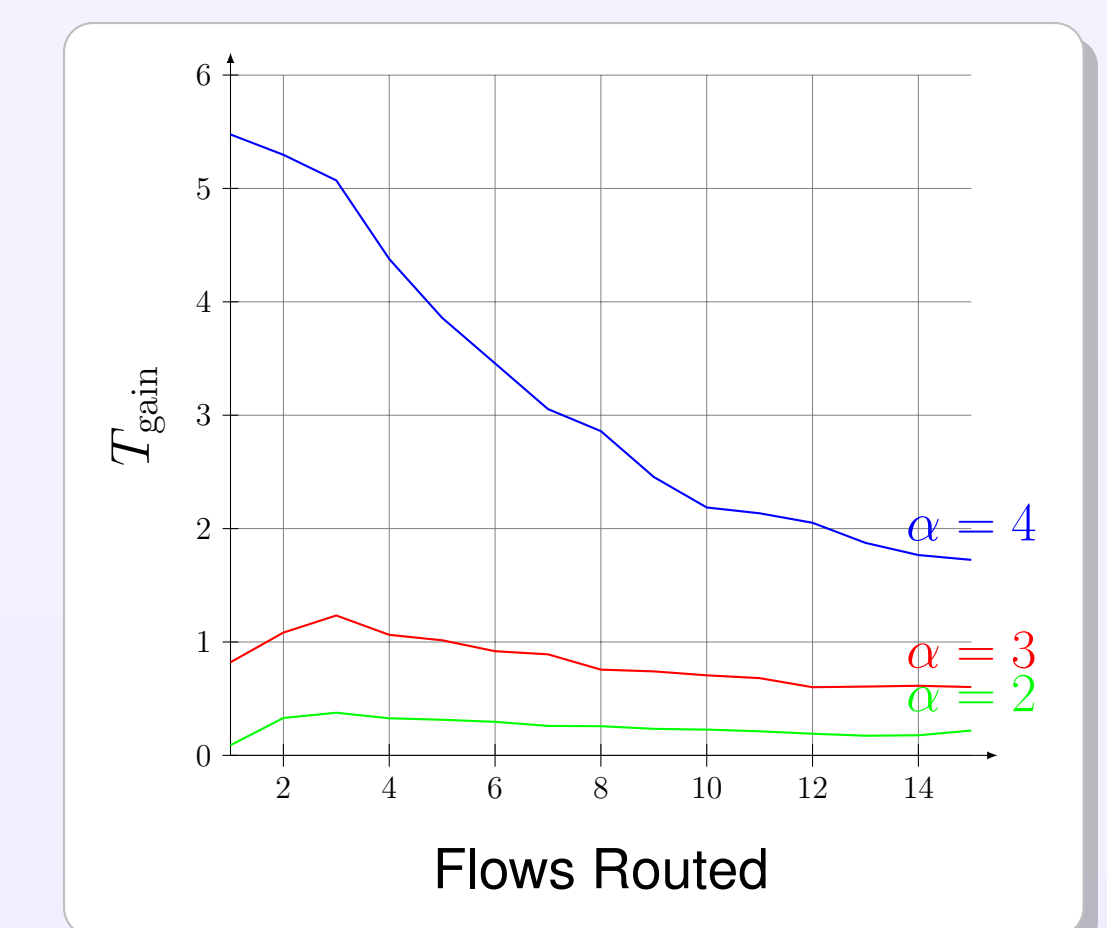


Random network with three flows, NFIC schedule is shown at right. Flow 3 (green) cannot transmit in timeslots 3 and 4 because Flow 2 (red) is using the node; its data is stored there instead. Flow 2 (red) is routed to the left of the straight-line path in order to be active at the same time as Flow 1 (blue).

Throughput Increases

Mean normalized throughput increase using NFIC allocation over round-robin one-hop allocation, constant power.

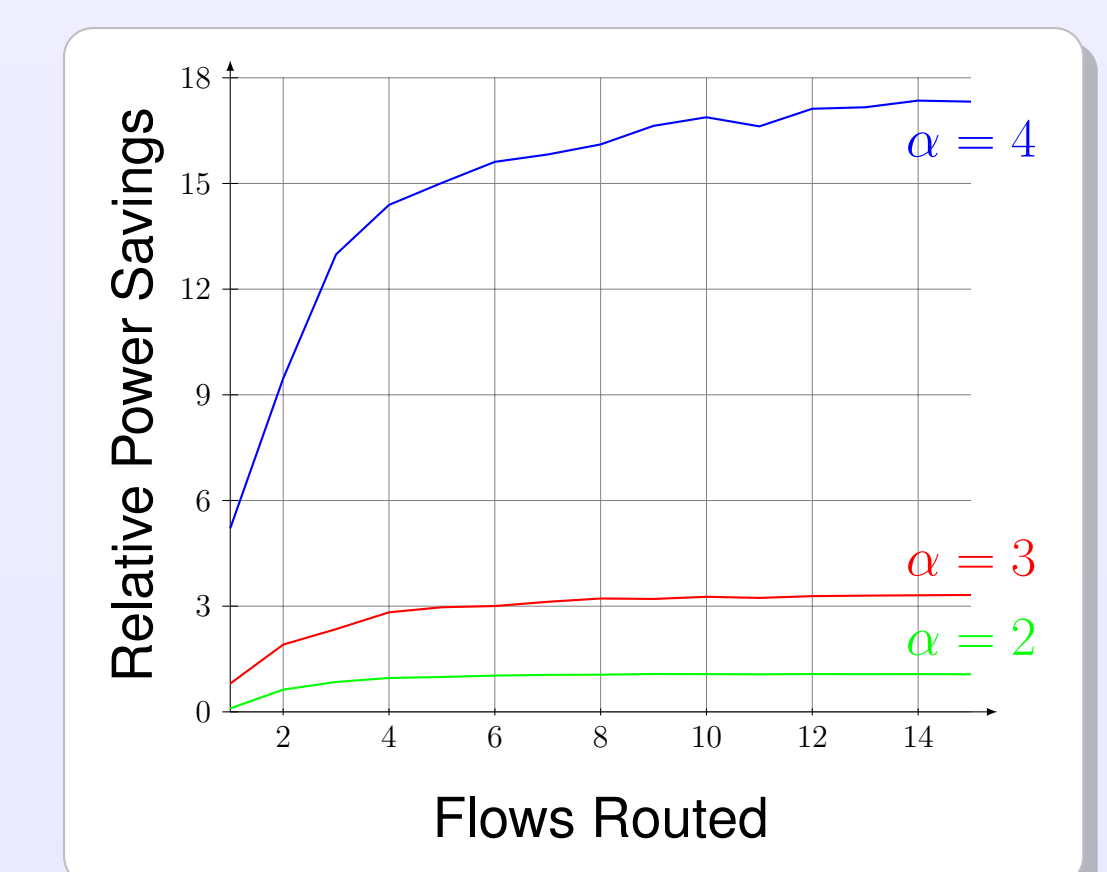
- For all reasonable pathloss, NFIC allocation improves throughput.
- Gains are higher for large α since the network is more "isolated," allowing resources to be allocated simultaneously: spatial opportunism.
- Gains diminish with many flows as a result of congestion.



Power Savings

For the same throughput as NFIC allocation, how much more power the RR scheme requires normalized by NFIC power.

- At low pathloss, NFIC allocation requires about half as much power as RR allocation.
- NFIC allocation is better at high pathloss, requiring only 10% as much power as RR for the same throughput.
- NFIC is able to exploit the multihop advantage as well as opportunistic spatial reuse.



Conclusions & Future Work

Results:

- Decomposition for efficiency: $O(N^3)$ operations
- General NFIC framework
- Strong improvements in throughput & power

Future Work:

- Bound optimality gap
- Derive better algorithms
- Study streaming packet cases