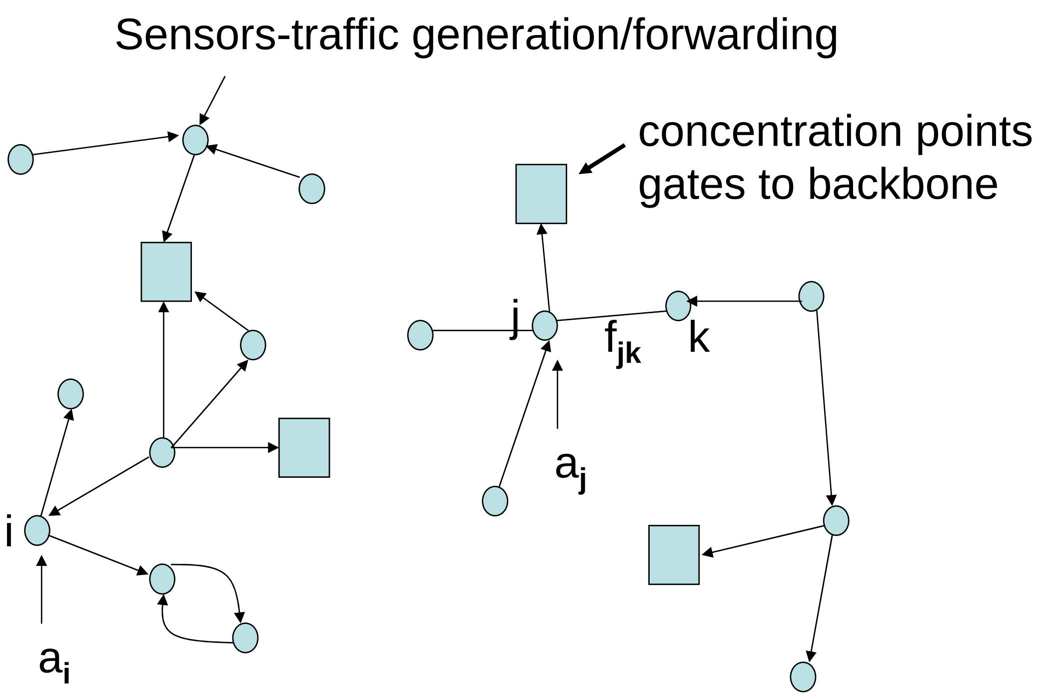
Coordination and resilience in ad hoc and sensor networks

Leandros Tassiulas
Univ. of Thessaly
Volos/GREECE
www.inf.uth.gr/~leandros

Parts of the presentation joint work with
Saswati Sarkar, Stavros Toumpis,
Leo Georgiadis, Jordan Koutsopoulos

Information Collection Network



Attributes of the network

- Minimal control over sensor node placement
- Deployment scenario: sensors randomly dispersed through a mortar to the (possibly hostile) terrain to be monitored
- Absence of central controller
- A node unaware of the topology besides knowledge of the identities of the other nodes within his range (one-hop away neighbors)
- The traffic load distribution unknown and unpredictable
- Sensor nodes die or destroyed unpredictably

Challenges

- Identify algorithms that accomplish the traffic forwarding task
- Deal with:

  -unpredictable traffic,

  -unknown and unpredictably changing topology

  -lack of central control
- Quantify the “goodness” of various algorithms in the current context

Traffic forwarding in a traditional setting

- Identify the end-to-end traffic load matrix
- Characterize the topology
- Obtain a traffic flow that supports the end-to-end traffic requirements and optimizes average delay
- Design the routing matrices at the nodes to realize the above computed optimal flow

In our setting nothing of the above applies!

Traffic considerations - dynamic operation

$a_i(t)$: amount of traffic generated at node i in [0,t] (arrivals)

$a_{ik}(t)$: amount of traffic transmitted from node i to k in [0,t]

$x_i(t)$: traffic accumulated in i at t

Flow conservation at node i, at t

$$\sum_{k=1}^{N} a_{ki}(t) = x_i(0) + x_i(t) + \sum_{k=1}^{N} a_{ik}(t)$$

Existence of steady state: stability

- Stochastic traffic: $\sup_{\{t>0\}} E[X_i(t)] < \infty$
- Deterministic traffic: $\sup_{\{t>0\}} X_i(t) < \infty$

Necessary and sufficient condition for stability

Assuming arrivals and cross traffic have long term avg.

$$\lim_{t\to\infty} \mathrm{a_i}(t)/\mathrm{t} = a_\mathrm{i}, \ \lim_{t\to\infty} \mathrm{a_{ik}}(t)/\mathrm{t} = f_\mathrm{ik},$$

Flow conservation at each node i

$$\sum_{k=1}^{N} f_{ki} + a_i = \sum_{j=1}^{N} f_{ij}$$

Link capacity condition $f_{ij} \le C_{ij}$

•Spatial traffic load vector $a = (a_1, \ldots, a_N)$

$$F_a = \{ f : f \text{ feasible network flow for } a \}$$

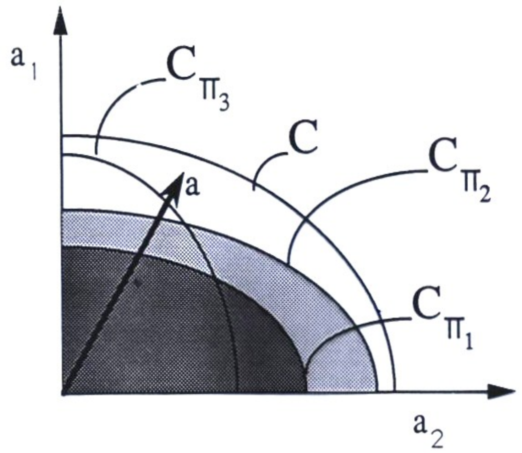
• A traffic load vector is feasible if there is at least one corresponding feasible network flow

• C: end-to-end throughput capacity region, includes all feasible traffic load vectors

Throughput Consideration

- Definition: Capacity region Cπ of a policy π: the set of arrival rate vectors <u>a</u> for which the system is stable under π
- Definition: Capacity Region C of the system:

$$C = \bigcup_{\pi \text{ activation policy}} C_\pi$$

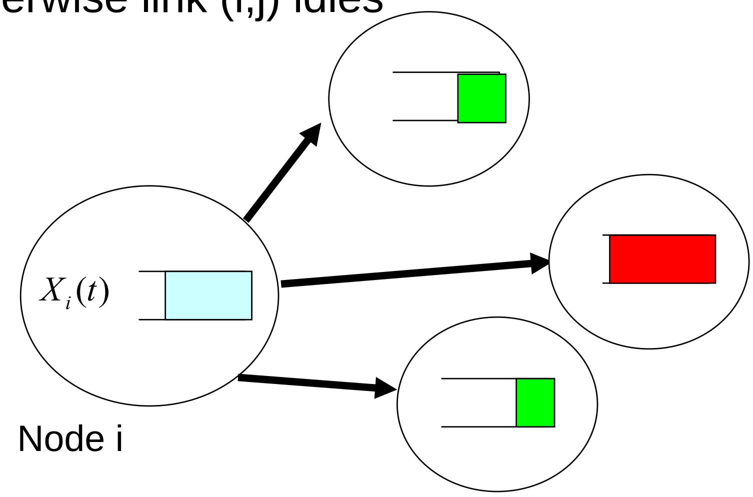


Adaptive Back Pressure routing and flow control

Each node i, <u>asynchronously with respect to other nodes</u>, observes the backlog X (in number of packets) of its outgoing neighbor, j, and

if $X_i > X_j$ sends a packet to j

otherwise link (i,j) idles



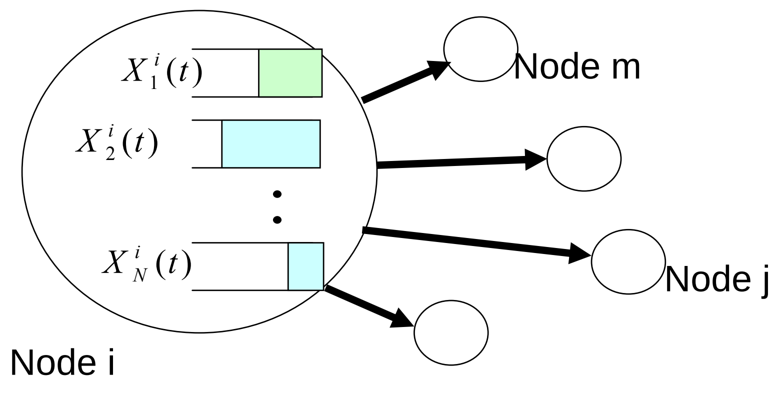
The adaptive backpressure flow control achieves maximum traffic forwarding throughput i.e. has capacity region equal to the system capacity region

Multiclass traffic forwarding

A packet in transit is characterized by its destination alone

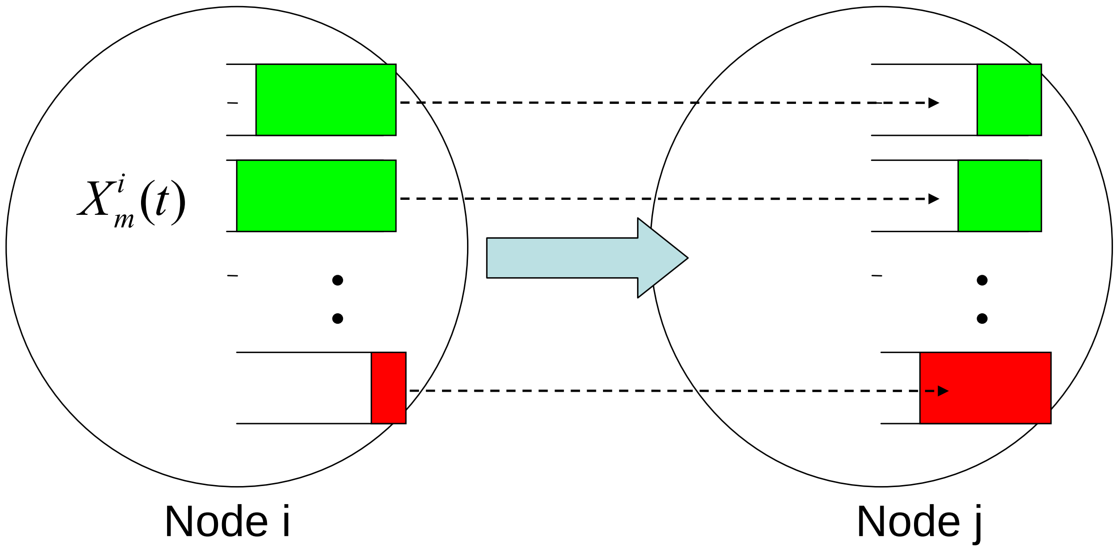
At each node packets of N traffic classes, one for each destination

One packet may be forwarded through each link



Back pressure flow control

If $X_m^i(t) - X_m^j(t)$ is negative then
class m is no eligible for transmission from i to j

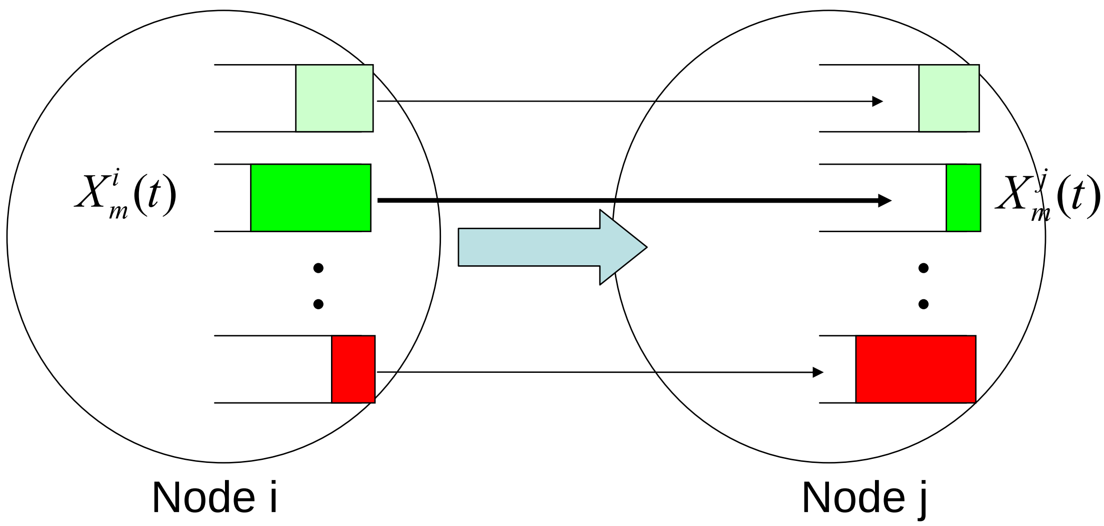


Class priority scheduling

Transmit a packet of class m for which

$$X_m^i(t) - X_m^j(t)$$

is maximum among all eligible classes



The combination of backpressure flow control with class priority scheduling achieves maximum traffic forwarding throughput in the general multiclass network

Operation in the overload region-fluid model

- If the traffic load vector $\mathbf{a}$ is out of the capacity region then there is no feasible flow, i.e., $\mathbf{F}_{\mathbf{a}} = \emptyset$
- Backlog accumulation at certain nodes inevitable
- Distribution of the backlog depends on the routing policy, i.e. f
- What are preferable backlog distributions and how do we achieve them?
- Generalize the notion of a flow to **superflow** in order to study the behavior of the system in over load

Flows, Superflows and overflows

- **Superflow f** : a flow vector that satisfies
  - ✓link capacity condition $f_{ij} \le C_{ij}$
  - ✓relaxed flow conservation condition $\sum_k f_{ki} + a_i - \sum_k f_{ik} = q_i \ge 0$
- **Overflow:** Backlog build-up rate vector $\mathbf{q} = (q_1, q_2, \ldots, q_N)$
- $\hat{\mathbf{F}}_{\mathbf{a}}$: set of superflows associated with traffic load vector a

  Note: $\mathbf{F}_{\mathbf{a}} \subseteq \hat{\mathbf{F}}_{\mathbf{a}}$
- $\mathbf{Q}_{\mathbf{a}}$: Set of feasible overflow vectors, i.e. realizable by some $\mathbf{f} \in \hat{\mathbf{F}}_{\mathbf{a}}$

  Note: if $\mathbf{F}_{\mathbf{a}} \neq \emptyset$ then $\mathbf{0} \in \mathbf{Q}_{\mathbf{a}}$

Throughput of a superflow

$\mathrm{T_f}$: Throughput of superflow f, the total amount of information reaching the sinks

by adding the flow-conservation equations at all nodes

$$\mathrm{T_f} = \sum_{i=1}^{N} a_i - \sum_{i=1}^{N} q_i$$

Efficient Overload Management

Drive the network to operating points where the overload vector is "good"

•The throughput is maximized

•The backlogged traffic is distributed in a *most balanced manner*

Partial ordering $\angle$ in R_+^n

Given vector $q=(q_1,q_2,\ldots,q_n)$ let $\hat{q}=(\hat{q}_1,\hat{q}_2,\ldots,\hat{q}_n)$ be the vector rearranged in *decreasing order*

$$q^1 \angle q^2$$

If

$$\sum_{i=1}^{m} \hat{q}_i^1 \le \sum_{i=1}^{m} \hat{q}_i^2 \text{ for all } m=1,..,N$$

We postulate q^1 *more balanced than* q^2

If for two superflows f^1, f^2 we have the corresponding overflow vectors

$$q^1 \angle q^2$$

then f^1 is better than f^2, we call it, *more balanced*

Why?

- Throughput under f^1 larger than under f^2
- q^1 lexicographically smaller than q^2

So what?

Buffer overflow will occur later for f^1 than for f^2

- For any convex function U() we have $\sum_{i=1}^{N} U(q_i^1) \leq \sum_{i=1}^{N} U(q_i^2)$

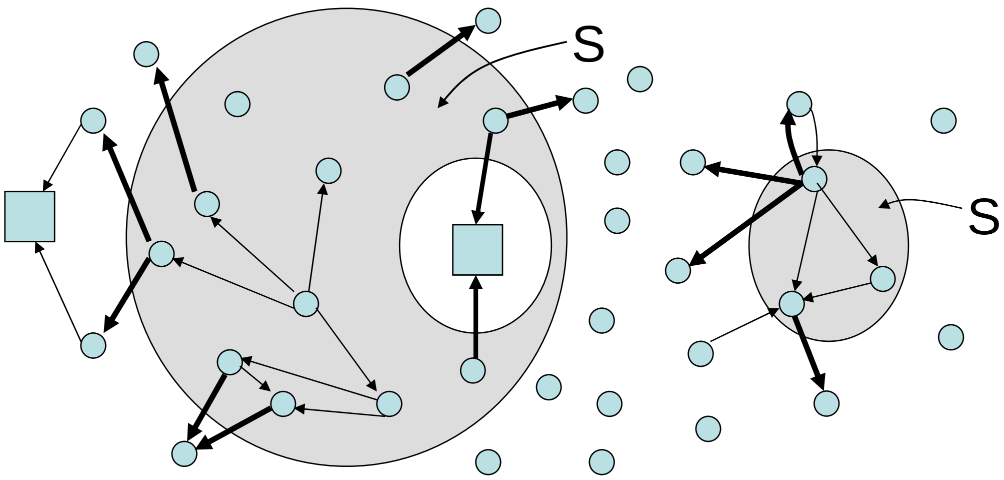
If there is an overflow q^0 that is more balanced than any other overflow, then it is optimal

Since $\angle$ partial ordering, an optimal overflow need not necessarily exist

We show

- There is a unique optimal overflow q^0, *most balanced* and we characterize it
- There is a class of superflows all of which achieve q^0
- A distributed iterative algorithm for computing most balanced superfllows
- A combinatorial $O(N^4)$ algorithm for finding most balanced superflows and associated q^0

Traffic load feasibility



- Cut (S, S^c): partion of nodes V in two sets S, and S^c = V - S
- L_{out} (S): set of links originating at some node of S and terminating either at some node of S^c or to a gateway node

Traffic load feasibility

- **Region** S: any subset of the nodes of the network
- **Local load of region** S: $a(s) = \sum_{i \in S} a_i$ *(all traffic generated in* S*)*
- **Cut capacity**: $C(S,S^c) = \sum_{e \in L_{out}(S,S^c)} C_e$
- <u>The local load should not exceed the cut capacity out of S</u>

$$a(S) \leq C(S,S^c) \text{ for all subsets } S$$

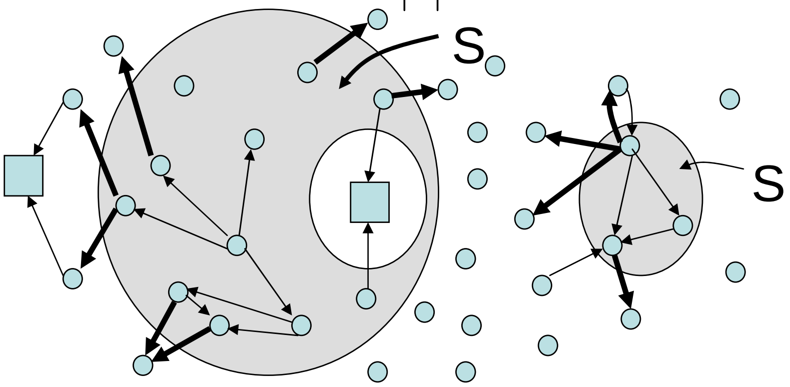
(simple consequence of max-flow min-cut theorem)

Bounds on overflow

- For any region S the aggregate overflow is bounded by

$$\sum_{i\in S} q_i \geq \left(a(S) - C(S,S^c)\right)^+$$

- Therefore $\max_{i\in S} q_i \geq \frac{\left(a(s) - C(S,S^c)\right)^+}{|S|} = W(S)$ *normalized overload of S*



✓ **Note:** structural property of the connectivity graph and the traffic distribution. *Holds for all routings **f***

Load Balancing Decomposition

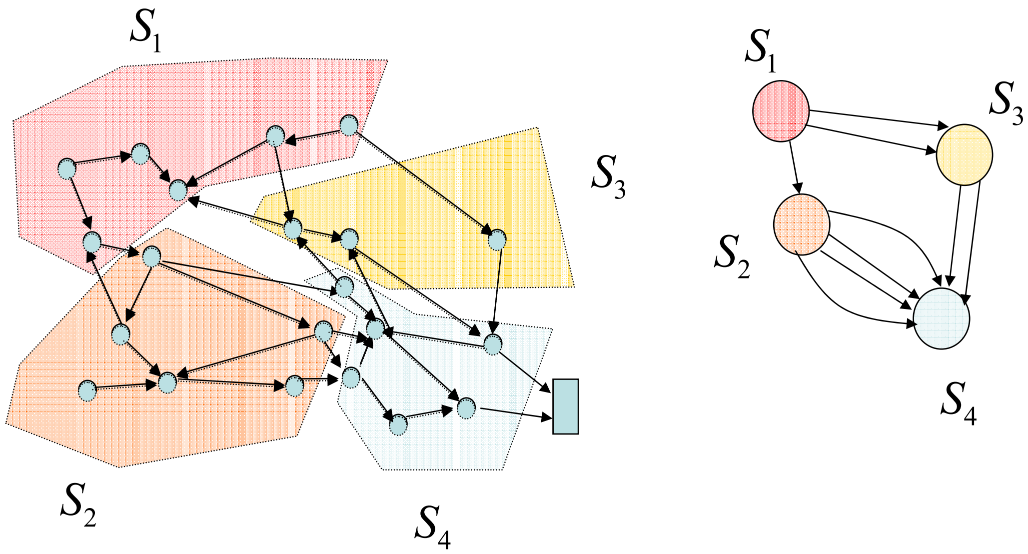
There is a unique partition of the connectivity graph in regions $S_1, S_2, \ldots, S_K$ derived by the iterative application of the following network reduction operation

- Let $R_1 = \max_{S\subseteq V}\{W(S)\}$, $S_1 = \bigcup_{S:W(S)=R_1} S$

Recall: V is the set of all nodes
Note: $W(S_1) = R_1$ (can be proved)

- Remove all nodes of S_1 and corresponding edges that are ending to a node of S_1
- At each node i in $V - S_1$, for which there is a link (i,j) originating at some node j in S_1, increase the exogeneous traffic load to $a_i + c_{ij}$
- From the reduced graph derive R_2 and S_2 similarly
- If the traffic load in the reduced graph is feasible the process terminates

Load Balancing Decomposition



SABP superflows achieve optimal overflow

- SABP is a superflow that satisfies the following

$$f_{ij} = \begin{cases} C_{ij} & \text{if } q_i > q_j \\ 0 & \text{if } q_i < q_j \\ \text{arbitrary} & \text{if } q_i = q_j \end{cases}$$

- Under an SABP the overflow vector is most balanced
- Any link from S_i to S_j is saturated if *i>j* and carries 0 traffic if *i<j*
- Links connecting nodes in the same region have arbitrary traffic

Note: (1) SABP's are not unique but they all have the same overload vector

(2) If the traffic load is feasible, every feasible flow in $F_{\underline{\mathbf{a}}}$ is trivially SABP

Computing SABP

- Start from arbitrary superflow $\mathbf{f}_1$
- Iteration

  Pick a link (i,j) randomly

  if $q_i > q_j$ and $f_{ij} < C_{ij}$ increase f_{ij} until either

  $f_{ij} = C_{ij}$ or $q_i = q_j$

  if $q_i < q_j$ and $f_{ij} > 0$ decrease f_{ij} until either

  $f_{ij} = C_{ij}$ or $q_i = q_j$

If the above iteration is applied repeatedly the corresponding sequence of overflows $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \ldots.$ converges to an SABP

Why?

SABP´s are the only fixed points of the above iteration

The lengths of the corresponding overflow vectors are nonincreasing, $|q^1| > |q^2| > \ldots.$,

Computing SABP

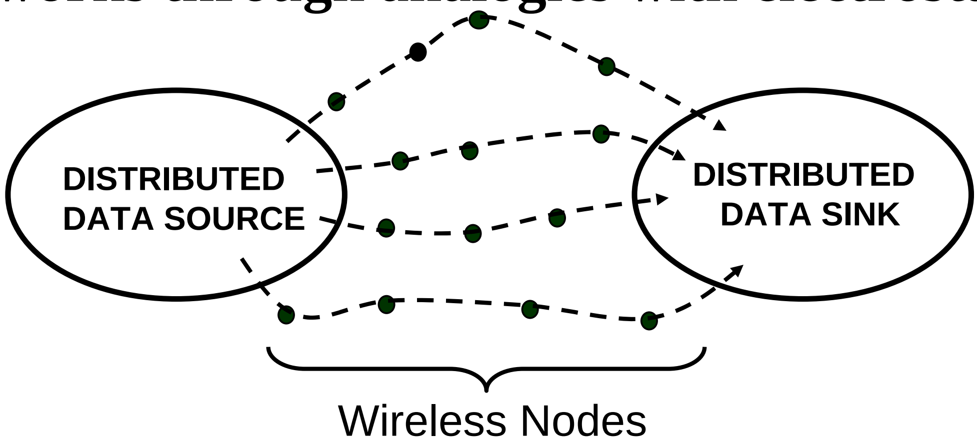
- The resulting algorithm converges asymptotically to SABP

- We have an $O(N^4)$ combinatorial algorithm for computing an SABP

- As a byproduct of the proof, it can be shown that SABPs are also optimal for:

$$\min_{\mathbf{q} \in \mathbf{Q}_\mathbf{a}} \sum_{i=1}^{N} q_i^{\gamma}, \quad \gamma \geq 1$$

Optimal node placement in large scale wireless networks through analogies with electrostatics



- What is the “best” placement for the wireless nodes?
  - The “best” placement minimizes the number of nodes needed to move a given volume of traffic
- What is the induced traffic flow?

- Large number of nodes in the network
  - on practical terms, on the order of 1000’s
  - many envisioned sensor networks are expected to have that size
- We take a ***macroscopic*** view of the network
  - We do not worry about transmissions to/from individual nodes
  - We only consider the flow of liquid information between various parts of the network
- We want to find the minimum number of nodes that can support a given level of traffic

Spatial fluid model

- **Information density function** r(x,y) (measured in bps per sq. m.):
  - If r(x,y)>0 (<0), information is created (absorbed) with rate |r(x,y)|dx at a small area of size dx, centered at (x,y)
- **Traffic flow function T**(x,y) (measured in bps per m):
  - At location (x,y), information flows towards the direction of **T**(x,y), and with intensity | **T**(x,y)|
- Conservation of flow:

$$\mathrm{div}(\mathbf{T})=r$$

Physical Layer

- **Node density function** d(x,y) (measured in nodes per sq. m.):
  - Number of nodes at a small area of size dx, centered at (x,y), is d(x,y)dx
- The more nodes are placed in an area, the more traffic can go through that area:
  - $|\mathbf{T}(x,y)| <= K\ d(x,y)^{1/2}$
  - This formula is the *fluid* equivalent of the Gupta/Kumar result
- The necessary condition of acyclic flow in the fluid domain

$$\mathrm{curl}(\mathbf{T})=0$$

“Packetostatics”

- The traffic flow **T**(x,y) and information density r(x,y) must satisfy:

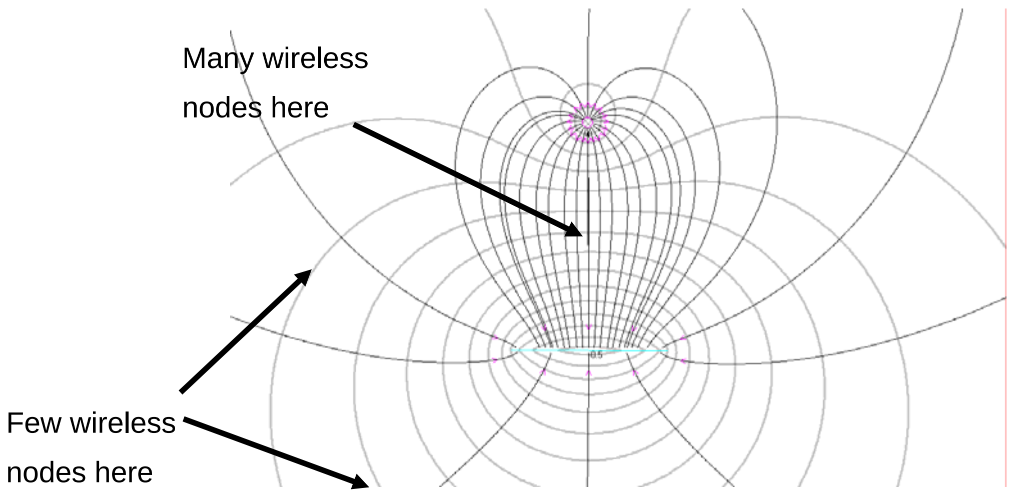
$$\mathrm{div}(\mathbf{T})=r,\ \mathrm{curl}(\mathbf{T})=0$$

- In Electrostatics, the electric field **E**(x,y) in a region of charge density q is described by:

$$\mathrm{div}(\mathbf{E})=q,\ \mathrm{curl}(\mathbf{E})=0$$

- So actually our problem is a 19^{th} century problem, and we can use standard techniques

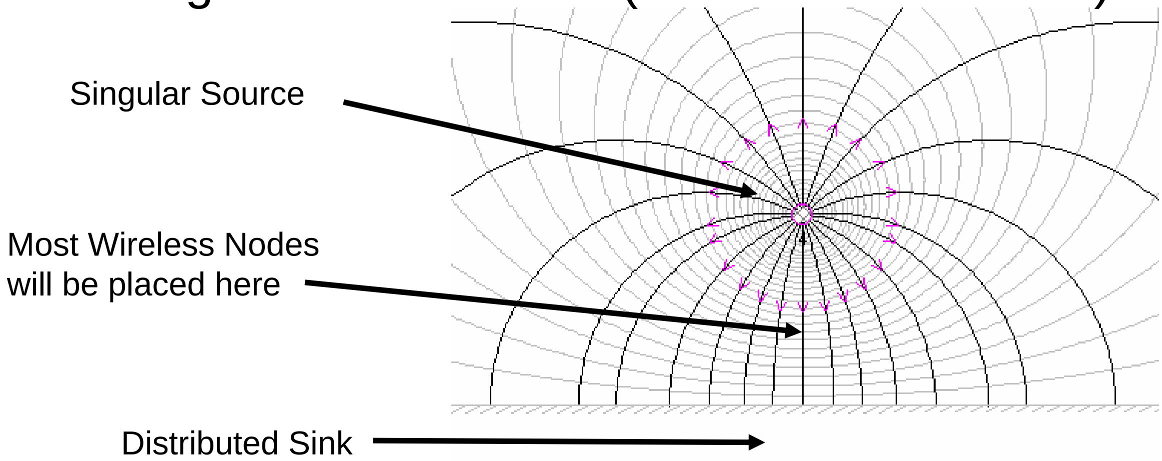
Example



- A singular source of information on top, and a distributed sink of information below

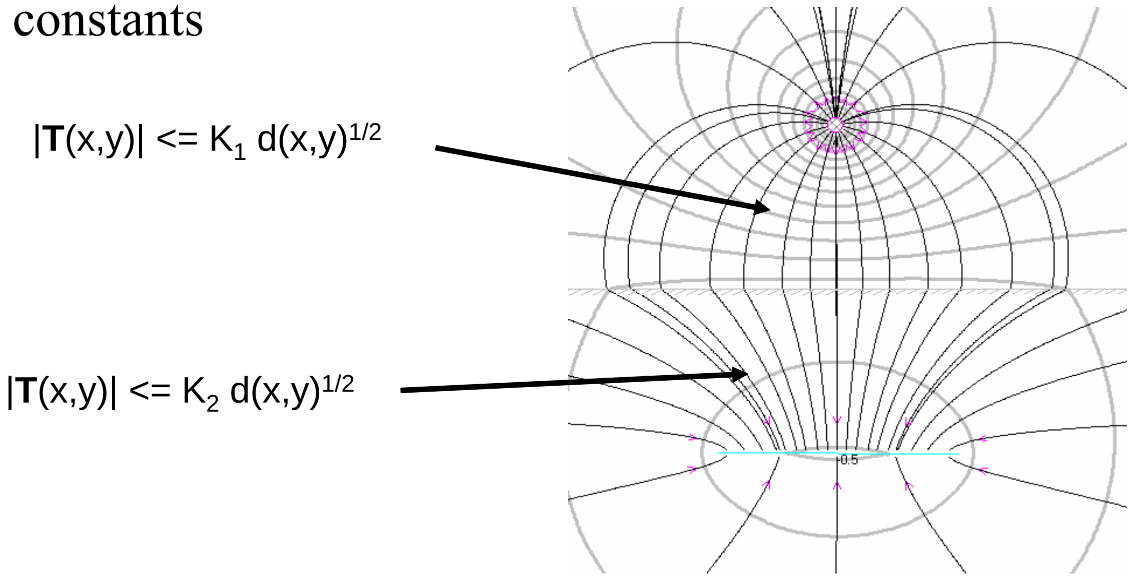
Degrees of Freedom in Placement of Sources/Sinks

- If we are free to place sources and sinks wherever we want on the surface of areas, the sources and sinks will be distributed like charges on conductors (Thomson's theorem)



Information Propagation in Different Environments

- Areas with different traffic carrying capabilities correspond to dielectrics with different dielectric constants



Access/physical layer control

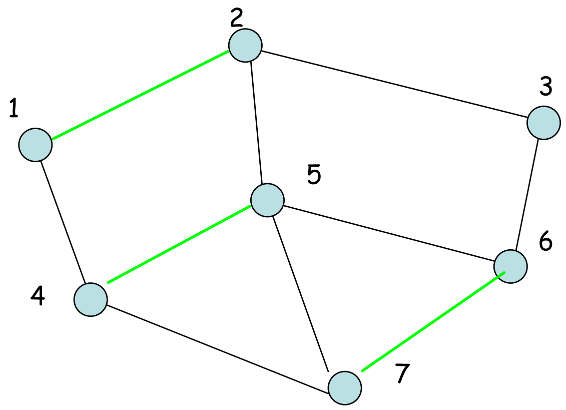
•Nodes control transmission power, access decision (transmit, don't transmit, which code (in CDMA) etc.), other physical layer parameters represented collectively by vector I(t)

• The environment changes as well due to mobility of the nodes and the environment itself; "topology" S(t)

• Cij (t)=Cij(S(t),I(t)): rate of bit pipe from i to j at t

• C(t) communication topology at time t determined partly by the environment S(t) (uncontrollable) and partly by the physical and access layer decisions I(t) (controllable)

Example Multihop, mobile ad-hoc network with SS signaling and single transceiver per node



Topology state S(t): the connectivity graph at t, indicating pairs of nodes that are within direct communication range

Access Control I(t): group of links designated to transmit at t. Any two links transmitting at the same time should not share a common node, thus I(t) a matching of the connectivity graph

Throughput capacity at the access layer

- Single hop traffic requirements

•Access Control vector I(t) represents the selection of various access and physical layer parameters at t

- Access Control policy designates I(t), t=1,2,… I(t) in A where A the collection of all possible access control vectors

Throughput capacity at the access layer: single hop traffic

Rate vector for some fixed state S(t)=s and access policy I(t)

$$C(s) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} C(s, I(t)), \quad I(t) \in A$$

Capacity region C(s) for fixed topology state s includes all rate vectors realized by any access policy

C(s) the convex hall of {C(s,I): I in A}

Capacity region C the expectation of C(s) with respect to the stationary distribution of topology process S(t) i.e.

C={C: C=E[C(s)], C(s)ε **C**(s)}

Dynamic Access Control to maximize throughput

- Select transmission rates to match demands adaptively
- **Max weight access control** policy selects I(t) to maximize

X(t)*C(s,I(t))

X(t) vector of packet backlog for each link

maxweight guarantees stability if arrival rate vector in C

Access control jointly with traffic forwarding

Select I(t) to maximize the following objective

$$\sum_{i,j=1}^{N} w_{ij} C_{ij}(S(t), I(t))$$

where

$$w_{ij} = \max_{m=1..N} \left\{ X_m^i(t) - X_m^j(t) \right\}$$

The joint scheme above achieves max end-to-end throughput

Dealing with complex optimization problems

Crucial step: select I(t) to maximize

$$\sum_{i,j=1}^{N} w_{ij} C_{ij}(S(t), I(t))$$

Use instead **randomized** low complexity scheduling

Select I randomly, let I(t) be equal to I or I(t-1) depending on which gives larger value to the objective function

Randomized scheme maximum throughput for a wide class low complexity randomization mechanisms

Randomized Algorithm for Access Control

- A randomized algorithm for optimal access control is represented by a probability distribution *P(X,.)* on *A*, parameterized by the weights X

  ↑ set of access v.

- Consider randomized algorithms with the property: if *I* has distribution *P(X,.),* then

$$(C): \; P(X : I^T = \max(X I^T)) \geq \epsilon > 0, \; \forall X$$

- Simple randomized algorithm with the above property: Select each I_{ij} by flipping a fair coin. If the resultant vector belongs to A, then that is I. Otherwise I = *0*.

  - Property *C* holds with $\epsilon = (1/2)^{NM}$

Closing Remark

Often times autonomy is not a choice but a necessity in current communication network designs and sophisticated new approaches are needed to deal with it