Scalable Coordination: Lessons from the Internet

Deborah Estrin

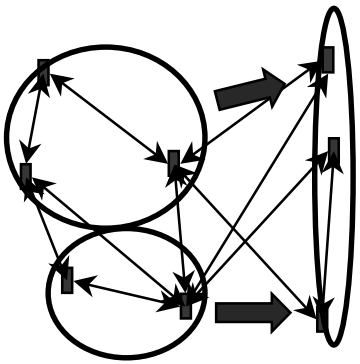
(Joint work with: John Heidemann, Ramesh Govindan, Chalermek Intanagonwiwat, Satish Kumar)

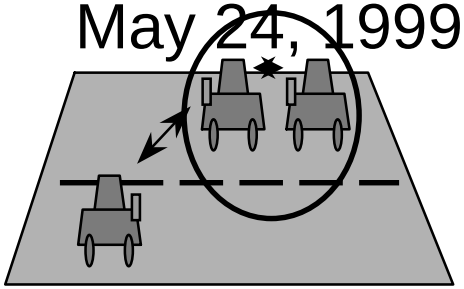
Computer Science Dept and

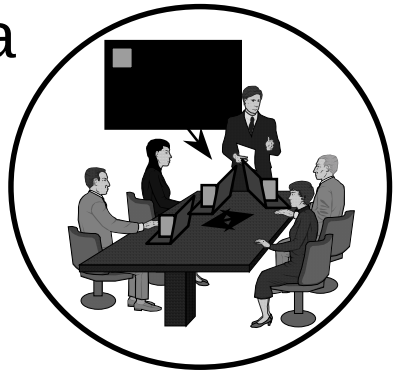
Information Sciences Institute

University of Southern California

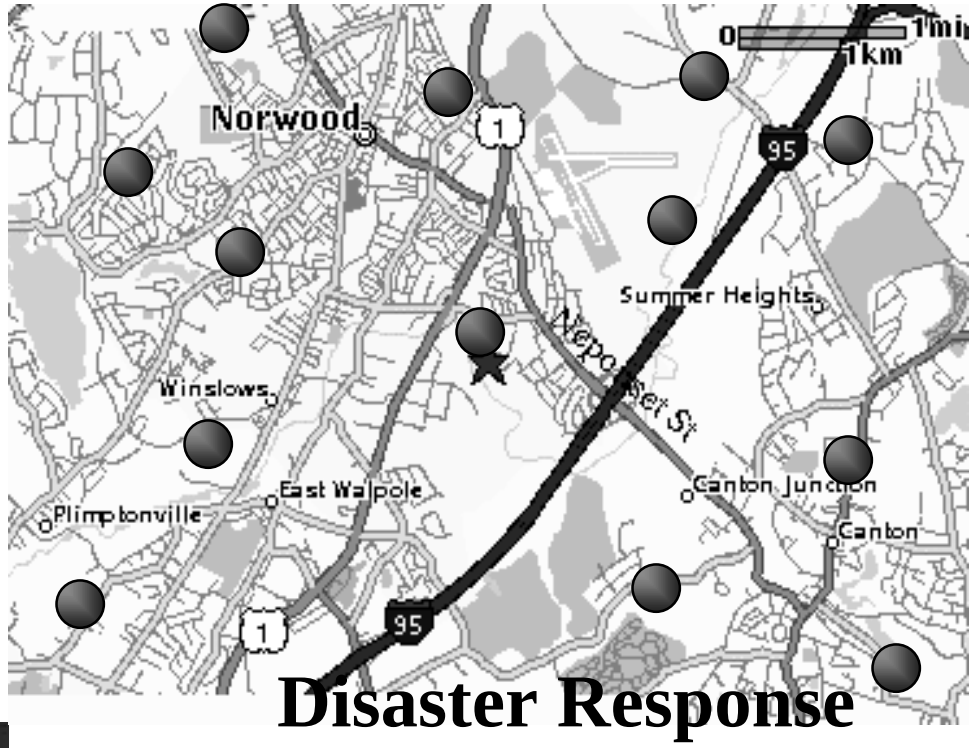
May 24, 1999

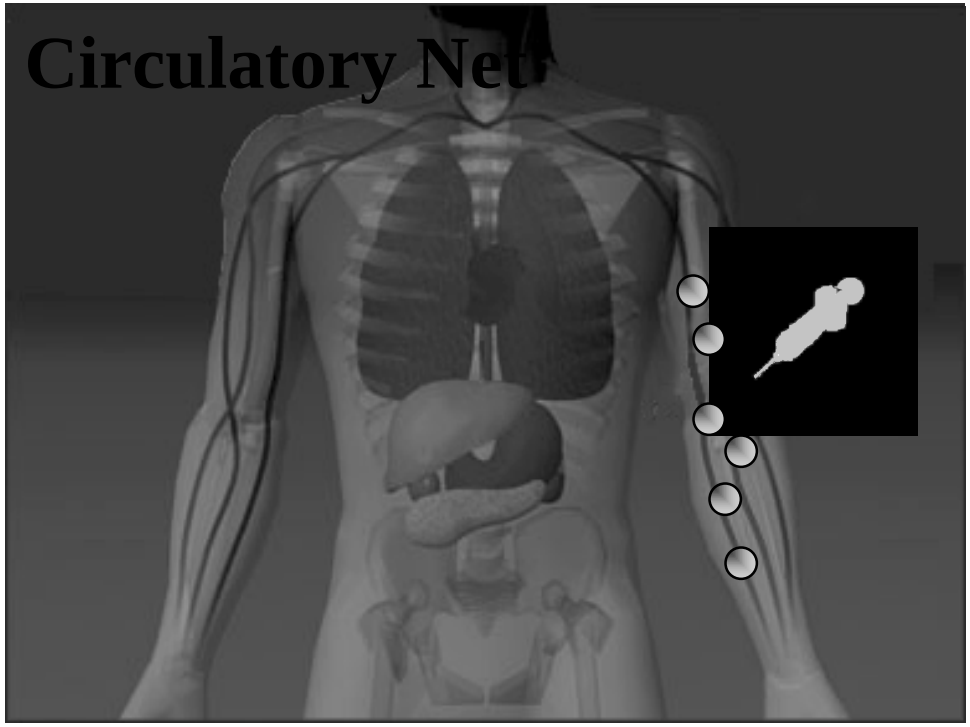






Embed numerous distributed devices to monitor and interact with physical world: in factories, hospitals, offices, homes, vehicles, and the human body





Leverage off pervasive physical locality between nodes and subject

Long-term motivating applications

- Embed large numbers of small, low-power, computationally powerful, communicating devices...
- Communicate to correlate and coordinate
- Design, deploy, and control ***robust*** distributed systems composed of tens of thousands of physically-embedded devices

The Challenge is Dynamics

- The physical world is dynamic
  - Dynamic operating conditions
  - Dynamic availability of resources--***particularly energy!***
  - Dynamic tasks
- Devices must adapt automatically to the environment
  - Too many devices for manual configuration
  - Environment is not under our control
- **Research challenge**

  Coordination and control algorithms for large scale, highly dynamic, unattended, distributed systems

Borrowing Ideas from the Internet

- Achieve desired global behavior through ***localized interactions***
  - Design for robust operation and incremental deployment
- ***Empirically adapt*** to observed environment--a priori assumptions are only hints
  - Design for continual change

Localized Algorithms: Examples

- Directed diffusion
  - Neighbor to neighbor propagation of Interests and data along gradients
  - Inspired by biological systems
  - Application to simple systems proposed by Van Jacobson
- Clustering
  - Hierarchy/aggregation key scaling technique
  - Use local algorithms to dynamically, adaptively configure clusters
- Adaptive Fidelity

Example 1: Directed Diffusion

- Basic concepts
  - Data is independent of producers and consumers
  - Data, interests and control diffuse via sequence of local interactions
  - Aggregation/local transformations in nodes
  - Use gradients to channel the propagation of data and interests
- Adaptability to network dynamics
  - Multipath delivery for robustness
- Network efficiency balanced with robustness
  - Longer network lifetime by adapting to available resources and steady state

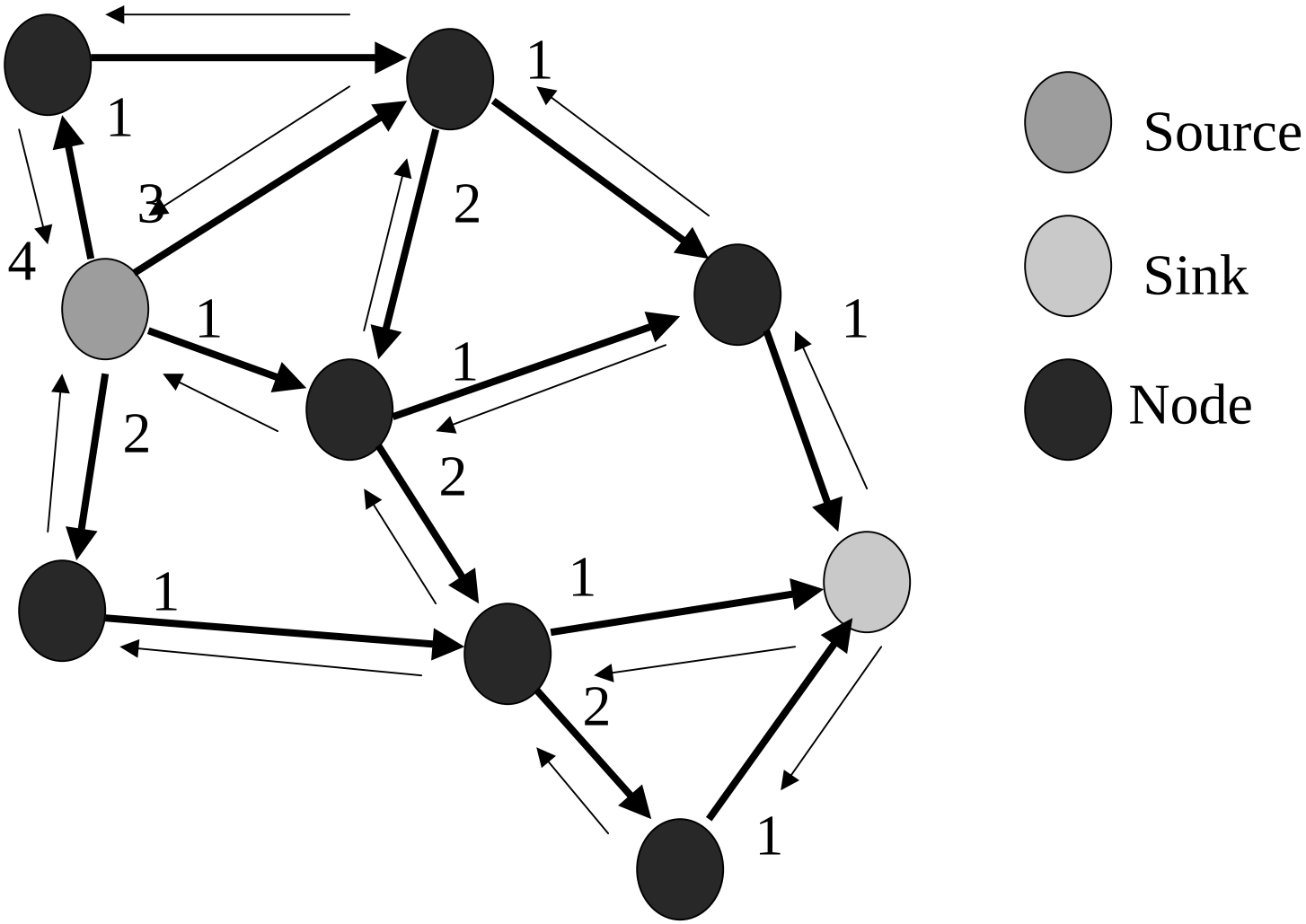
Directed Diffusion: Local Algorithms

- Gradient Establishment
- Reinforcement
- Inhibition

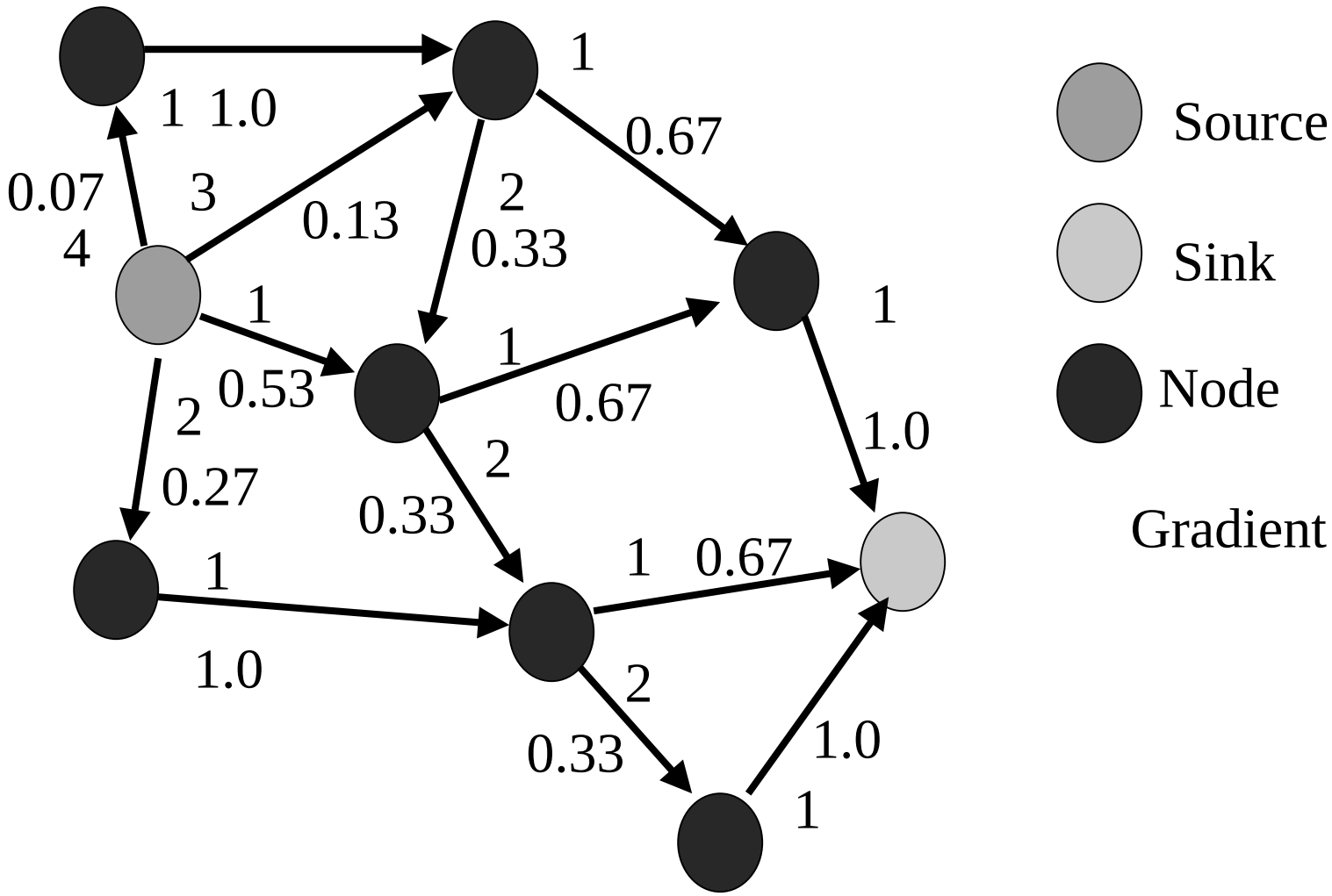
Gradient Establishment

- We have many ways to do it. This is just one of them.
- Sink expresses interest for data
- Each node ranks each link giving it interest
- Gradient = 2^(n-rank)/(2^n - 1)

Ranking



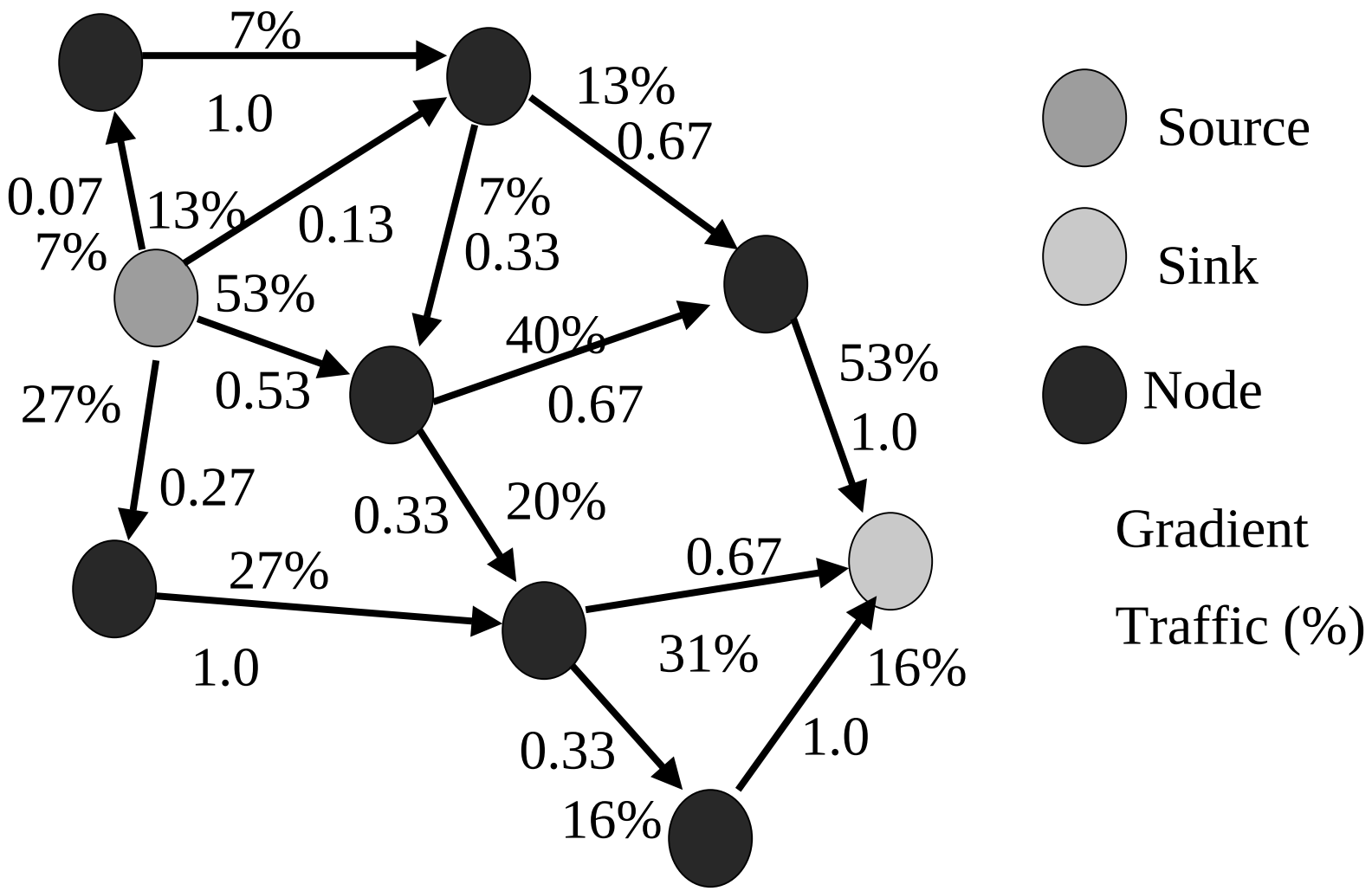
From Rank to Gradient



Data Dissemination

- Single-path or multi-path delivery
- Traffic is sent proportional to gradient
  - deterministically, or probabilistically
  - redundantly with different resolution, or distinctly
- In this example, we show multi-path delivery with no redundancy.

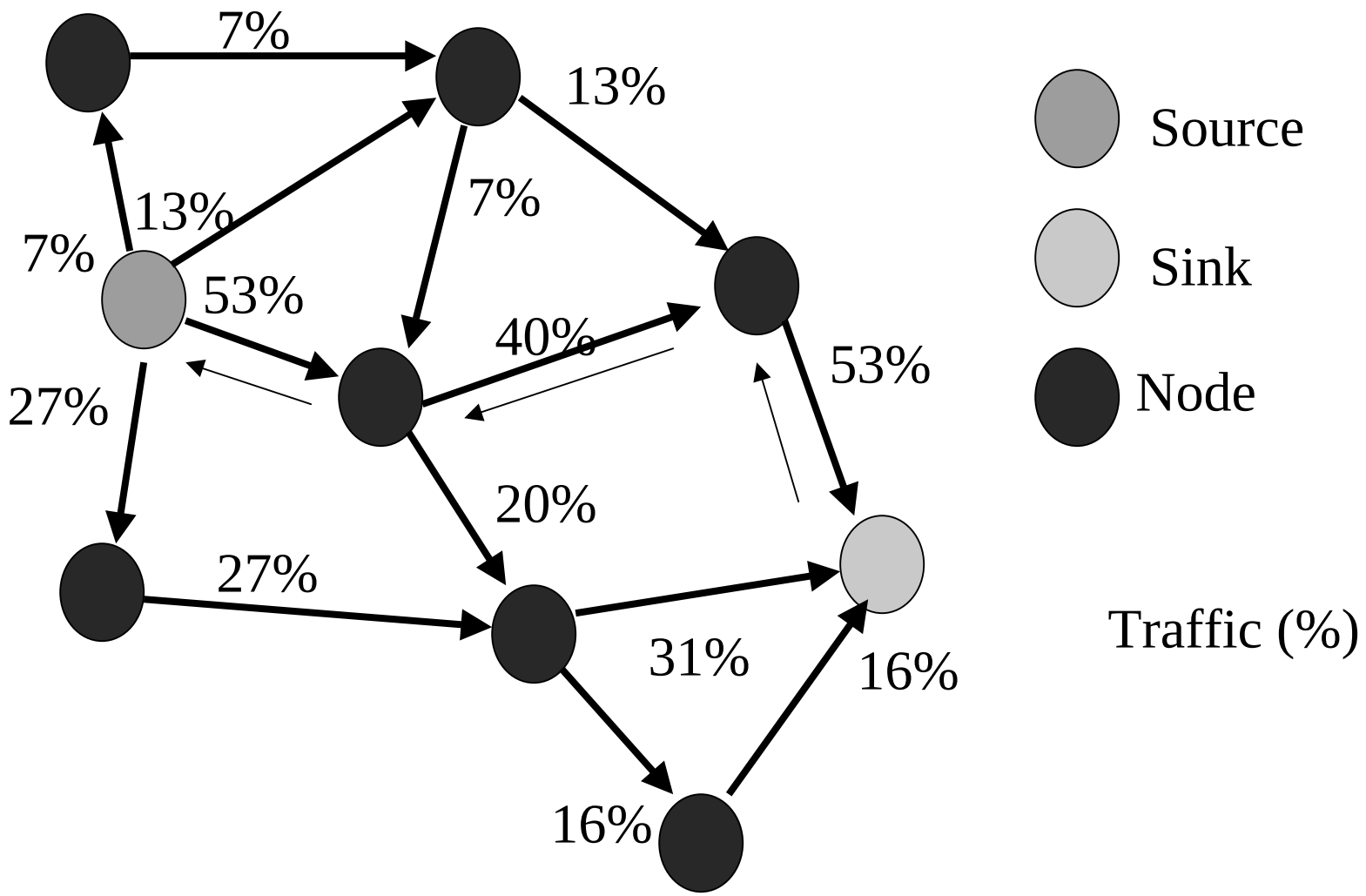
Traffic on each link



Reinforcement

- Each node can reinforce a link based on
  - amount of data received from the link
  - average delay or variance
  - average loss or variance
  - etc.
- In this example, we use amount of data received.

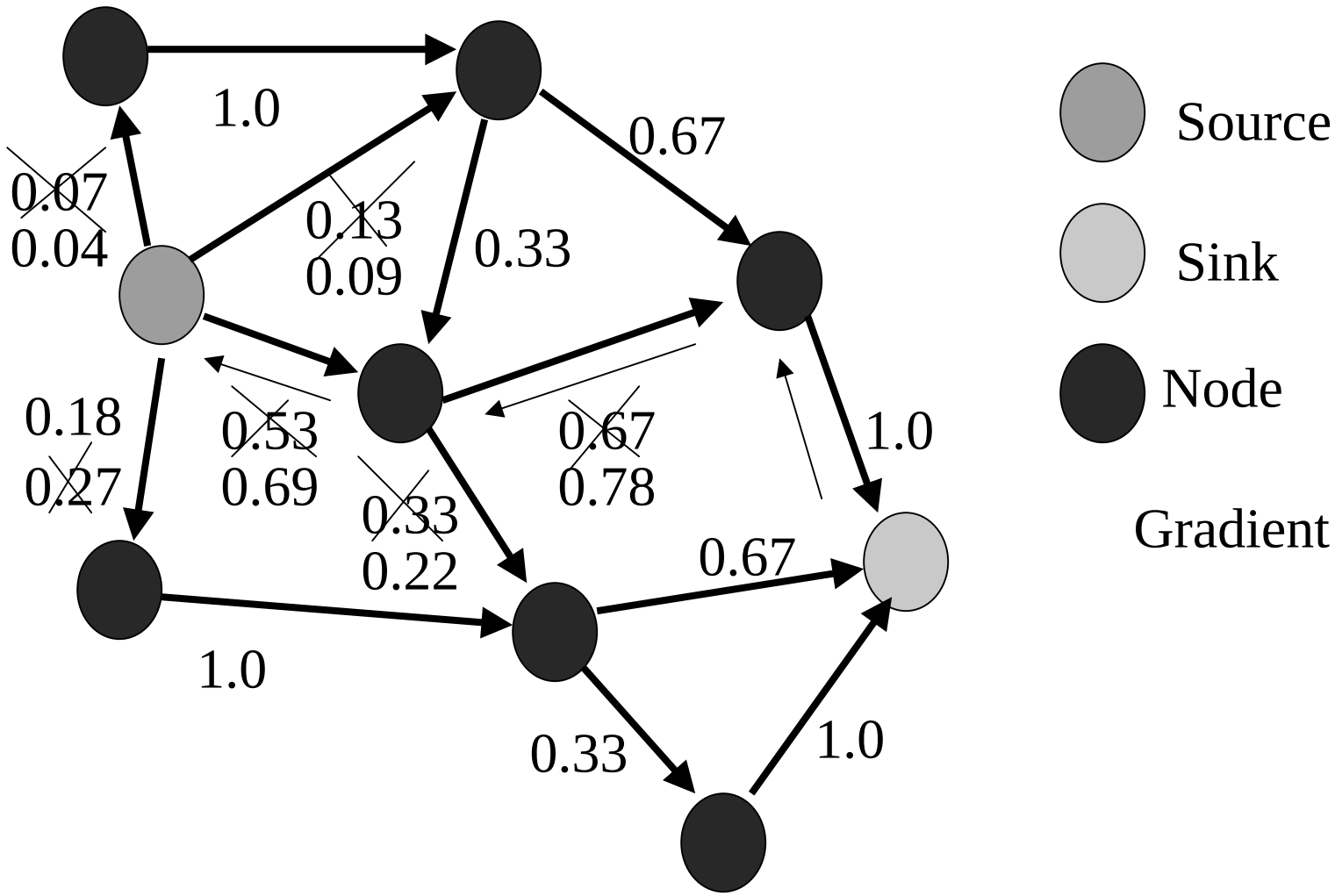
Reinforcement Example



Reinforcement (cont.)

- Upon receiving a reinforcement signal, the node recalculate gradients.
- Again, we have many ways to update gradients. Here is one.
- Gi = Gi + 0.5 where i is the reinforced link
- Normalize all gradients to keep sum = 1

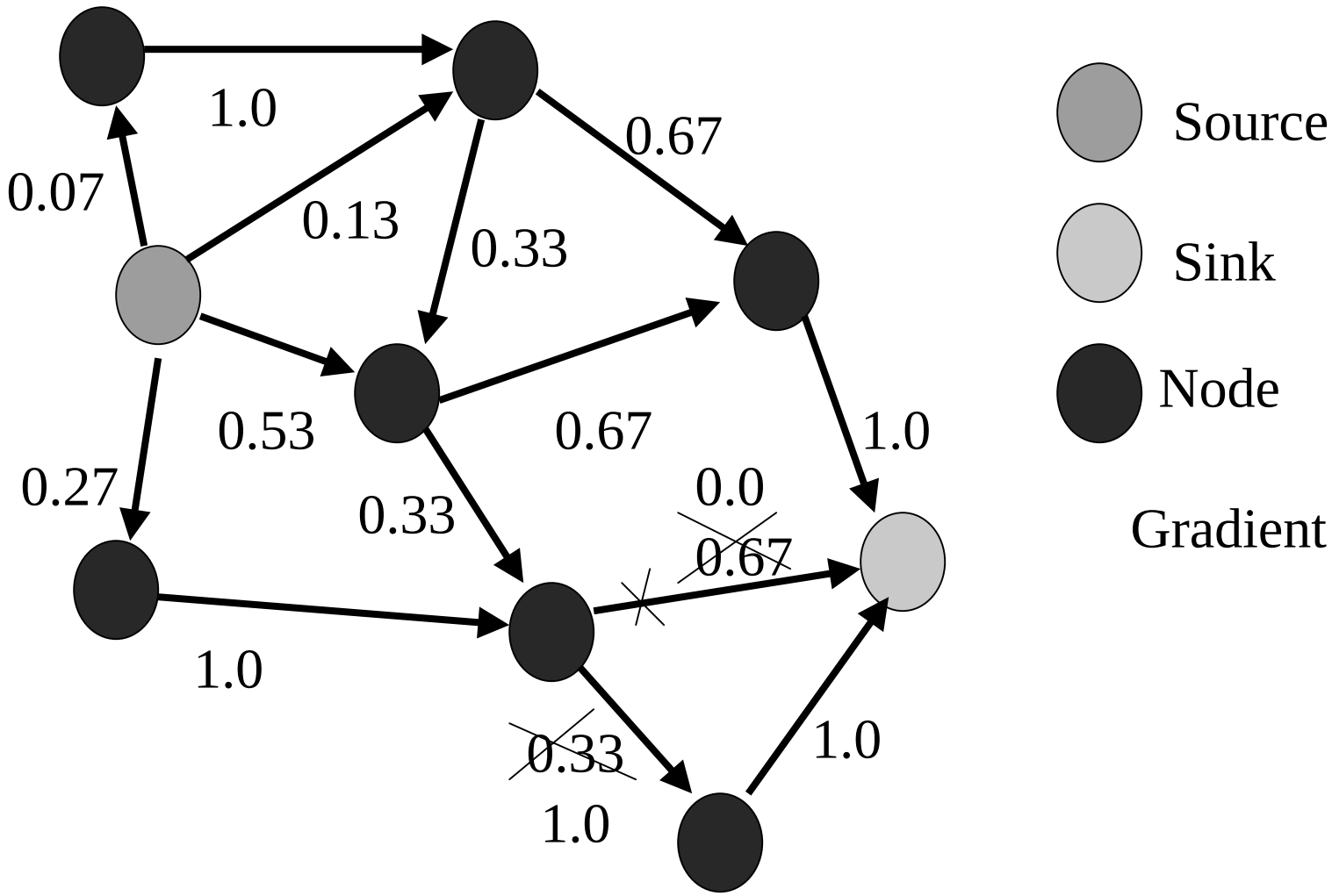
Reinforcement (cont.)



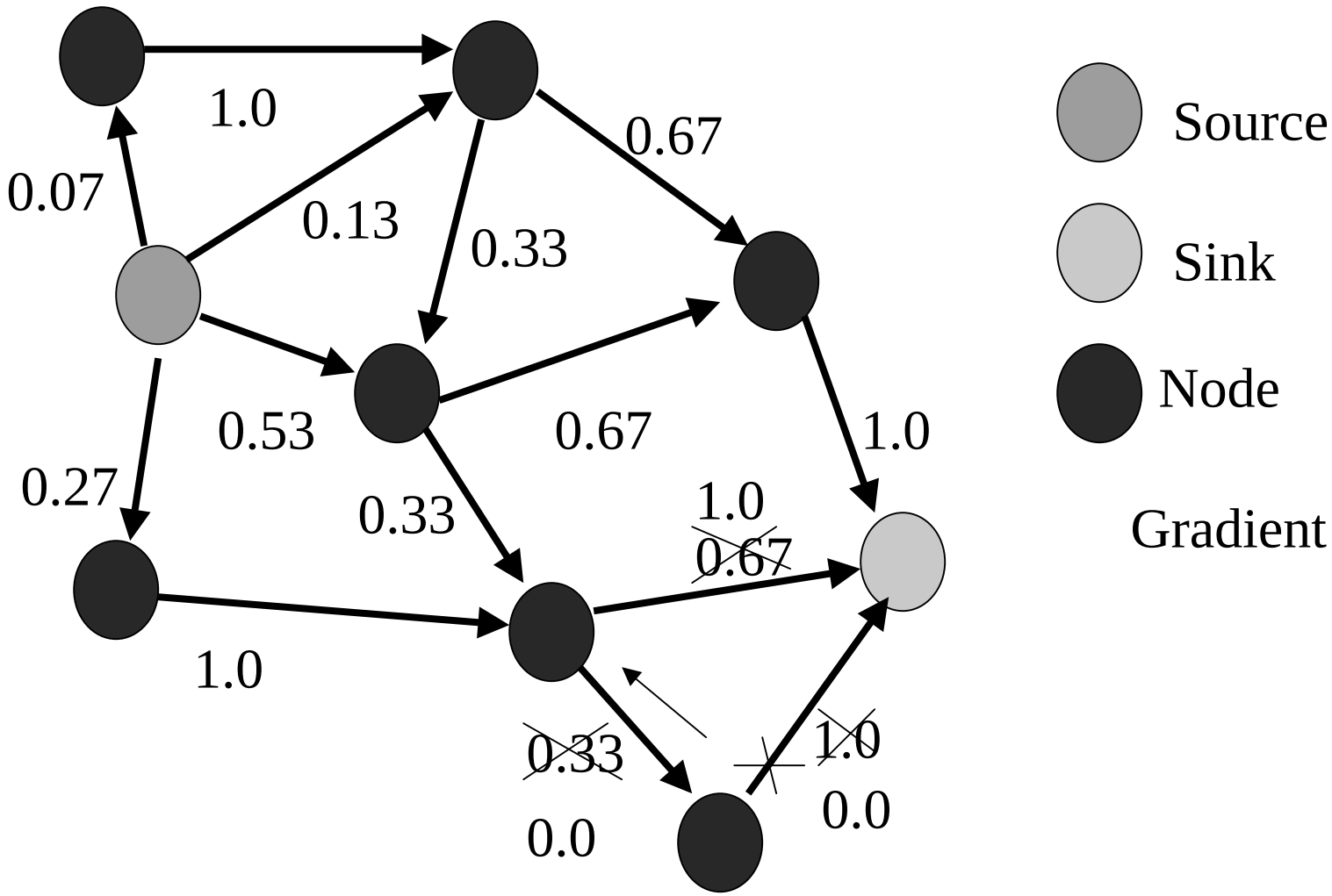
Adaptability to network dynamics

- Set gradient for the failure link to be 0 then normalize gradients.
- If the failure link is the only alternative, send inhibit signal upstream.
- We treat the link giving inhibit signal just like the failure link.

Network Dynamics Example



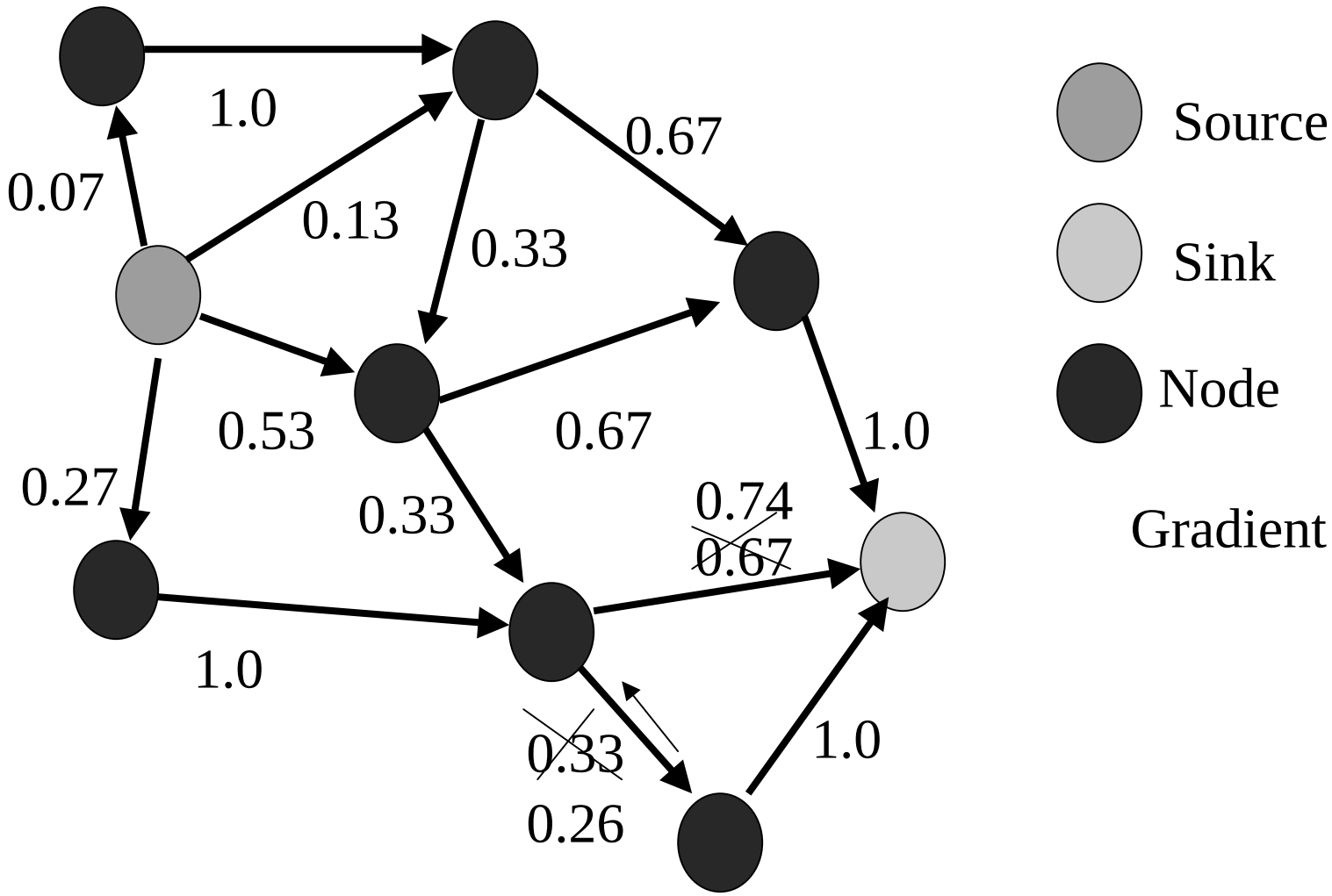
Inhibition signal example



Longer network lifetime

- Some nodes may have limited power, so they may want to deflect traffic to other directions.
- They will send negative reinforcement. Again, here is a way to do it.
  - Upon receiving negative reinforcement
    $G_i = G_i - 0.1$ and normalize all gradients

Negative Reinforcement



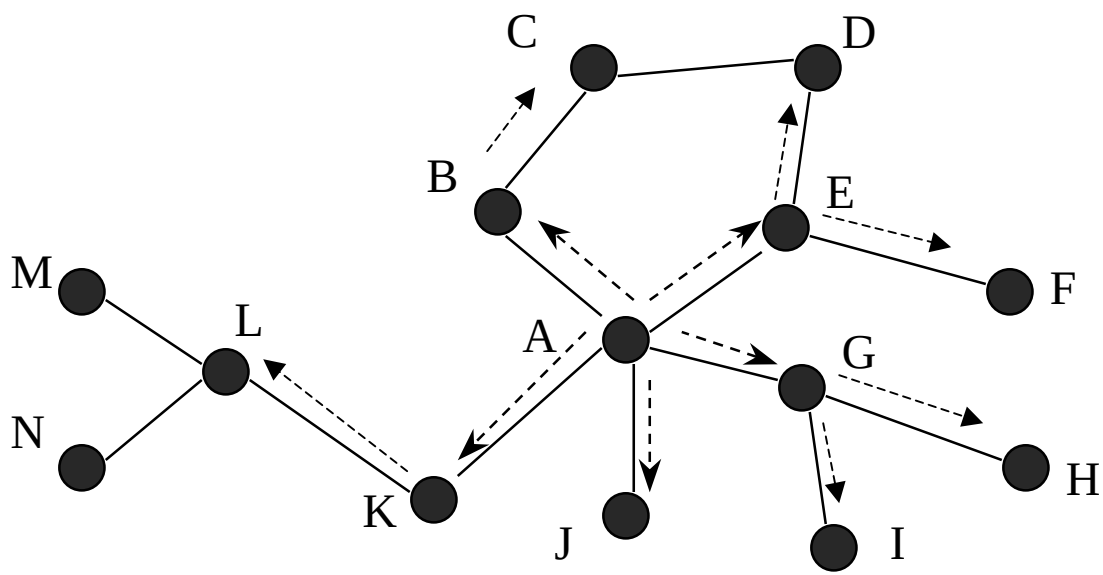
Example 2: Localized Clustering

- Supports improved scaling, robustness, and resource utilization for sensor control tasks
  - Summarizing events
  - Deciding sensor on-off schedules
  - Object location
- Sensors self-organize into multi-level hierarchy with minimal configuration
  - Improved scaling of state and communication overhead
  - Efficient adaptation to network dynamics
  - Life of the network increased through adaptation to changing energy levels

Local interactions

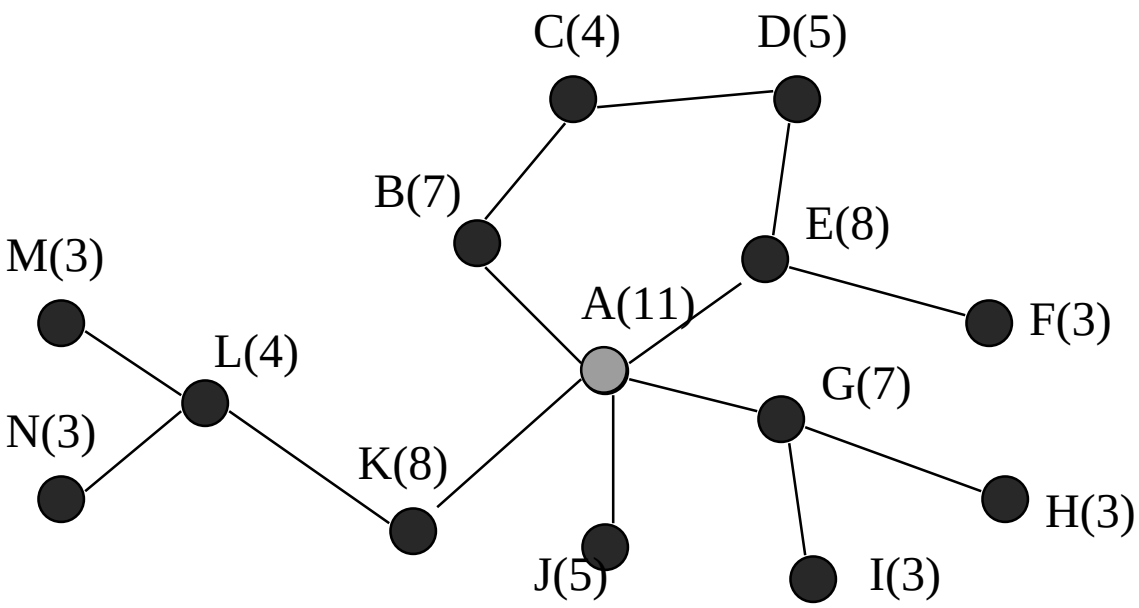
- Promotion/Demotion based on remaining energy and messages from local neighborhood
- Periodically advertise potential children for each of the sensor's levels
- Choose appropriate parent (e.g., closest) for each level using received advertisements
- Stop promotion after specified level reached or no other advertisements seen for a given level

Hierarchy construction



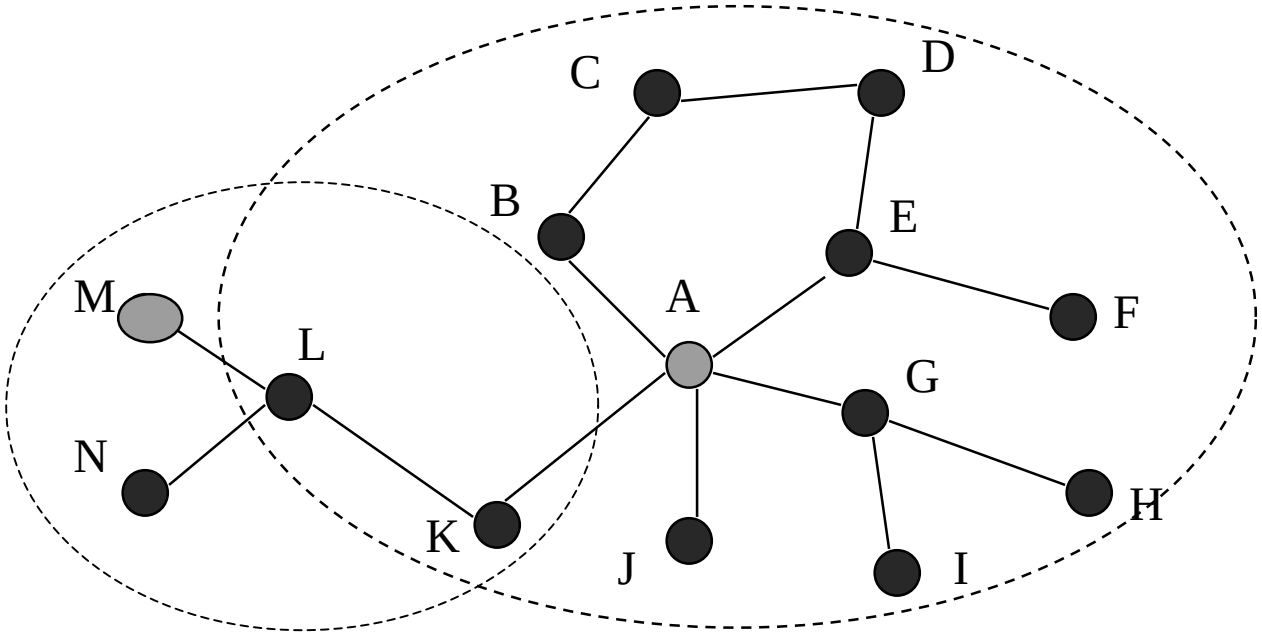
- Sensors initially at level 0
- Sensors send advertisements to other sensors within r0 (e.g. 2) hops and start a wait timer proportional to r0

Promotion criteria



- After wait timer expires, sensors start promotion timer based on energy and number of sensor advertisements heard
- Sensor promotes itself to level 1 when timer expires if it does not already have a parent
  - e.g. A hears more advertisements, so promotes itself first

Promotion criteria and Parent selection

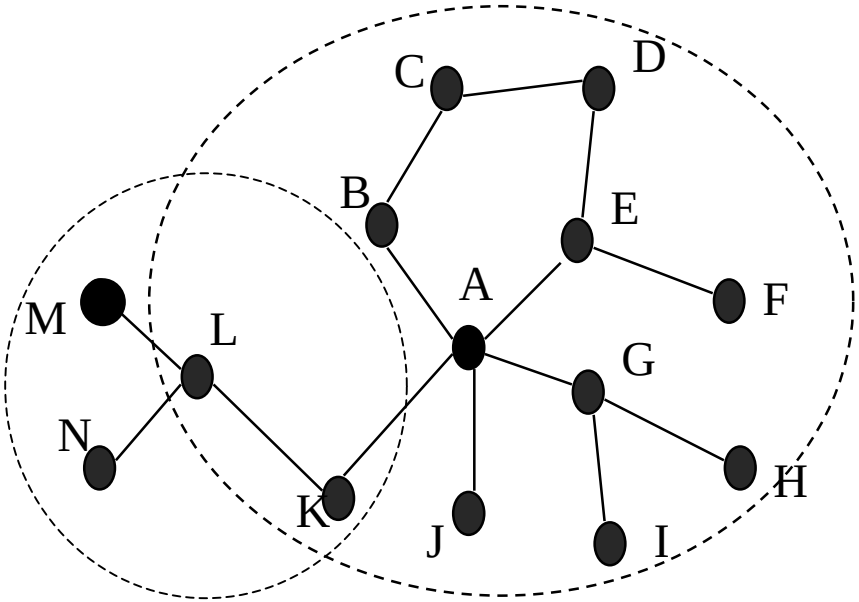


- On promotion, level 1 sensor advertises potential children (I.e., IDs of level 0 sensors heard from) to r1 (e.g., 4 hops)
  - e.g. A at level 1 sends adverts with B,C,D,E,F,G,H,I, J, K ,L
- Level 0 sensor picks closest level 1 sensor as parent if its ID is in potential children list of level 1 sensor
  - Ensures parent and child LMs see each other

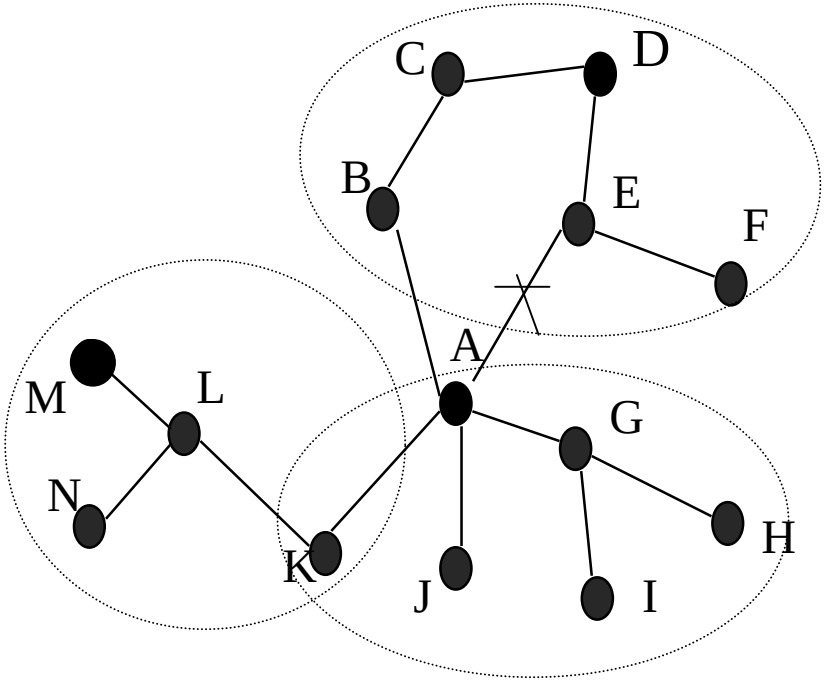
Promotion/Demotion Criteria

- Nodes successively promote themselves until
  - Specific level reached or
  - No other same level sensors exist (I.e., sensor is root of the hierarchy)
- Sensor demotes itself:
  - If it has no children and it can see a potential parent, or
  - If all its children are covered by potential parent, or
  - If its energy falls below a threshold function of its children's energy (e.g., less than 50% of the maximum energy among its children)

Hierarchy Adaptation



A) Before any failure

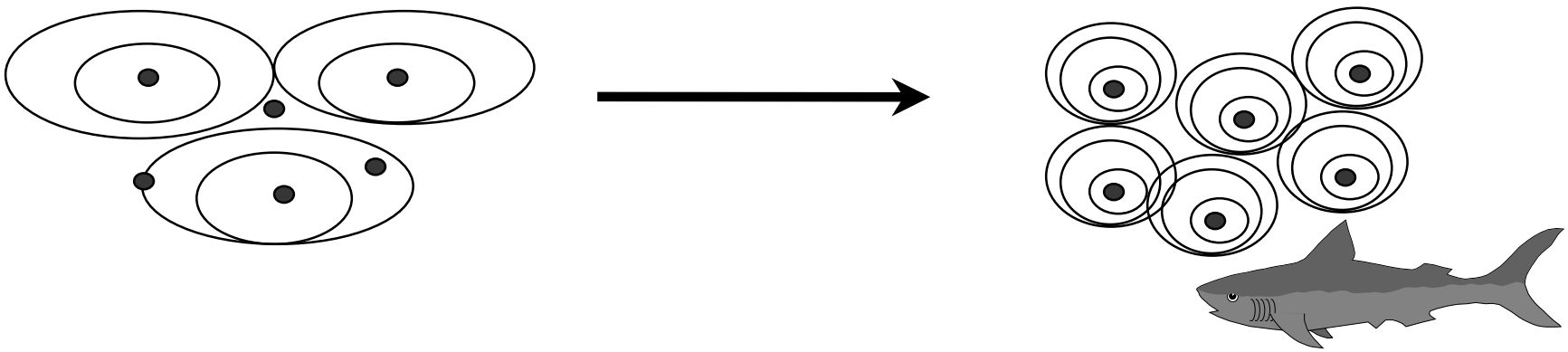


B) After “link” A-E fails

- Sensors continually start wait timers
- At end of wait timer
  - If no parent seen, sensors start promotion timer based on advertisements heard and local energy (unless they are at highest level) or
  - Sensors demote themselves if any of the demotion conditions occur

Example 3: Adaptive Fidelity

- To get a better picture turn on more sensors
  - Nodes adjust their coverage, sampling rate, communication frequency based on neighbor density, power levels, reports from direct neighbors...



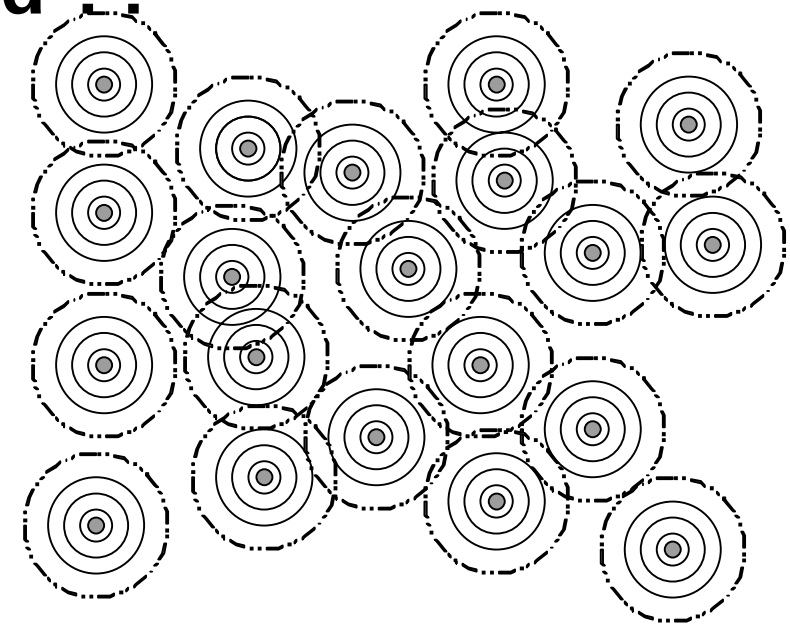
  - Automate analysis of ways to improve fidelity: self-configure to mobilize more nodes when needed, or turn-off nodes when not needed to extend lifetime

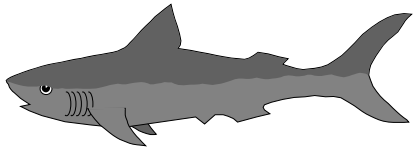
Challenge for Global System Characterization

Given a system composed of nodes running locally adaptive algorithms, how do we characterize and quantify global behavior??

sources required ??

data accuracy ??





responsiveness ??

cascading failure modes ??

Radio assumptions we would like to understand better

- Adjustable communication range
- Relative cost of transmit, receive, idle (but cue-able)
- Interference
- Near range propagation models
- Technology trends