TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks

Presented by Smith & Klein
ODU-CS895 Mobile Computing

Background[1]

- Application: Remote Sensors (vs cell phones, e.g.)
- Sensor net nodes unaware of topology characteristics
- Congestion arises from standard, 2-state implementation
  - Dormant State
    - Conserves energy (lengthens net lifetime)
    - Characterized by low traffic volume
    - Limited number nodes (routing path) concurrently active
  - Crisis State
    - High traffic volume
    - More data coming in than nodes can handle = congestion
    - Data loss may occur (dropped packets) → loss of "fidelity"

[1]Note: Includes intro & related work sections

Background (continued)

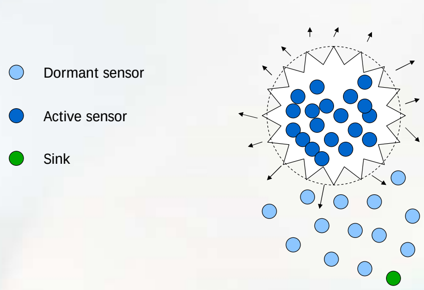
- Traditional traffic control methods are unacceptable
- Most address capacity (traffic control)
  - CODA, congestion detection/avoidance
  - Proportional bandwidth allocation
  - Limit source rate
  - Flow control
- Some ~address fidelity (via traffic control)
  - Event-to-Sink Reliable Transport protocol (ESRT)
  - Prioritized medium access control (MAC)
- Ideal: Balance fidelity & capacity
  - Reducing source traffic → critical information lost
  - Problem occurs from topology-unaware aspect
  - Awareness is hard problem
    - Need end-to-end network knowledge
    - Need local picture, too

Congestion in Sensor Networks

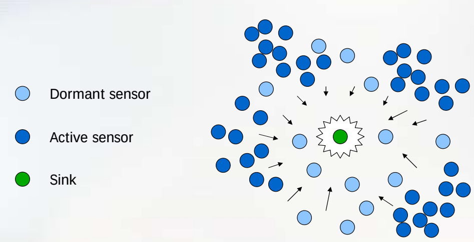
- Congestion = incoming data > transmission capacity
- 3 Typical hot spot scenarios:
  1. Source Hot Spot
  2. Sink Hot Spot
  3. Intersection Hot Spot
     - Merging traffic
     - Crossing traffic
- Sensor nets are dynamic
  - Deployment strategies can mitigate source & sink hot spots
  - Intersection hot spots need ad hoc solutions

Consider the examples on the following slides

Scenario 1: Source Hot Spot

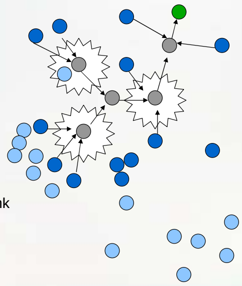


Scenario 2: Sink Hot Spot



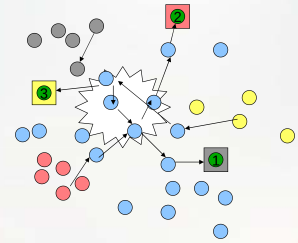
Scenario 3a: Intersection Hot Spot (Merging Traffic)

- Dormant sensor
- Active sensor
- Sink
- Sensor on route to sink



Scenario 3b: Intersection Hot Spot (Crossing Traffic)

- Path/dormant sensors
- Sinks (1, 2, 3)
- Sensors: source #1
- Sensors: source #2
- Sensors: source #3

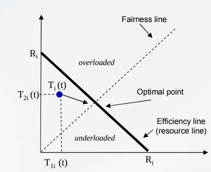


Traffic Control vs Resource Control

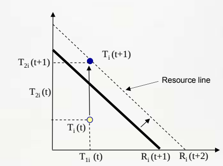
- **Traffic Control**
  - **Lots of research in this area**
  - **Assumes fixed resource provisioning**
  - **AQM used in wired networks**
  - **Resource utilization and fairness (balancing data volume with resource availability for optimal result)**
  - **Not as practical in sensor nets because of sensor net lifecycle is dynamic (varies with input, battery life, interference, etc)**
- **Resource Control**
  - **Goal is fidelity management (get key data through)**
  - **Bring extra resources on line when they're needed**
  - **Few efforts geared toward this aspect**

Effective resource control requires topology awareness – which ones to turn on/off to maintain optimal performance

Frameworks of traffic control and resource control



(a) Traffic control



(b) Resource control

- Optimal efficiency occurs when

$$T_{1i}(t) + T_{2i}(t) = R_i$$

- This is when traffic = resource availability
- Unrealistic for sensor net

- Resource availability varies with time

$$R_i(t+1) \rightarrow R_i(t+2)$$

- More resources are allocated to higher fidelity-requirement path

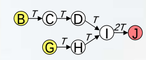
Capacity Analysis Model Description

- **Capacity = max throughput of sinks**
  - **Less than perfect 1-hop max because of interference**
  - **TARA estimation based on theoretical optimum**
    - **$(m/n)C^{max}$**
    - **(m/n) is the capacity fraction (m=n in 1-hop)**
    - **C^{max} is the 1-hop maximum throughput**
- **4 mapping steps to estimate capacity**
  1. **Topology map**
  2. **Spatial interference graph**
  3. **Spatial reuse graph**
  4. **Colored graph**
  5. **Time frame assignment**

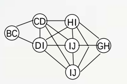
Capacity Analysis Model: Graph Example

- **Core of the TARA approach**
- **Estimates end-to-end throughput of different topologies**
- **Uses graph-coloring method to solve**

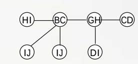
Goal: Capture the degree of interference of a given topology



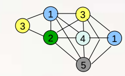
Topology Graph



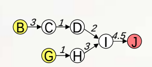
Spatial Interference Graph



Spatial Reuse Graph



Colored Graph



Time Frame Assignment

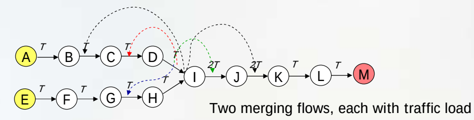
Estimating Capacity (for a Large-Scale Network)

- Number of nodes has big impact on time
  - Graph coloring takes time to solve
  - NP hard problem
- Re-image the topology in terms of bottleneck links
  - Reduces number of nodes addressed
- Calculate only the capacity fraction of bottleneck
  - This will be a smaller subset
  - Can be calculated within acceptable time limits

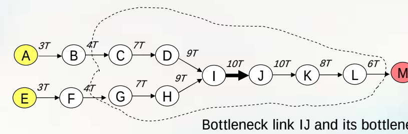
Finding Bottlenecks

Congestion Sum **of a link indicates if it is a bottleneck**

Bottleneck Zone **is much smaller than whole topology**



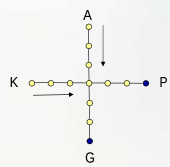
Two merging flows, each with traffic load T

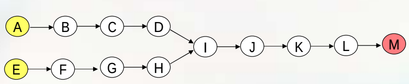


Bottleneck link IJ and its bottleneck zone

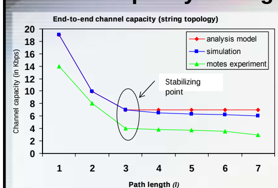
Topologies

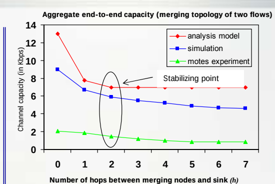
- Intersections occur in many topologies
- Three key types are considered
  - String
    - One path from I to J to K to M
    - Loss of data due to "full" path
  - Merging
    - Path from A to M and E to M intersect at I
    - Sink can become overloaded
  - Crossing
    - Path from A to D and B to E cross at C
    - Multiple cross types possible





Channel Capacity: String & Merging Topologies





$$\varepsilon = \begin{cases} 1 & \text{if} \quad l = 1 \\ ½ & \text{if} \quad l = 2 \\ \mathbf{1/3} & \text{if} \quad l \geq 3 \end{cases}$$

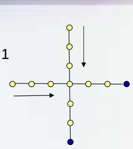
Leads to Lessons #1 & #2

$$\varepsilon = \begin{cases} n/(n+1) & \text{if} \quad h = 0 \\ n/(2n+1) & \text{if} \quad h = 1 \\ n/3n = \mathbf{1/3} & \text{if} \quad h \geq 2 \end{cases}$$

Leads to Lesson #3

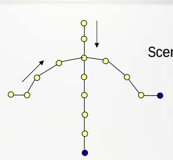
Crossing Flow Scenarios

Scenario 1

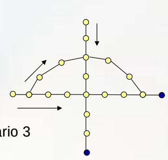


Forwarder congestion by crossing traffic

Scenario 2

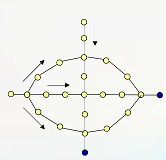


Alleviating congestion by alternative path



Scenario 3

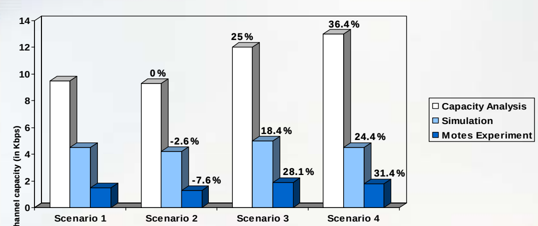
Alleviating congestion by additional path



Scenario 4

Alleviating congestion by multiple additional paths

Channel Capacity of Crossing Flow Scenarios



Aggregate end-to-end channel capacity (crossing topology of two flows)

This leads to Lesson #4

"Actual experiments have more interference than simulations… [and] more complicated interference than the model"

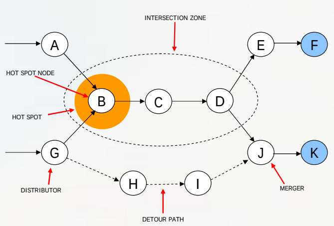
Lessons

1. Shortening a string topology does not increase the capacity if the resulting capacity fraction remains the same
2. If the node whose incoming traffic volume is less than C^{min} experiences congestion due to interference with other flows, the congestion can be eliminated by re-routing traffic onto the *non-interfered* path
3. Capacity of a merging topology can be increased by moving the merging point within a small number of hops from the sink
4. To increase capacity of a crossing topology, at least one flow should have multiple paths, and split its traffic onto these paths

Topology-Aware Resource Adaptation Scheme

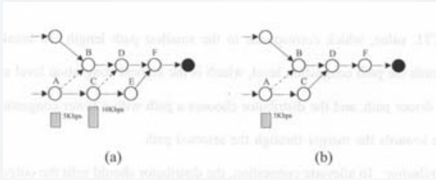
- Increasing Resources During Crisis States
  - Congestion detection & hot spot node
    - Actual congestion detection much harder than in wired networks
    - TARA looks at both buffer occupancy and channel loading
    - Hot spot declared when upper watermark level is reached
  - Traffic distributor
    - Maintain extra data field (neighbor packet injection count)
    - Control packet throttles upstream until a distributor is found
  - Traffic merger
    - Downstream control packet seeks merger point
    - On path between distributor and sink
    - Location and congestion are both important (affects capacity)
    - (slide 22, node J qualifies as merger)
  - Detour path
    - Uses REQ packet flooding to locate candidate distributor
    - Ties broken by path congestion comparison
  - Traffic distribution
    - Split outgoing traffic between original and detour paths
    - Streams aggregate data to same sink
    - Weighted fair share scheduling

Illustration of TARA



Topology-Aware Resource Adaptation Scheme (continued)

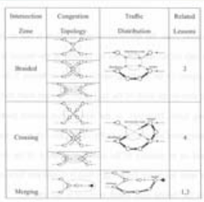
- Shrinking Resources During Dormant States
  - Increase resources during congestion
  - Return/reduce during low traffic



Shrinking resource provisioning using low watermark

Intersections & Congestion

- Interference zones for 2 flows (column 2)
- Possible detour paths & traffic distribution (column 3)
- Column 4 applies to lessons (slide 20)



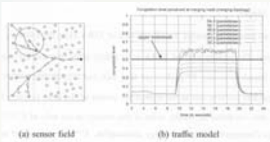
Three types of intersection zones

Performance Evaluation

- Performance metrics
  - Fidelity index: F/F^0
  - Total energy consumption:

$$E^{total} = \sum_{i=1}^{n}(p^{send} \times t_i^{send} + p^{recv} \times t_i^{recv} + p^{idle} \times t_i^{idle}),$$

- Simulation environment



Topologies for congestion control strategies

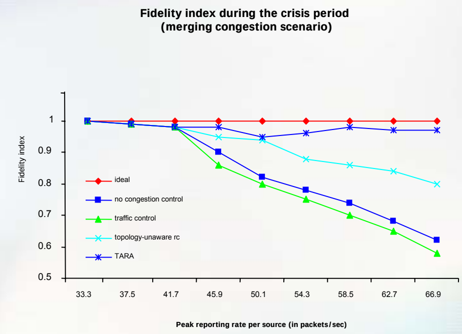
Topology examples with crossing intersections

No congestion control and traffic control

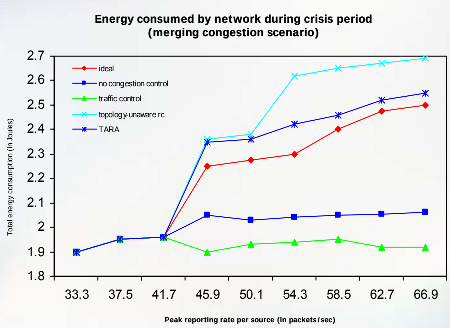
Topology-unaware rc

Ideal rc and TARA

Fidelity index during crisis period

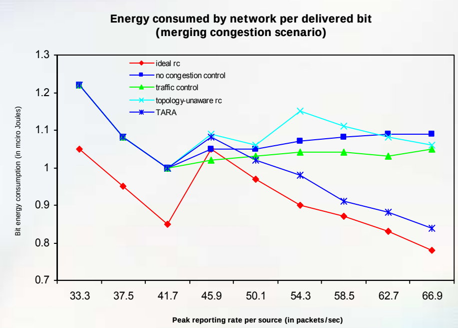


Total energy consumption by network during crisis period



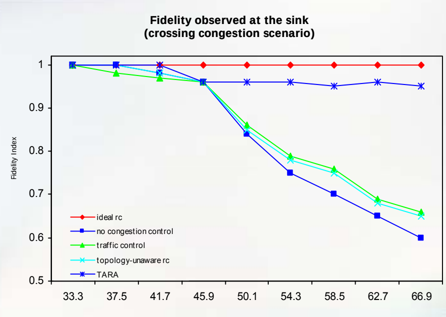
TARA delivers 70% more packets @ 33% increased energy consumption (but only 4% higher than ideal energy consumption at this delivery rate)

Bit energy consumption during crisis period

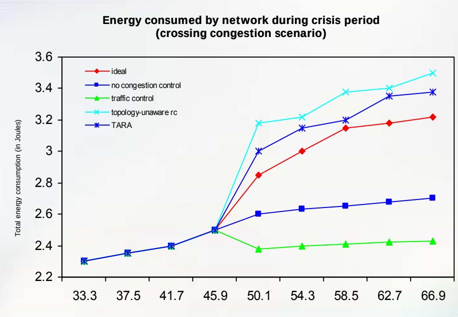


Detour path is more utilized as traffic rate increases – not a linear event

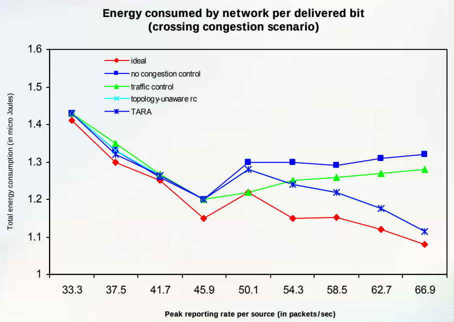
Fidelity index during crisis period



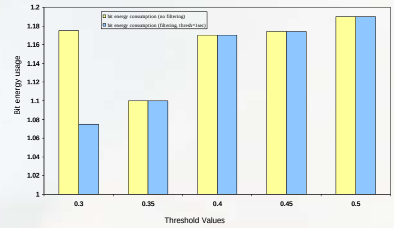
Total energy consumption by network during crisis period



Bit energy consumption during crisis period



Comparison with upper watermark values



Setting the appropriate watermark level is tricky

Stated goals

- **Primary goal: satisfy fidelity requirements**
  - **Is this achieved?**
  - **Could something else do a better job in this application scenario?**
- **Total energy consumption**
  - **Specific goal not stated**
  - **Assumed goal is to not cause excess energy consumption**
- **Bit energy consumption**
  - **Again, specific goal not stated**
  - **Assume target is effective energy usage wrt end user application**

Discussion: Positive Points

- **Predict reader's questions and address them.**
  - **Examples**
- **Used 3 approaches to test their idea**
  - **Mica2 motes running TinyOS**
  - **NS2 simulation**
  - **Model's theoretical results**
- **Interesting theory, with apparently useful application**
  - **Mica motes experiments worked**
  - **Field testing would be nice**

Discussion: Issues

- **Minor point: grammatical errors scattered throughout**
- **List only 3 hot spot scenarios. Are there others that should be addressed?**
- **Is graph coloring approach practical in a deployed sensor net with large numbers of nodes?**
- **How does this solution impact lifetime of sensor net (energy)?**
- **What happened to transient congestion?**
- **Does increasing paths (to relieve congestion) really result in increased fidelity (i.e., increased fidelity)**
- **Transitions from string & merging topologies to crossing topologies is awkward.**

References

- **Link to the author's web page on TARA: http://paul.rutgers.edu/~jwkang/research/tara.html**
  (note: the paper has not been published yet)
- **C. T. Ee and R. Bajcsy. *Congestion control and fairness for many-to-one routing in sensor networks*. In ACM SenSys, pages 134--147, 2004 http://citeseer.ist.psu.edu/context/2676965/0**
- **Bret Hull, Kyle Jamieson , and Hari Balakrishnan. *Mitigating Congestion in Wireless Sensor Networks* ACM SenSys 2004, Baltimore, MD, November, 2004. http://nms.lcs.mit.edu/papers/hbhfc-sensys04.pdf**