(Looking for) A SWiss Army Knife for Wireless Sensor Networks

Ioanis Nikolaidis
University of Alberta
Computing Science Department
<yannis@cs.ualberta.ca>

UNIVERSITY OF ALBERTA

Motivation

- Distance between literature and practice.
- Questions about the proper development tools.
- Search for what might be useful abstractions.
- Ability to quickly move from idea to realization.

Misplaced “Academic” Motivation

Early work in the area called for aircraft “raining” homogeneous sensor nodes to be used to monitor an unfriendly area (e.g., theater of war operations) and act autonomously for sufficiently long period to complete a task (until the end of war?!).

Realistic applications are much more different.

What Really Happened

- Largest military outdoor deployment was hierarchical and not at all random (manual placement to deal with RF propagation issues).
- Vehicle tracking easier to support from aerial assets (need not be expensive aerial assets).
- The essential problem is that of establishing intent, requiring sequence of images (high bandwidth).

- Scientific uses are even more constrainted (environmental issues related to battery disposal, habitat disturbance, etc.).

Source: Greg Pottie, November 2008.

Research Agenda

- Networking Using Small Capability Devices
- Application-Specific Sensor Network Protocols
- Abstractions for Sensor Application Development
- Information Management in Sensor Networks
- Services Infrastructure from Sensor Networks
- Scalable Simulation & Emulation Techniques

Where Academic Research “Fails”

- Partial Development of Ideas
- Restricted Scale of Experimentation
- Insufficient Explanation of Hidden Costs
- Entrenched Tools
- Entrenched Modelling & Simulation Techniques
- Implicit Assumption of More Powerful Devices

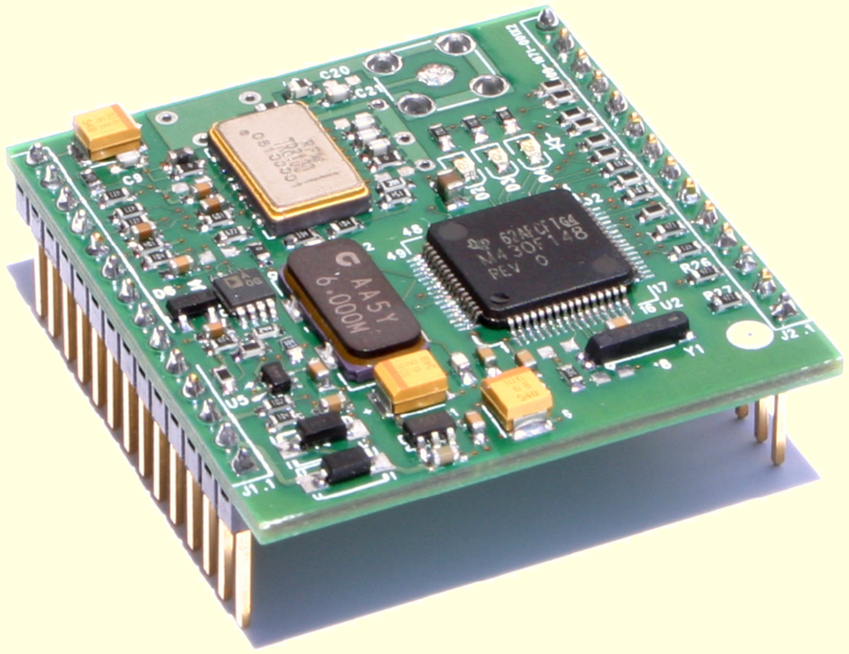
The Main Problem

- A significant part of the literature is implicitly assuming high capability devices (even for “elementary tasks” like routing).
  - Will they work in “really small” platforms?
- Moore's Law can be interpreted in two ways. The persistence on a single interpretation hinders our appreciation of future possibilities.
- What is more “important”?
  - 10 billion nodes at $10 a piece?
  - 200 million nodes at $500 a piece?
- What is easier to “upgrade”?
  - An electric shaver?
  - Your most recent version of MS Windows?

Our Toys

Platform: DM2200

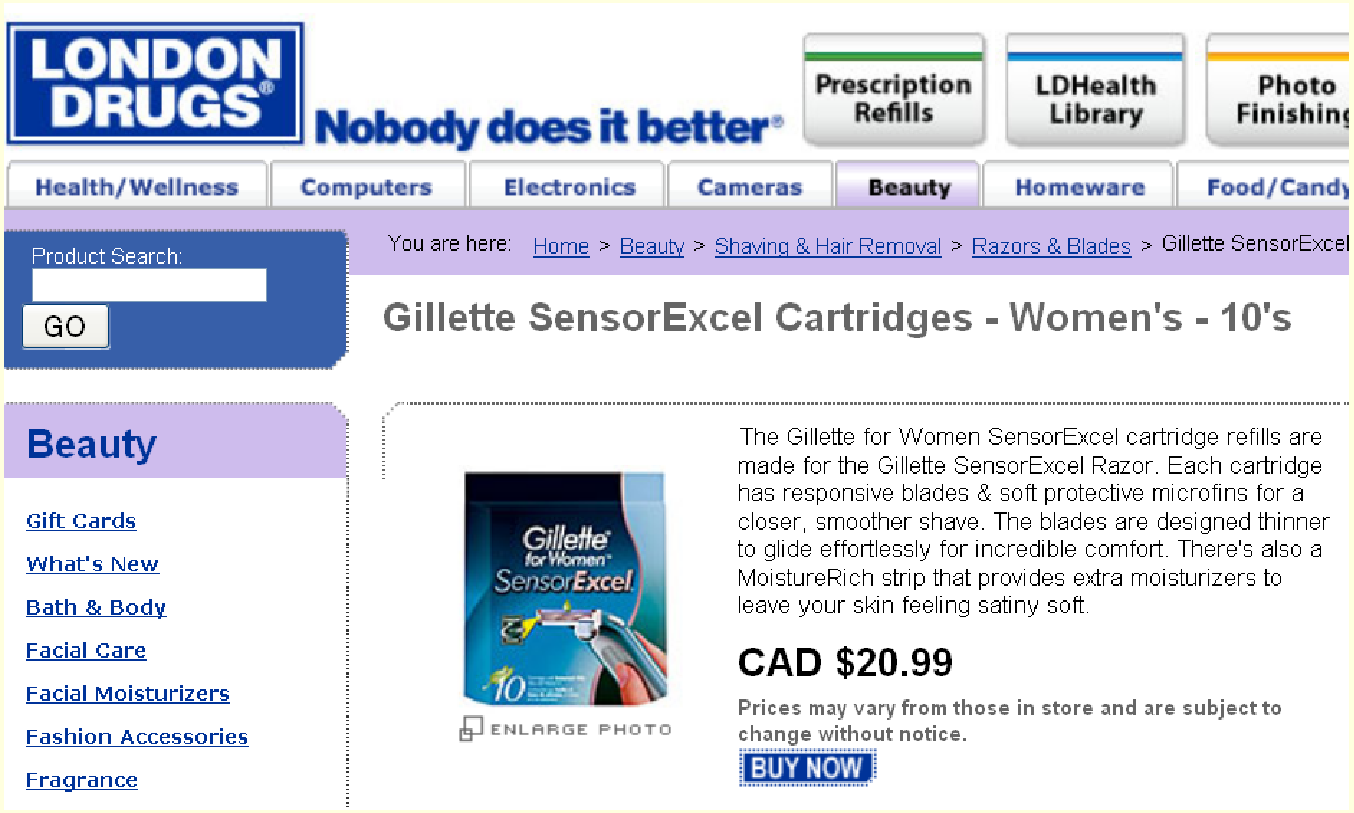
- RFM TR8100
- TI MSP430F148
  - 48 KB Flash
  - 2 KB RAM
- 916.5 MHz
  - 916.3-916.7
  - OOK on BPSK spreading
  - 9.6 kbps



www.rfm.com

Disposable Computing

- Devices $20 or less can be thrown away after a (possibly short) useful life, but still need code.



You are here: Home > Beauty > Shaving & Hair Removal > Razors & Blades > Gillette SensorExcel

Gillette SensorExcel Cartridges - Women's - 10's

The Gillette for Women SensorExcel cartridge refills are made for the Gillette SensorExcel Razor. Each cartridge has responsive blades & soft protective microfins for a closer, smoother shave. The blades are designed thinner to glide effortlessly for incredible comfort. There's also a MoistureRich strip that provides extra moisturizers to leave your skin feeling satiny soft.

CAD $20.99

Prices may vary from those in store and are subject to change without notice.

The Paradox

- Cheap devices require expensive development!
- One has to account for:
  - Code development cost.
  - Code reuse capabilities.
- Code production is the bottleneck to testing the great ideas found in the literature.
  - Simulation is a poor substitute.
- What helps development:
  - Sufficiently high-level abstractions (but limited OS).
  - A “natural” composition mechanism.

Development Challenges

- Abstraction & interface layer:
  - express elementary sensor behaviors, and
  - make them accessible to programmers.
- Multi-tiered code distribution tools & architecture:
  - hierarchical composition of sensing systems,
  - P2P composition of sensing/actuating systems.
- Integration of security mechanisms:
  - tension between security & limited capabilities.

Developmnent for Sensor Networks

- Currently No Code Reuse (most apps. one-off)
- Challenges:
  - Low capability platforms cannot afford the luxury of a full “commodity grade” OS; select the right abstractions that fit the platforms.
  - Little manufacturer interest in reusable software components outside those running on their own (sometimes proprietary) platform.

A Ratio to Remember

1000:1

Is this a New Problem?

- “Old” Distributed Computing Paradigm: development/programming for **large scale** decentralized computing BUT relatively reliable nodes, communication & adequate bandwidth (plus usually homogeneous nodes).

[Interest declined due to the re-emergence of centralized computing (as client-server model).]

Relation to Classic Control Systems

- Avoiding Sampling & Reporting at Nyquist Rate
  - Compressive Sampling
  - Perfect vs. Rate-Distorted Reconstruction
  - Fusion/Voting near the Phenomenon
  - Need for Good Models of the Phenomena
- Spatial Sampling $\leftrightarrow$ Sensor Deployment Density
- Close Coupling of Actuation to Sensing

A Word about Standardization

- Claim: “*Wireless Sensor Networks are not yet successful because the protocols have only recently been standardized, e.g., ZigBee.*”
- What we should be asking is:
  - *“Does the developer spend more time because of DL+PHY lack of standardization?”*
  - *“Does the developer's work become significantly more difficult when dealing with a proprietary DL+PHY vs. a standard one?”*
- Standardization at the higher layers is a struggle.
  - Some brave efforts from the Open Geospatial Consortium are reasons for hope, albeit “verbose”.

The TinyOS Story

- Admittedly the first serious attempt to provide an open-source OS for wireless sensor networks.
- Currently, a source of frustration for many developers. Value added products are not “free”.

- Model: event handlers and tasks
  - Event handlers cannot be preempted.
  - No task preemption as such (in “vanilla” TinyOS).
  - Tasks executed in order posted (in “vanilla” TinyOS).
  - No multi-threading as such.
  - Dynamic memory allocation curtailed.
  - “Wired” components useful but potentially hard to track down how overall functionality composed.

The PicOS Alternative

- Protocol designers can (ought to be able to!) describe protocols as finite state machines.
- A thread model allows for natural expression of concurrency across “independent” strands of logic.

- PicOS: an OS tuned to small platforms:
  - Implement concurrency as co-routines.
  - Co-routines reduce the stack requirements.
  - Express each process/thread as a FSM.
  - Process preemption possible at state boundaries.
  - Interrupts can preempt processes.
  - Interrupts deliver “events” to processes/threads.

```
process (sniffer, sess_t)
    char c;
    entry (RC_TRY)
        data->packet = tcv_rnp (RC_TRY, efd);
        data->length = tcv_left (data->packet);
    entry (RC_PASS)
        if (data->user != US_READY) {
            wait (&data->user, RC_PASS);
            delay (1000, RC_LOCKED);
            release;
        }
        ...
    entry (RC_LOCKED)
        ...
    entry (RC_ENP)
        tcv_endp (data->packet);
        signal (&data->packet);
        proceed (RC_TRY);
endprocess (1)
```

A Higher Level View

- Building the OS is fine, but what about some basic abstractions to help the developer? Which ones?
- "Solution": read current literature, find patterns and translate them to a handful of abstractions.
- Some first attempts:

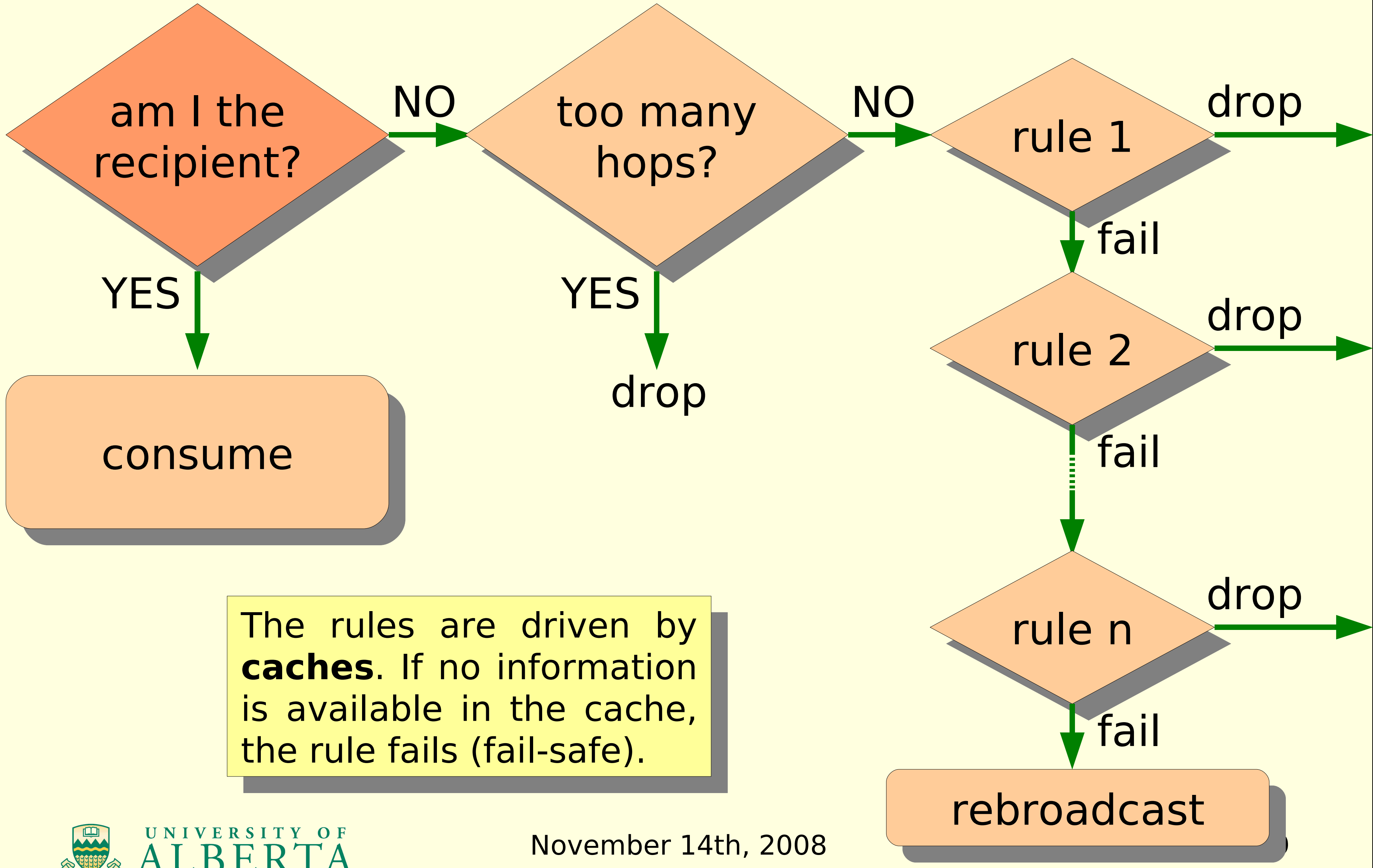
  The Emergence of Networking Abstractions and Techniques in TinyOS, by Levis et al. (NSDI 2004).

  Logical Neighborhoods : A Programming Abstraction for Wireless Sensor Networks, by Luca & Gian (DCOSS 2006).

Prime Candidates for Abstraction

- Path (route)
- Neighborhood
- Spanning structure
- Region
- Duty cycle
- …

How to Route in Wireless Sensors



The rules are driven by **caches**. If no information is available in the cache, the rule fails (fail-safe).

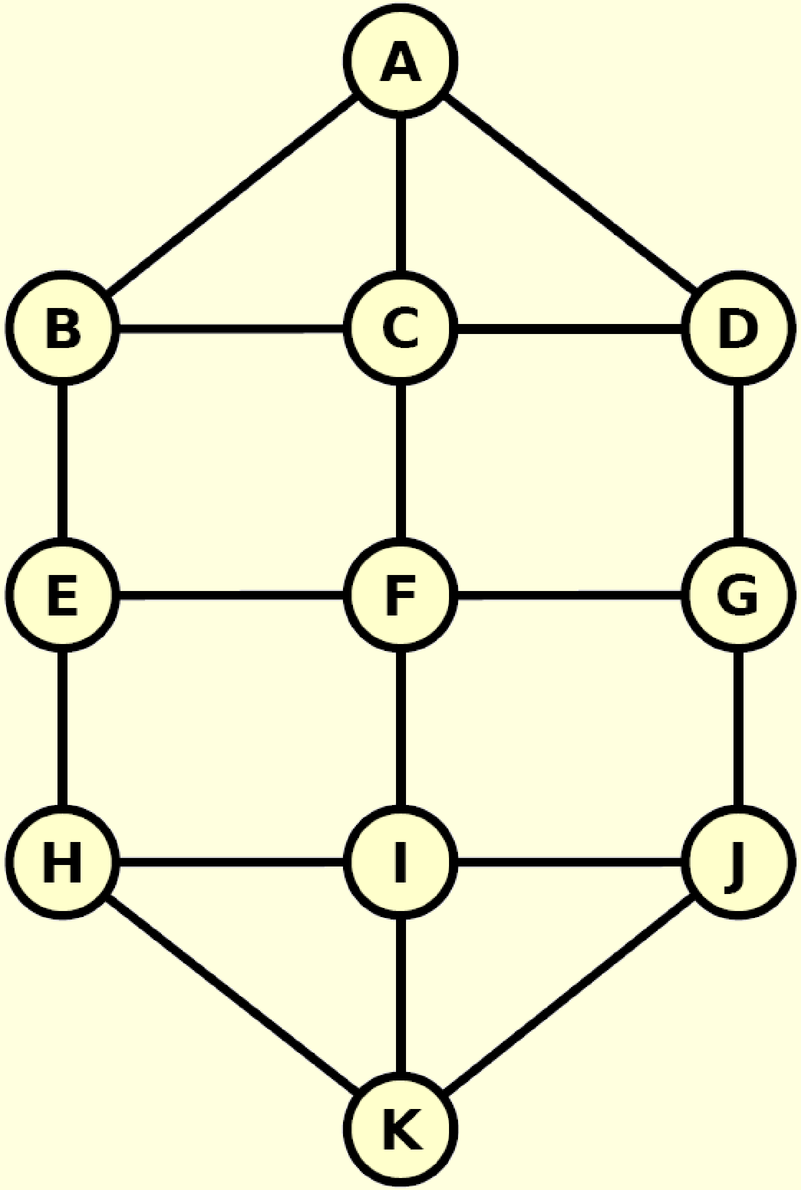
Spanning Structures

- (Mainly) Spanning Trees

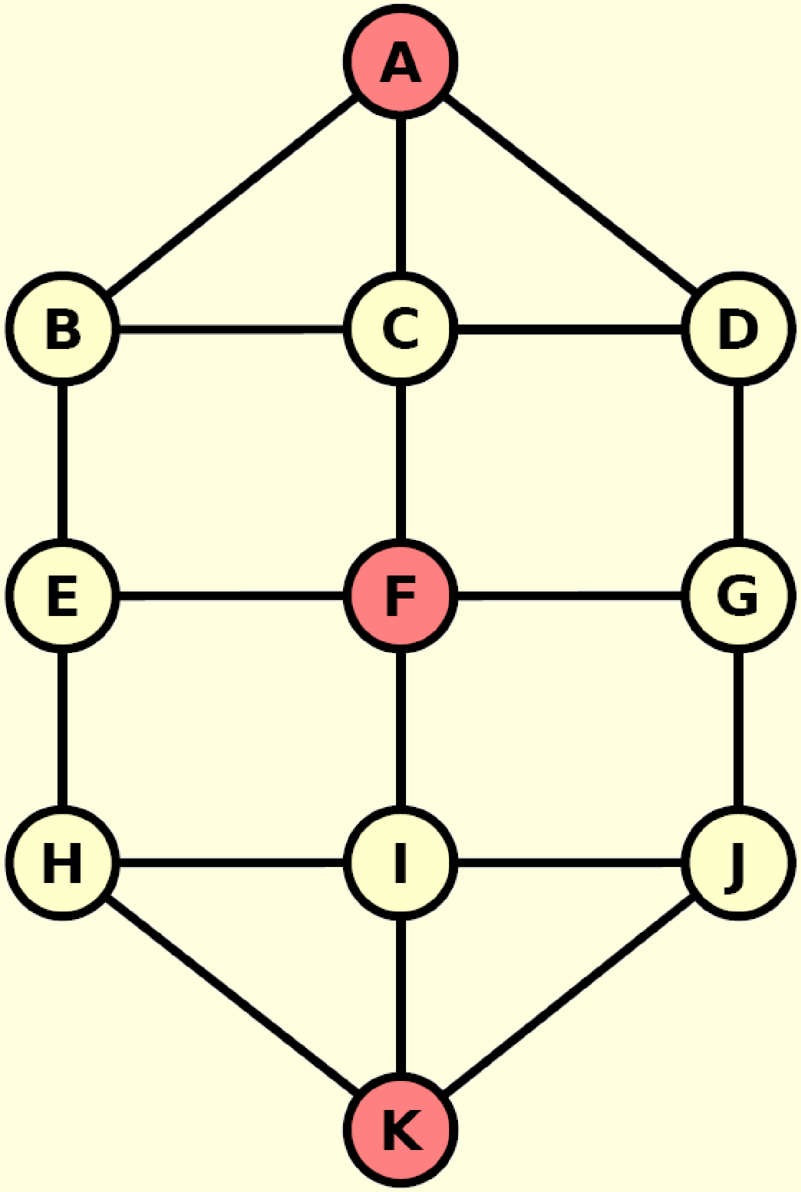
- Features:
  - All nodes reachable.
    - Basic ingredient for data collection and reporting.
  - Ordering via the parent/child relationship.
  - Logical separation of “interior” and “leaf” nodes.
  - Restricted forms possible.
    - For example spanning within a geographic area.
  - Spanning tree can be tuned to application needs.
    - What is the most useful spanning tree?

Clusters and Spanning Structure

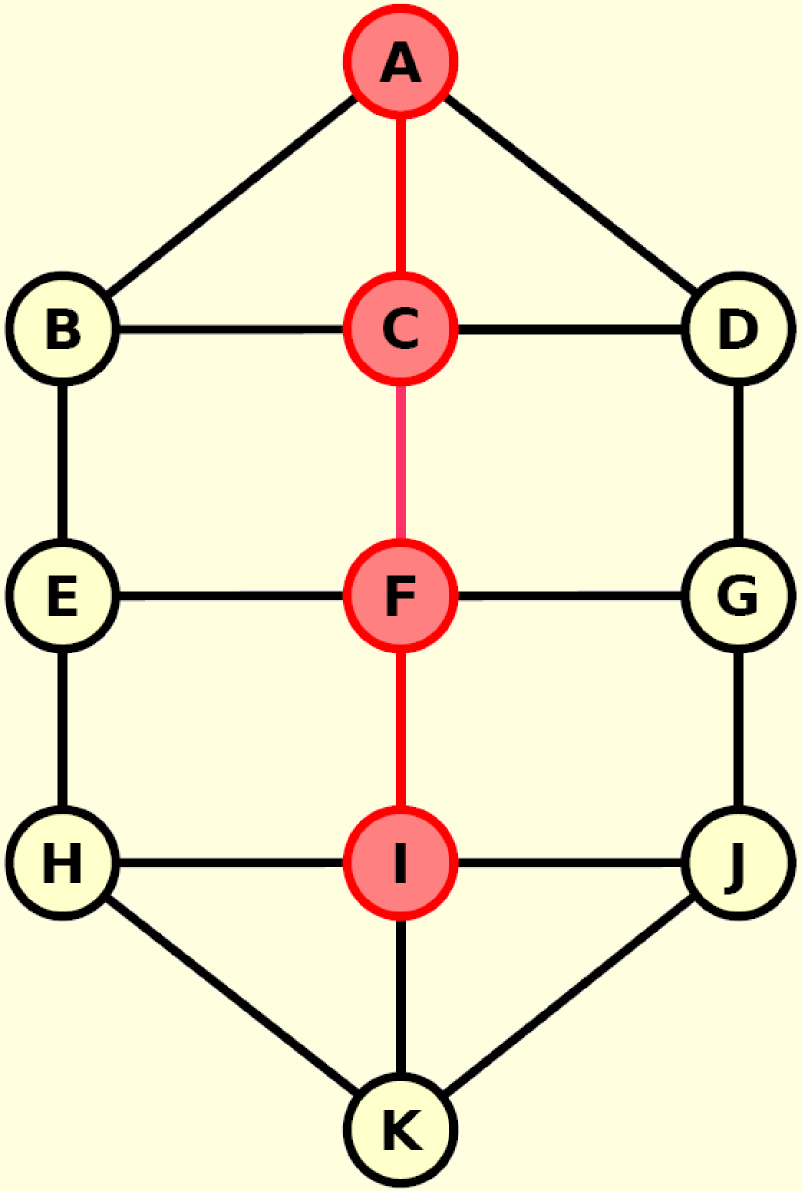
- Name the interior nodes *cluster-heads* and the leaf nodes as cluster *members*.
- The vast majority of cluster-based logical structures present in the literature provide **also** a strategy for **connecting** the clusters.
- Translate the what is a “good spanning structure” question to what is a “good clustering structure”.
- Hence, check out the (Connected) Dominating Set.



An example physical topology.



A Dominating Set.



A Connected Dominating Set.
(4 transmissions)

CDS Spanning Structures

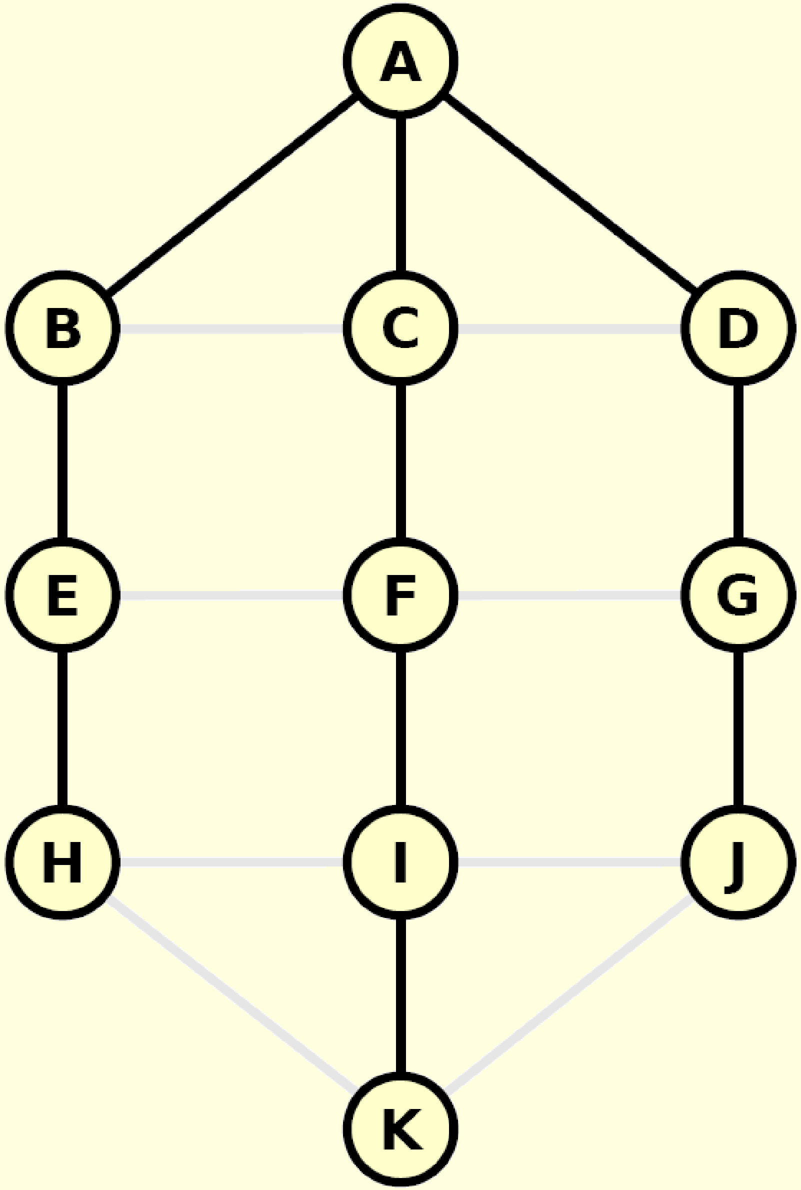
- We are interested in the Minimum CDS (MCDS)
  - It represents the minimum number of transmissions to “get to all nodes.”
  - An NP-hard problem (even on UDGs) with some known approximations including distributed ones.

- MCDS has received attention already:
  - Routing in mobile ad-hoc networks: OLSR
  - Multicasting in mobile ad-hoc networks, etc.
  - As the other side of the coin: “Leafy” Spanning Trees

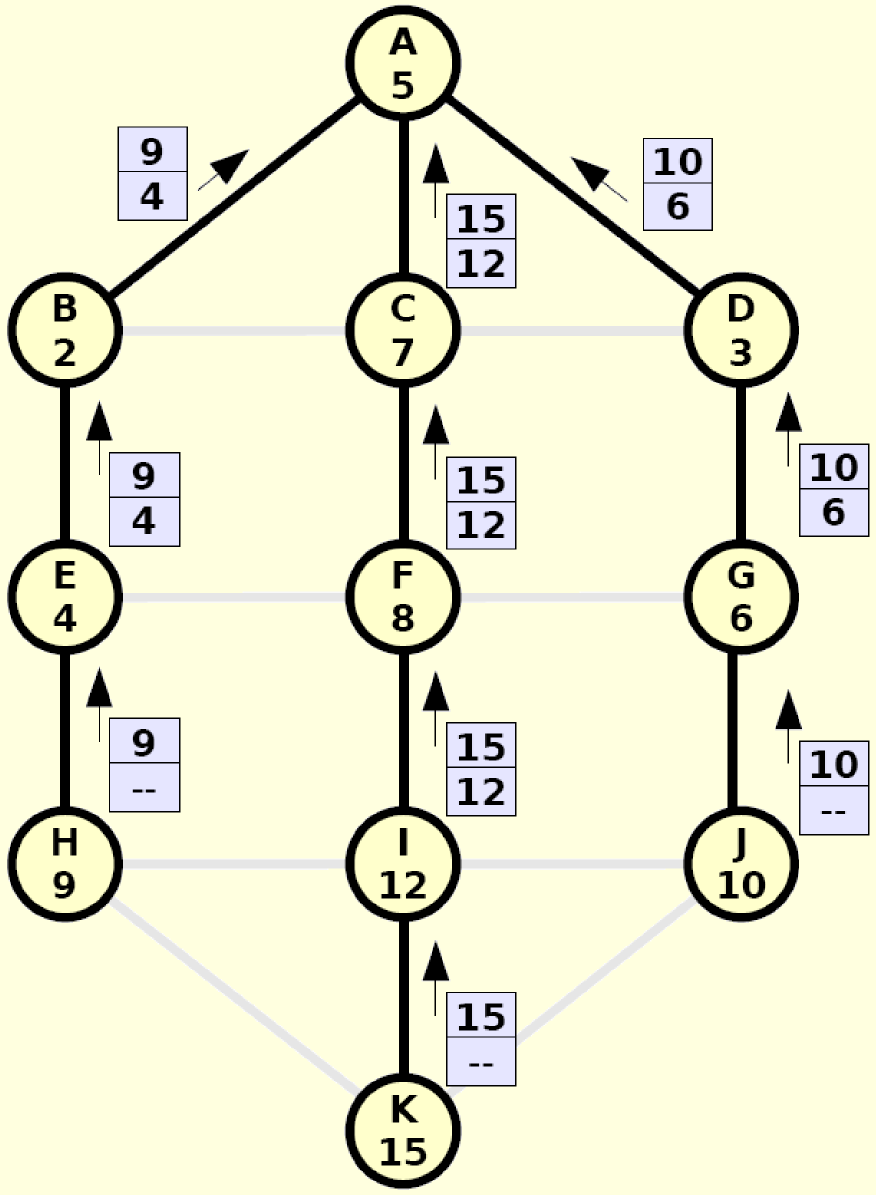
- Is MCDS a useful “universal” spanning structure?

An Example: Top-*k* Query

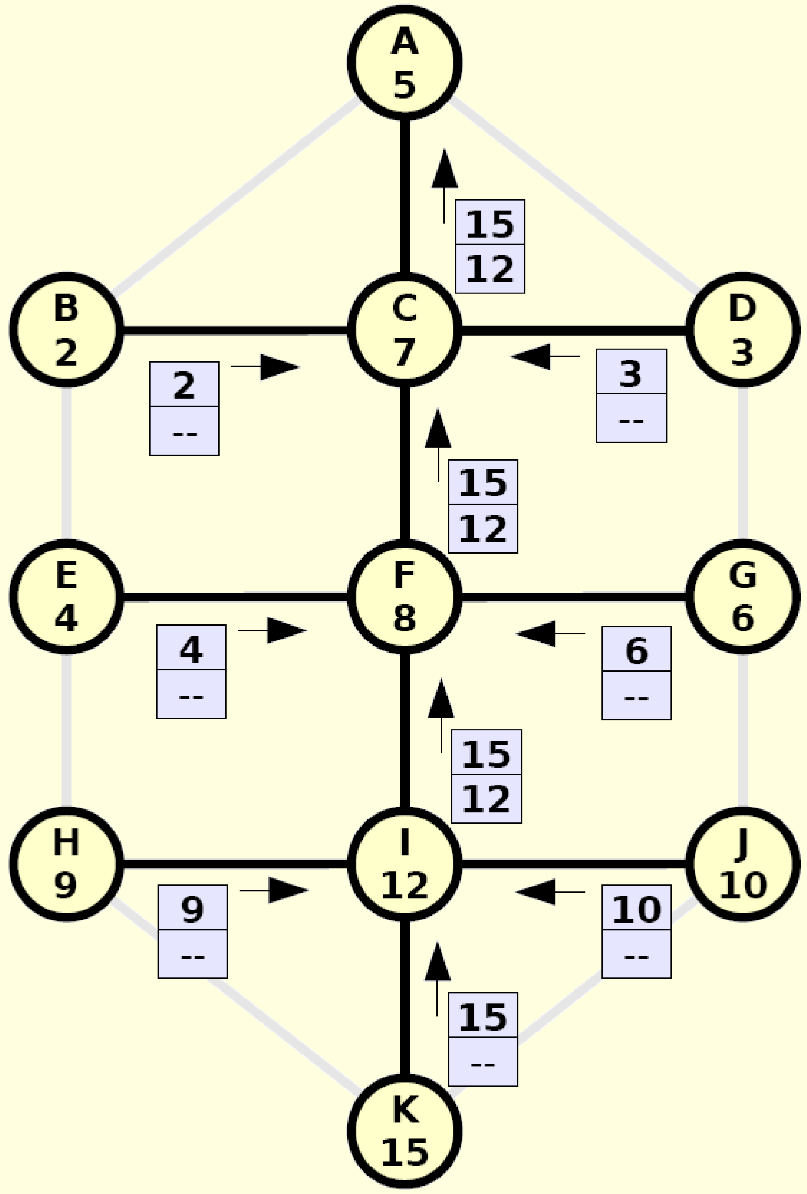
- Collect (periodically) information from sensors such that the top-*k* values can be determined.
- A Database Sensor Query Processing favorite.
  - Easily expresses min-*k*, max, and min.
- The literature calls for a spanning structure (root=sink) without caring about its characteristics.
  - Usually what is used is a minimum spanning tree, or a shortest path tree (SPT).
  - We will compare against a Dominating Set Tree (DST) constructed as approx.(MCDS)U{Root}



An SPT spanning tree.
(8 messages)



TAG [Madden at al., 2002] version of top-*2* on SPT. (17 message units)



TAG [Madden at al., 2002] version of top-*2* on CDS. (13 message units)

Lessons

- A spanning structure based on an underlying “clustering” (like MCDS) introduces sensitivity to the density and scale of the network that might not be a good match for “naive” algorithms if the cluster does not perform “aggressive” reduction of data volume.
- A good use of an MCDS-based tree is when it is known that the protocol has to employ broadcasts for “state update” (like the threshold).
- But broadcasts should be used sparingly, so a suggested match would be for algorithms where broadcasts are “amortizable” over longer time frames.

Instead of Conclusions: Pie-in-the-sky

- Provide (web) applications access to sensor data by transforming sensors to “first class citizens” of the cyber-infrastructure (à la SOA).
- Technical Issues: reliability, security, routing, energy consumption, hostile deployment environments, long-term maintenance.
- Challenges:
  - abstractions (expressions of elementary protocols) & interface layer to express elementary sensor behaviors to make them accessible to programmers, ...
  - ... and we did not even touch the need for multi-tiered code distribution tools & architecture

Thanks

NSERC
CRSNG

Alberta Advanced Education and Technology

olsonet