CREATE-NET
via Solteri 38, Trento
Italy

“Pervasive communication environments: the BIONETS perspective - 1st part”

F. De Pellegrini
fdepellegrini@create-net.org

Autonomous Computing
in Smart Environments

University of Fribourg, 23 - 24 November 2006

Wireless sensor networks

- *“Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances.”* [Ak]

- Sensed phenomena with a wide spectra:
  - temperature, humidity, pressure
  - mechanical stress
  - position speed and direction of an object
  - . . .

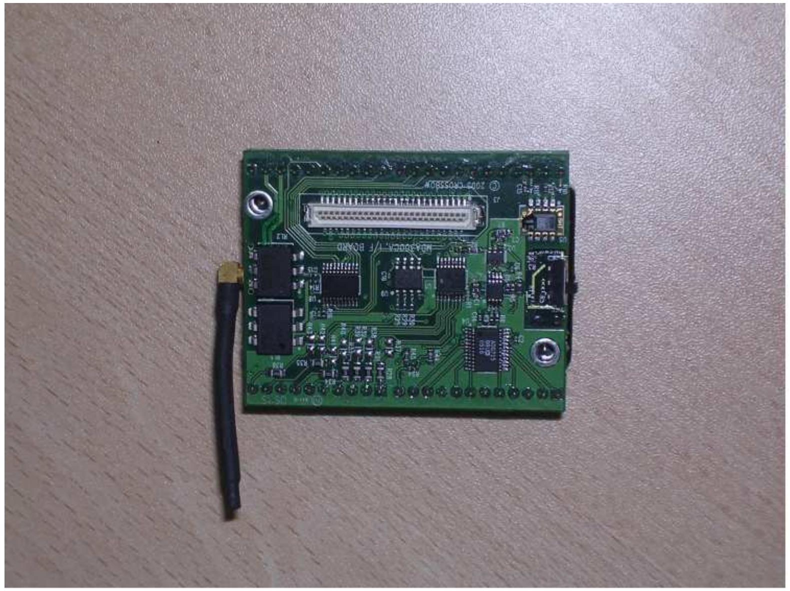
- Advantage of WSNs: sampling locally a field, with spatial resolution depending on the sensor nodes density

- Temporal resolution: each sensor can report a time series of data

Wireless sensor networks

- Applications: environmental monitoring, military (surveillance, tracking), health monitoring, home automation ...
- Typical features required to sensors:
    - cheap: < 1 \$ per sensor
    - extremely low powered
    - the smaller the better
    - support very dense deployment
    - ease of deployment (often called self-organization)
- Once deployed, maintainance should be minimal
- If cost permits, better replace/add new units
- Power consumption: sensing, processing and communications.

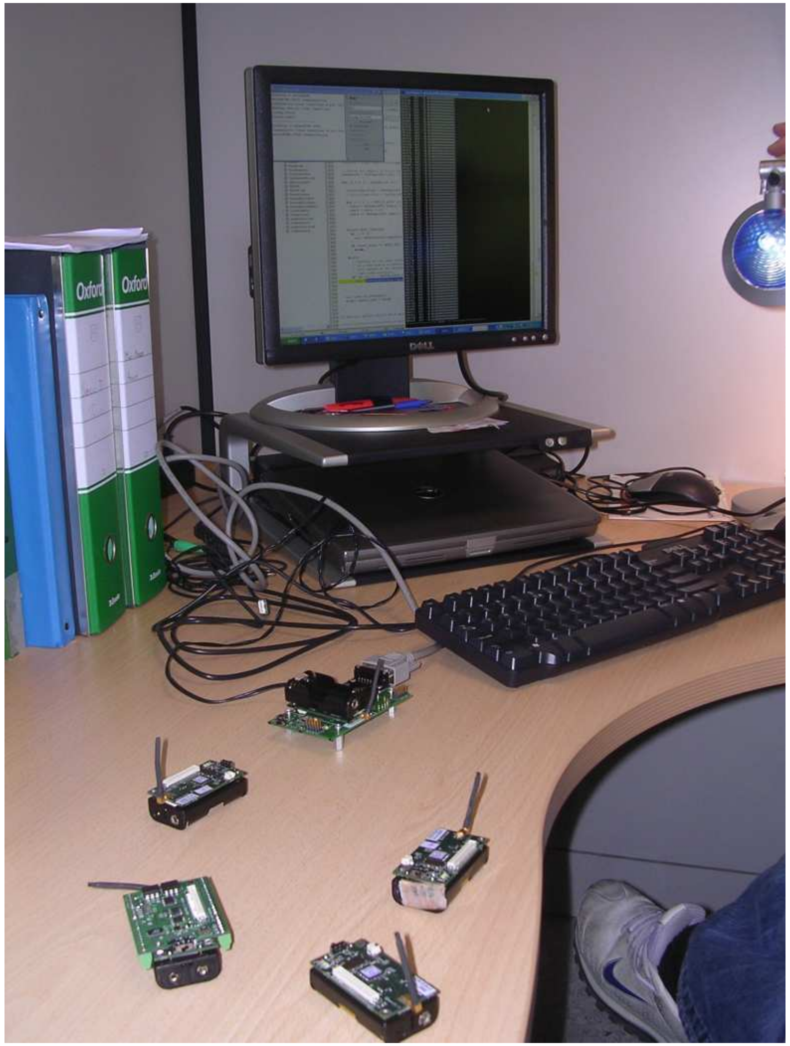
Sensors nodes - Building Health Monitoring



courtesy of Ing. David Tacconi

- Application: Torre Aquila - Trento, historical building
- Remote server collecting data on: temperature, pressure and misallignement of building structure
- Sensor node: CrossBow Mica Mote
- Sensing board: xy accelometer, microphone, termometer
- Node mounting full OS: Tiny-OS UC Berkely
    - Radio messaging
    - Multi-hopping from mote to mote
    - Low power modes
    - Sensor measurements and signal processing of external peripherals

Sensor nodes - Building Health Monitoring



courtesy of David Tacconi

- Development: nesC on the PC
- Base station: allows the aggregation of sensor network data onto a PC
- Base station acts as a sink
- Accelometer shoots when perturbation occurs
- Avoid wasting resources for *"normal conditions"*: run-length encoded data

Directed diffusion

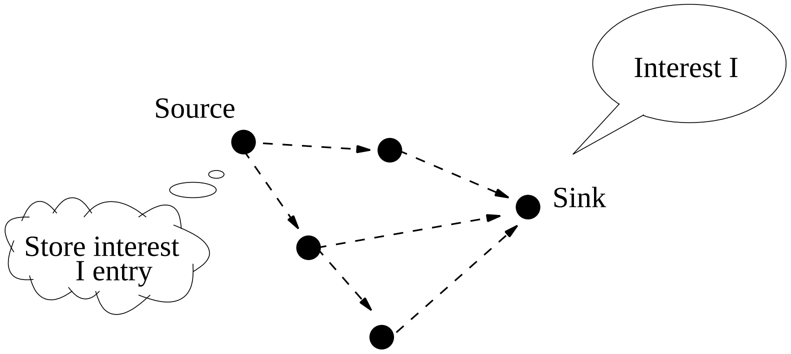
- Sensor network achitecture is *data centric*
- Directed diffusion [Es1,Es2]: reference paradigm
- Sensors execute certain tasks, referenced according to predefined attributes
- Elements of directed diffusion [Es1]: interests, names, data messages, gradients and reinforcements
- Iterest messages generated at *sinks*:

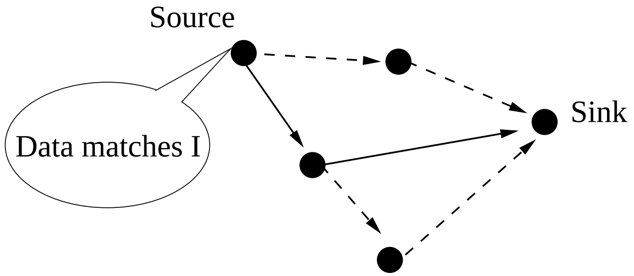
```
type = wheeled vehicle // detect vehicle location
interval = 20 ms // send events every 20 ms
duration = 10 seconds // for the next 10 seconds
rect = [-100, 100, 200, 400] // from sensors within rectangle
```

- Data messages are generated by sensors responding to interests:

```
type = wheeled vehicle // type of vehicle seen
instance = truck // instance of this type
location = [125, 220] // node location
intensity = 0.6 // signal amplitude measure
confidence = 0.85 // confidence in the match
timestamp = 01:20:40 // event generation time
```

Directed diffusion





- No source-destination pairs as in IP
- Kind of *semantic addressing* matching the application
- Sink node express *interest* propagating a query (initial flooding)
- Initial gradient setup is exploratory, i.e. low update rate requested
- Any potential source node stores the entry in a local cache
- +/- Reinforcement of routes
- Loop avoidance

Conveying data

- Conveying data across the network becomes the first step in order to make them available at the sink: we would like to make it *“efficiently”*
- From the application down to the bit transmission, collecting data in pervasive/sensorized environments: Where is the bottleneck?
- What we do want from the network architecture is
    - No need for fixed topology
    - Mobile units do take part to the network topology
    - No wiring
    - Easy replacement/maintainance
- Also, we would like to be free to add/replace/switch-off as many nodes as we want with no need for reconfiguration
- But, packing all this nice features altogheter usually has a price ... efficiency
- Not enough: devices should last the longest the possible, i.e., make a parsimonious use of batteries.

Routing

- Traditional techniques for ad-hoc networks
  - Flooding: robust and fast but expensive; MAC problems with broadcast (ex. 802.11)
  - Proactive Routing Techniques: determine routes independent of traffic pattern. Destination-Sequenced Distance-Vector (DSDV)(Bellman-Ford)
    - nodes maintain each destination in network,
    - control traffic
  - Reactive: determine routes when needed. Dynamic Source Routin (DSR), Ad-Hoc on demand Distance Vector routing (AODV)
    - source node broadcasts route request packets RREQs
    - destination and intermediate node with reply RREP packets
    - route metrics: hops, link quality ...
- Not suitable for sensor networs: energy constraints the applicability of route maintainance, topology simply "aslept"
- Directed diffusion: data centric paradigms [KamSurvey]

MAC solutions

- TDMA/FDMA: requires syncronization and scheduling, collision free
- Random Access (ALOHA): slot-less or slotted
- Carrier sense multiple access (CSMA): slotted or slot-less
- Polling: master/slave

Problem: idle listening consumes a lot of energy

Solutions: sleep as much as possible
Nodes do not sleep when they carrier sense: refined techniques

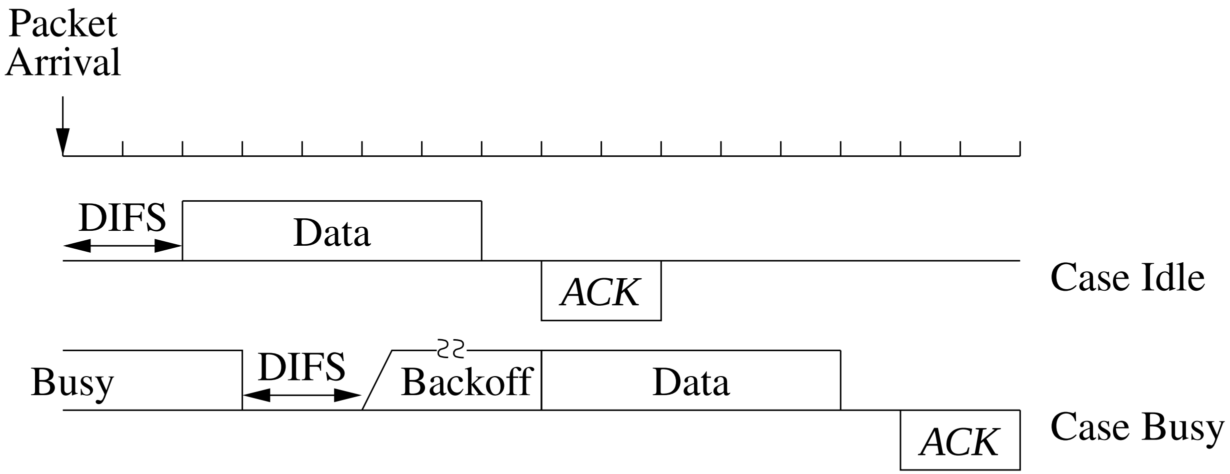
- TynyOS S-MAC: periodic listen and sleep, fixed duty cycle, need to synchronize neighboring nodes schedules
- TynyOS T-MAC: adaptive duty-cycle
- TinyOS B-MAC: wake up and check channel status, send long preambles

Off the shelf MAC: CSMA/CA

- The “ancient” standard for *ad-hoc* networks is represented by the IEEE802.11x family
- The reason for its success: plug-and-play, ad-hoc mode and cheap hardware implementation
- The basic mechanism for medium access control is the well known CSMA/CA protocol
- Basically the *de-facto standard* of wireless technologies
- No wonder: fashionable also IEEE802.15.4 - Zigbee used for Mica Motes
- What is going to happen for large size deployments under CSMA/CA, does it scale?
- Known results under *saturation* conditions

CSMA/CA

- Mechanism: a node listens to the medium (CS: Carrier Sense)
- When the medium is perceived idle for longer than a DIFS: transmit
- If the medium is busy: freeze the counter and decrement after one idle slot
- If no ACK is coming back: either error or collision, retry
- Retry up to a maximum number of attempts



Limit performances of CSMA/CA

- Bianchi’s model or also fixed point analysis duable in saturation [Bianchi,KumMior]
- The system is bounded by the number of nodes
- Listening to the medium and to freezing the backoff counter has a cost: delay
- Basically: service time is increased at each alien tranmission
- Even worse: collisions and retransmissions are overheard due to the protocol
- Closed form analysis for the service time consumption of the system [ZaDe1]

$$w_s = T_s + \sum_{j=1}^{n_c} T_{c,j} + \tau_B$$

- Channel sensing becomes a dramatic waste of energy [ZaDe2]

Service time of CSMA/CA

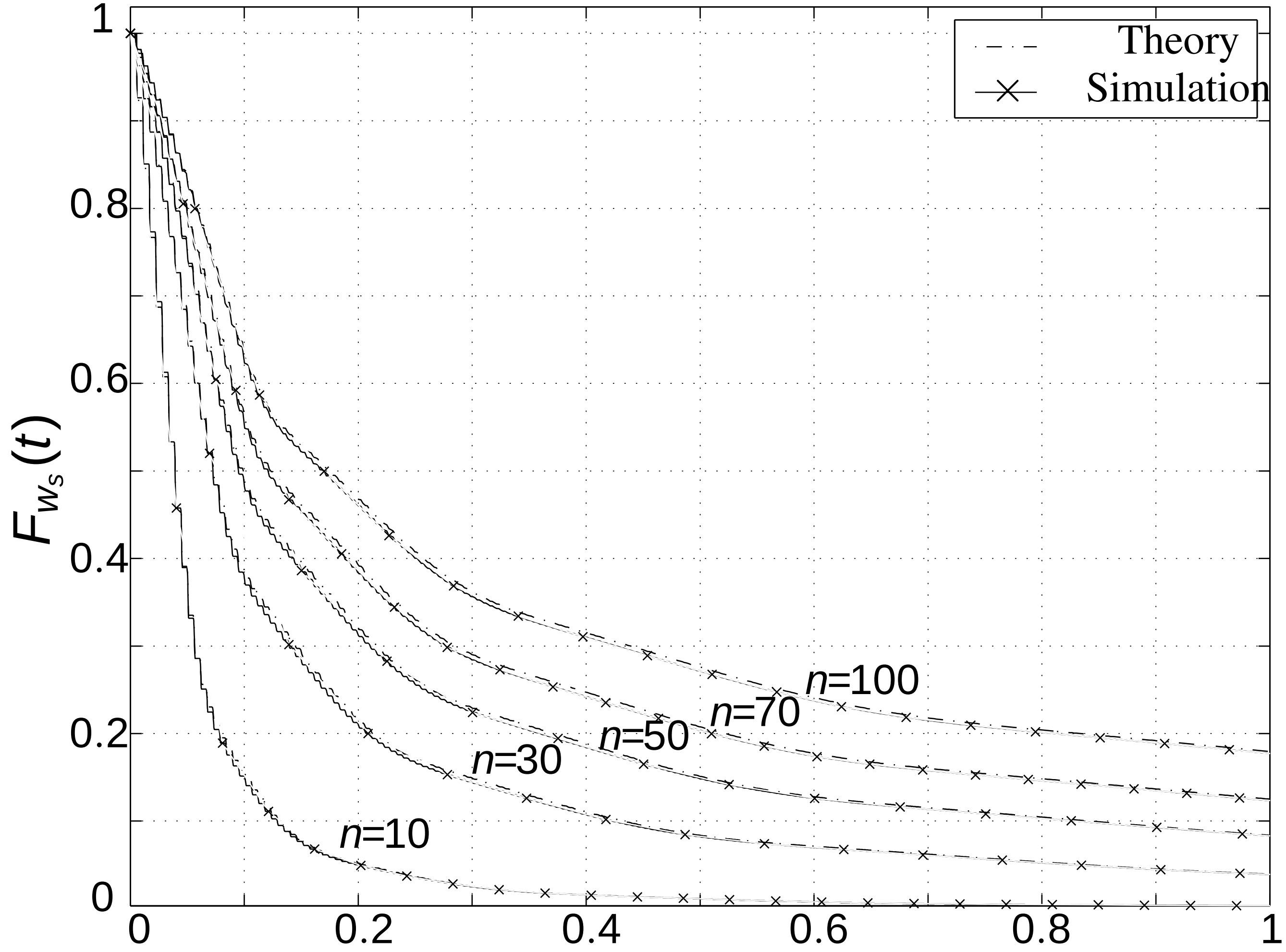


Figure 1: Cumulative Distribution Function of the service time w_s ($n = 10, 30, 50, 70, 100$, packet–length of 600 bytes.

Service time of CSMA/CA

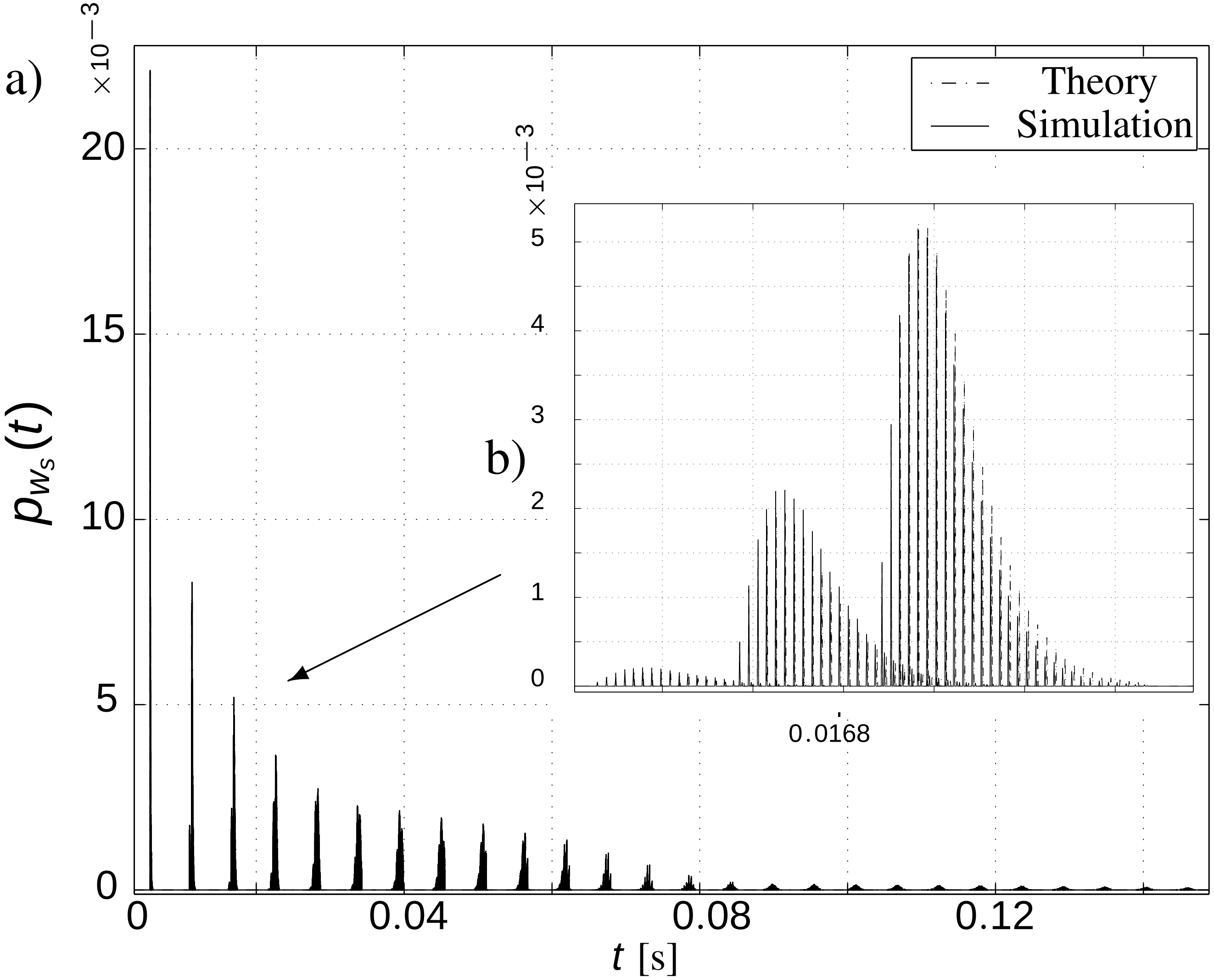


Figure 2: a) Mass distribution function b) Detail of the third spike.

Idle mode impact on energy consumption

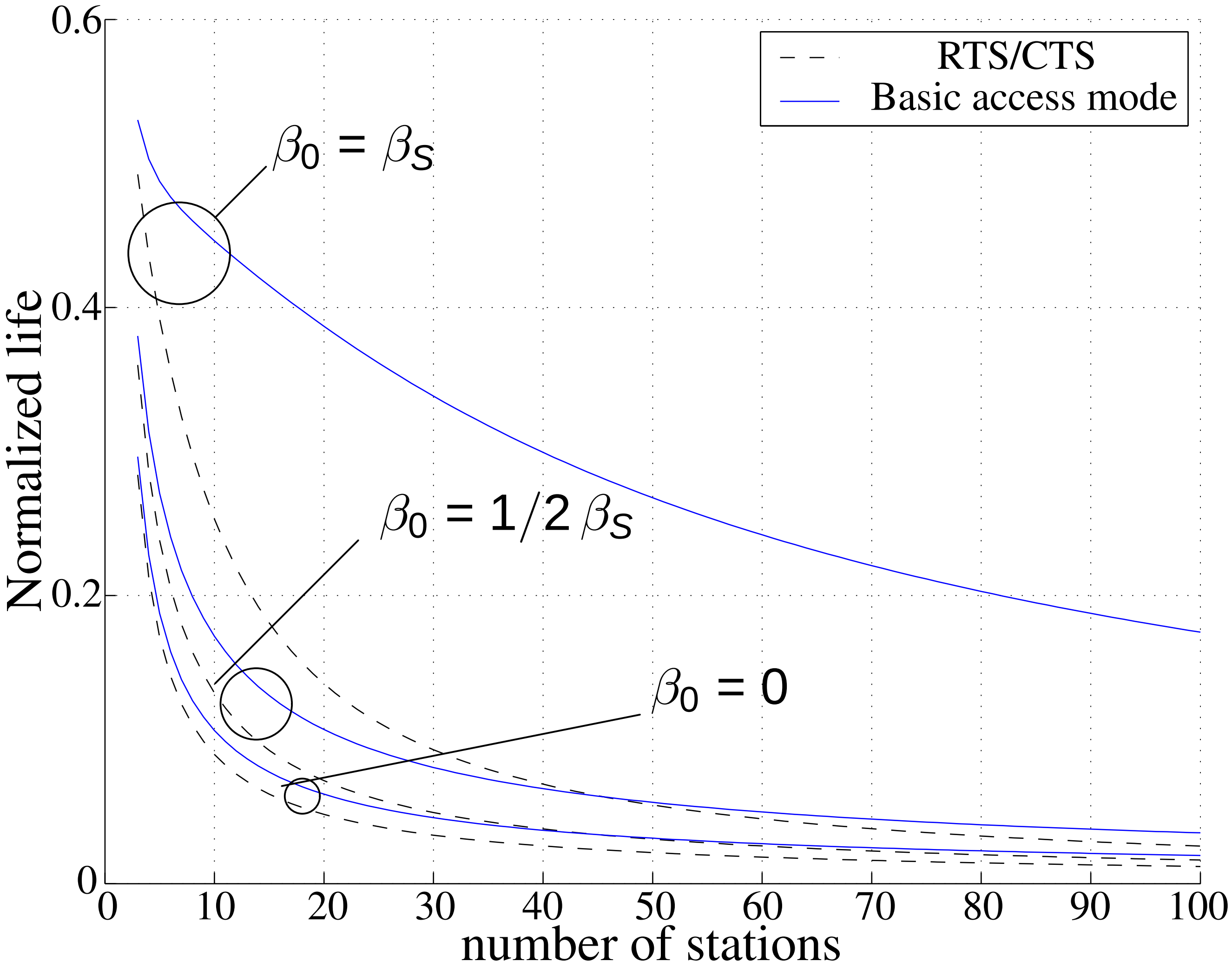
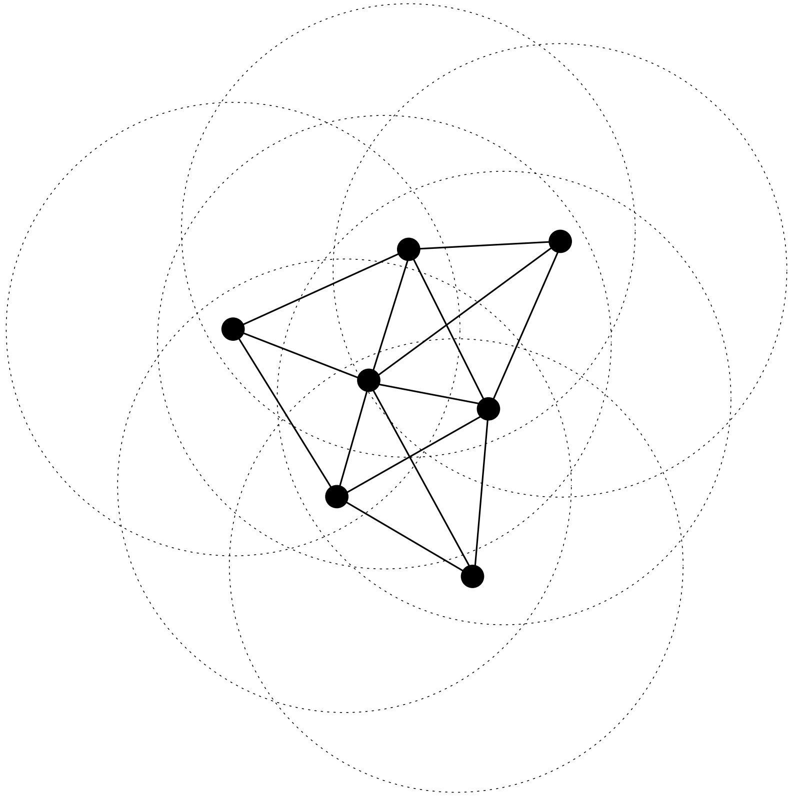


Figure 3: Normalized Life of stations: basic access mode (solid line) and RTS/CTS access mode (dashed line), $\beta_0 = 0$, $\beta_0 = 1/2\beta_S$ and $\beta_0 = \beta_S$.

Ad-hoc Networks

Wireless Ad-hoc networks



- The most flexible network architecture we can think about: *ad-hoc* means shaped to obtain a particular aim.
- Example: deploying a wireless sensor network would be made much easier made [SmartDust] style, the configuration of the network being dictated by the landscape peculiarities
- What kind of network would one obtain?
- There exist strict *constraints on the scalability* of ad-hoc networks
- Several studies on the capacity and connectivity of ad-hoc networks since 2000 cornerstone work [GuptaKumar]

Static ad-hoc networks

- Several connectivity models
- Simplistic view: two nodes communicate if they are within radio range (Boolean model)
- But RF connectivity is subject to interference, links actually depend on every other transmission
- Success depends on interferers: if node i transmits to node j
- Physical model: $SNIR(i,j) = \frac{P_i/|X_i - X_j|^\alpha}{N_0 + \gamma \sum_{k \neq i} P_k/|X_i - X_k|^\alpha} > \beta$
- Protocol model: $|X_j - X_j| < (1 + \Delta)|X_j - X_k|$, for any other node k

For both models Gupta and Kumar proved that the order of capacity for random networks unitary sphere/disk is

$$\lambda(n) = \Theta(1/\sqrt{n \log n})$$

Static ad-hoc networks

- Consequences: it seems we cannot design our favorite “pervasive” ad-hoc networks disregarding scalability issues
- Even with *“perfect scheduling algorithm which knows the locations of all nodes and all traffic demandes, and coordinates wireless transmissions temporally and spatially to avoid collisions which would otherwise result in lost packets”*
- Proof in [GuptaKumar]: upper and lower bound
- Three factors into play (protocol model):
  - length of routes: we have to traverse several relays
  - number of nodes
  - area “occupied” by useful transmissions

Possible solution is *locality* as suggested in [GuptaKumar]: sources communicating only to nearby nodes (smart homes)

Static ad-hoc networks

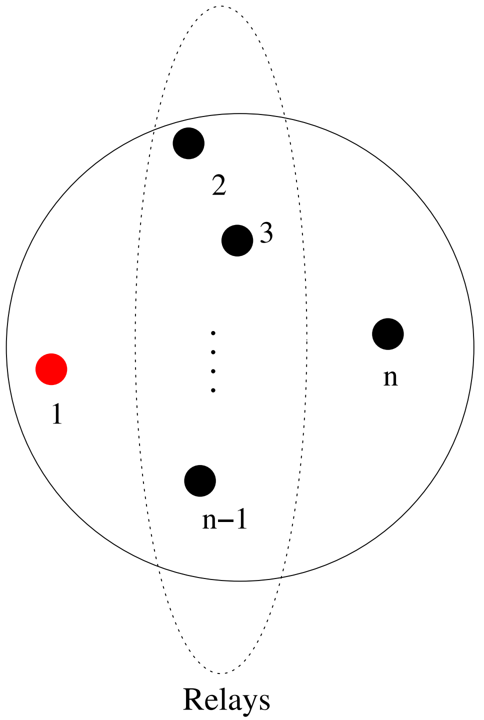
- Later result [DT]: connectivity and the capacity depends on the attenuation function $l(|X_i - X_j|)$
- Everything can be expressed in terms of the interference ratio

$$SNIR(i,j) = \frac{P_i l(|X_i - X_j|)}{N_0 + \gamma \sum_{k \neq i} P_k l(|X_i - X_j|)} > \beta$$

- Physical reasons force the attenuation function to be bounded $l(x) < M$
- The trade off is between capacity (β) and connectivity (γ)
- Dousse and Thiran: if the attenuation function is positive and bounded, then the transport capacity is bounded above by a constant, independently of the number of nodes
- Uniform traffic matrix: $O(1/n)$

Effect of Relays

1st hope: Gastpar and Vetterli [GV]: coding can improve ad-hoc network capacity



- One source-destination pair
- The remaining $n-2$ nodes act as relays
- Relays cooperate with source and destination in order to make them communicate
- Allow for arbitrary complex coding schemes
- Half duplex scheme: source transmits every two time slots
- Signal model as in [GK], Gaussian relay network, power of noise N
- Power of the source $\leq P$, (cumulative) power of the relays $\leq (n-2)P$

Effect of Relays

The result holds asymptotically: upper and lower bound match as n diverges

- Upper bound: broadcast channel bounds $C_{upper} \leq \frac{1}{4}\log_2(1 + (\sum_{k=2}^{n} \alpha_k^2)P/N)$
- No relay closer than $r_0 > 0$ to the source: then fading coefficients α_k are bounded under suitable assumptions
- Lower bound: constructive scheme;
  - At $t + 1$ relays repeat signal received from source in slot t, scaling down power accordingly
  - Destination decodes: $\hat{X}_n(t+1) = \gamma_1 Y_n(t) + \gamma_2 Y_n(t+1)$
  - Finally: $C_{lower} \geq \frac{1}{4}\log_2(P/D_1)$, $D_1 = \frac{PN}{N+Pf}$
- [Gastpar - Vetterli] The capacity $C_{lower} \leq C \leq C_{upper}$, upper and lower bound converge and C is in the form:

$$\log(1 + ||\alpha||P/N)$$

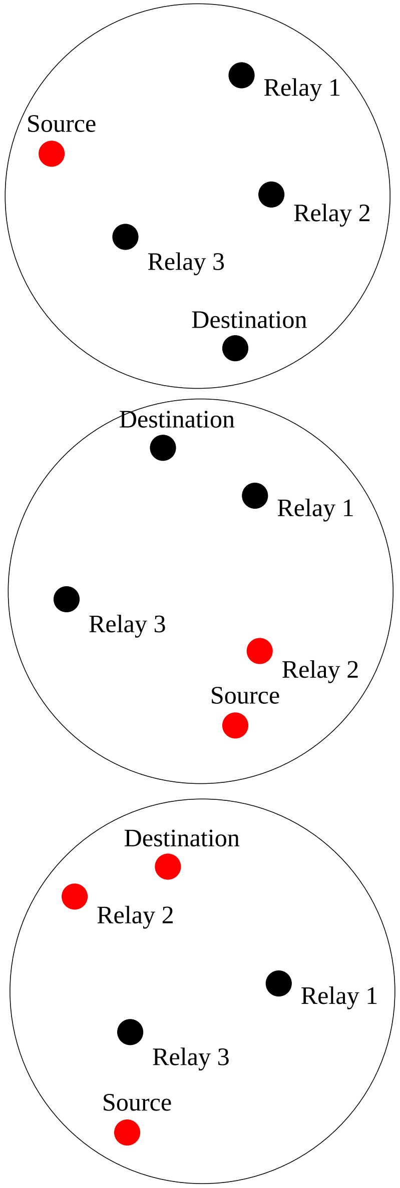
The impact of mobility

- 2nd hope: work from Glossglauser and Tse [2002]
- Search for a mechanism that *does not degrade with the number of nodes.*
- Technical assumptions:
  - n nodes in the unit circle of unitary area
  - fixed pairs Source-Destination
  - nodes follow i.i.d trajectories
  - node positions are uniform over the disk at each point in time (this does not scale with the number of nodes)
  - slotted time model
  - Success: $SNIR(i,j) = \frac{P_i \gamma_{ij}}{N_0 + (1/L) \sum_{k \neq i} P_k \gamma_{kj}} > \beta$
  - $\gamma_{ij} = |X_i - X_j|^{-\alpha}, \ \alpha > 2$

The impact of mobility

- From the static case: typical length of S-D paths is $\sqrt{n}$
- Relaying is the bottleneck? Let us try to transmit directly.
- But, it does not scale: in [GT] the maximum number of successful allowed transmissions per slot is $\Theta(1)$.
- This is bad: throughput per session is smaller than $n^{-1/(1+\alpha/2)}$ and $\alpha > 2$. ($o(\sqrt{n})$)
- Hence *some* relaying is due.
- Key intuition: relay the less the possible.
- How less?

The impact of mobility



The *two-hop relaying protocol* proposed in [GT]:

- Step 1: source node has the packet
- Step 2: first hop, source node handes off the packet to a relay neighbor
- Step 3: second hop, the relay neighbor deliver the packet to the destination

This scheme *under suitable assumptions* provides a *constant* long term source destination throughput, i.e. Glossgauser and Tse proved that using mobility in wise manner, one can make per session throughput a constant in the number of nodes $\Theta(1)$.

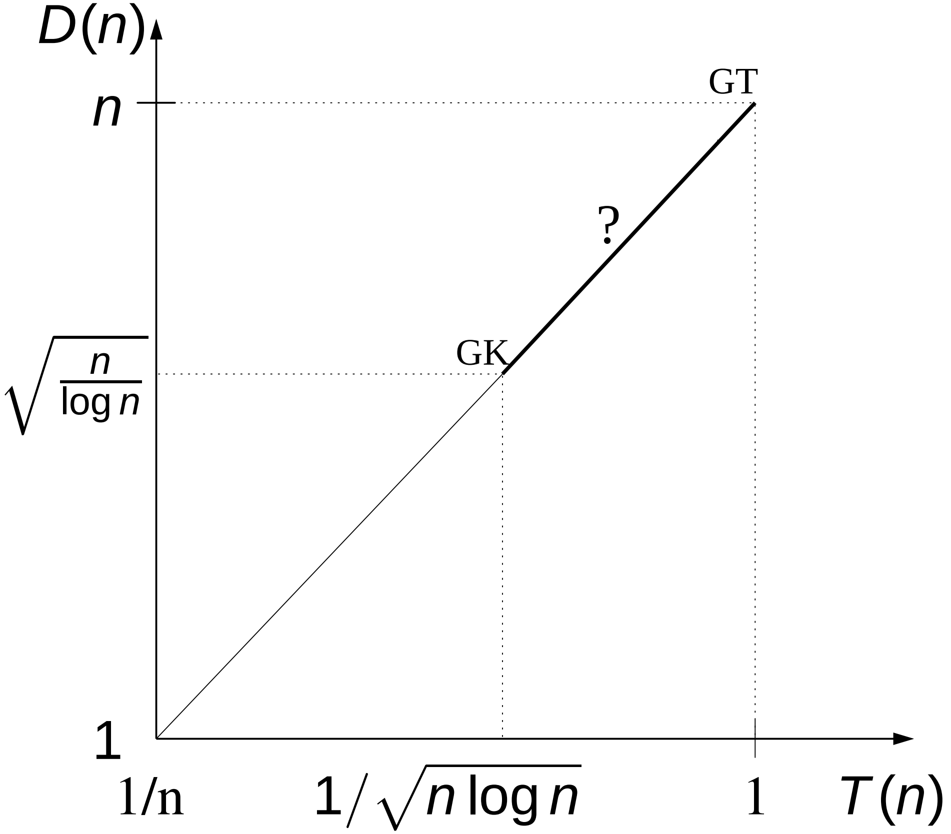
The impact of mobility

- Constructive proof in [GT] pivoted on the 2-hop relay protocol
- The protocol works picking a sender density $\theta \in (0, 1)$
- Pick randomly a set of n_S sources out of n nodes, $n_S = \theta n$
- The remaining $n_R = n - n_S$ are designated receivers
- Keep only N_t sources such that interference permit transmission
- Implement half duplex (odd and even time slots)
- The 2-hop protocol is able to sustain $\Theta(n)$ feasible source-destination pairs N_t (!)
- Due to symmetry: probability that feasible S-D pair i and j is picked is $\Theta(1/n)$
- Throughput per feasible pair: $\Theta(1/n)$
- Sum up on the $n - 1$ possible relay “paths” the result in follows

[Glossglauser-Tse] The two-phased algorithm achieves a throughput per S-D pair of $\Theta(1)$, i.e. there exist a constant $c > 0$ such that

$$\lim_{n\to\infty} Pr\{\lambda(n) = cR \text{ is feasible}\} = 1$$

The impact of mobility



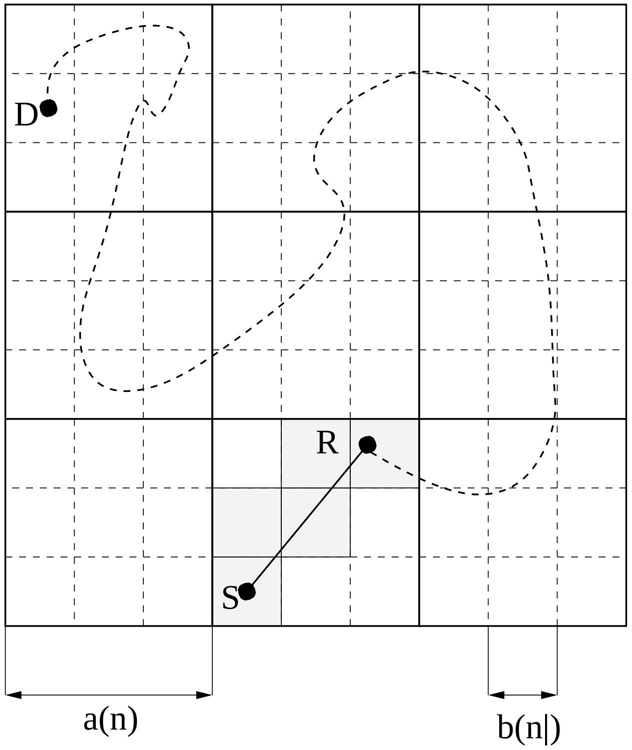
- Mobility enhances capacity: at which cost?
- In [GT] already pointed out: delay would diverge.
- El Gamal et. al. [ElGa] in 2004: formalization of the trade-off between throughput and delay
- Three schemes on the unit torus
- Static ad-hoc (Scheme 1):

$$\downarrow D(n) = \Theta(\sqrt{n/\log n}) \quad T(n) = \Theta(1/\sqrt{n \log n}) \downarrow$$

- Pure mobility (Scheme 2):

$$\uparrow D(n) = O(n) \quad T(n) = \Theta(1) \uparrow$$

The impact of mobility



- Scheme for covering the intermediate case
- Divide the unit torus into $a(n)$ cells; each cell divided into $b(n) = \Theta(\log n/n)$ subcells
- Cellular TDMA scheme proposed in [ElGa]: $\Theta(na(n))$ packet time slots per cell slot (speed and packet length shrink with n)
- Split each packet time slot into two subslots
- Subslot A: send to destination if in the same cell, otherwise, send along cells along segment SR to a relay picked randomly
- Subslot B: each node picks another node at random in the same cell and send a packet destined to it

[El Gamal et al.] If $v(n) = o(\sqrt{\log n/n})$ then the scheme

$$T(n) = \Theta(1/\sqrt{na(n)\log n}), \quad D(n) = O(1/a^{1/2}(n)v(n))$$

where $a(n) = O(1)$ and $a(n) = \Gamma(\log n/n)$

Connectivity issues

- Deep result: connectivity and capacity are strictly related in ad-hoc networks [DFT]
- Percolation theory: $B_n = [o, \sqrt{n}]^2$, extendend model, Poisson point process
- *Full connectivity* within B_n has a cost: radius has to increase as $\log n$
- Conversely, for every positive fraction of nodes $0 < \theta < 1$, there exist $r > 0$ such that a connected cluster of nodes contains a fraction θ of the nodes

Discarding part of the nodes things get better:

[Dousse, Franceschetti, Thiran] (LB) For any $0 < \theta < 1$, there exist a rate $R > 0$ independent of n, such that there exist a subset of the nodes of size $n\theta$ in which each node can send data to any other node at rate R w.h.p.

[Dousse, Franceschetti, Thiran](UB) For any rate $R > 0$, the fraction of nodes that can send data to any destination at that rate is at most θ w.h.p., where $\theta = \mathbb{P}[I \geq \frac{N_0}{P}(2^{2R} - 1)]$, where I is the shot noise defined by $I = \sum_{x \in N}$ l(||x||), and N is a Poisson process with unit density ($\theta < 1$).

Processing and communications

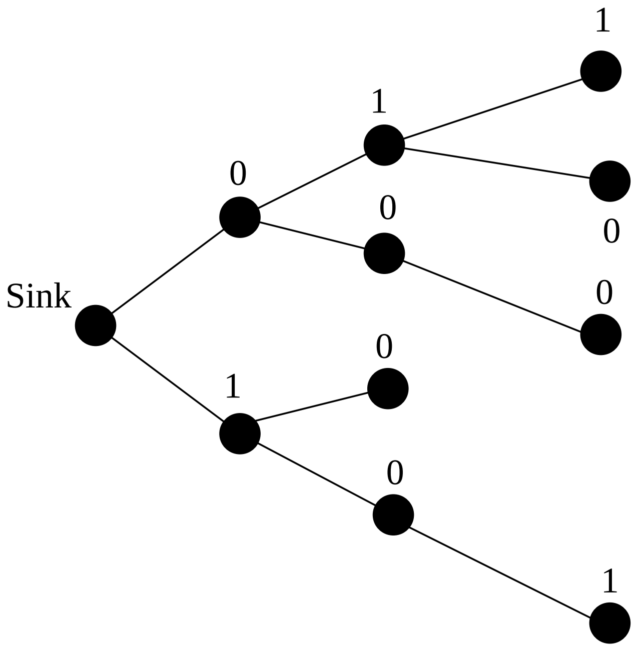
Processing and communications

- Sensing: huge variability of random field
- High sample rate means large amounts of data to be collected and processed
- Sampling should be fast enough to guarantee both *accuracy* and *delivery*
- The trade-off is between communications and processing
    - Concrete example [KaiserW]: 1GHz carrier frequency; antenna elevation of 1/2 wavelength; BPSK transmission, 10^{-6} BER, distance loss $\alpha = 4$, Rayleigh fading, ideal (nonoise) receiver.
    - Energy: 1Kb a distance of 100 m is approximately 3 J.
    - Processor: 100 MIPS/W executes 3 million operations

Quite clear: *if possible process locally* and save on communications

Processing and communications

- Simple(istic) example: majority voting in a n sensors WSNs
- Two events: $E = \{0, 1\}$; typical alarm detection
- Estimation: decide 1 if the number of ones detected is less than the number of 0s



- Assume: transmitted bits dictate the cost
- First strategy: send data to sink and decide

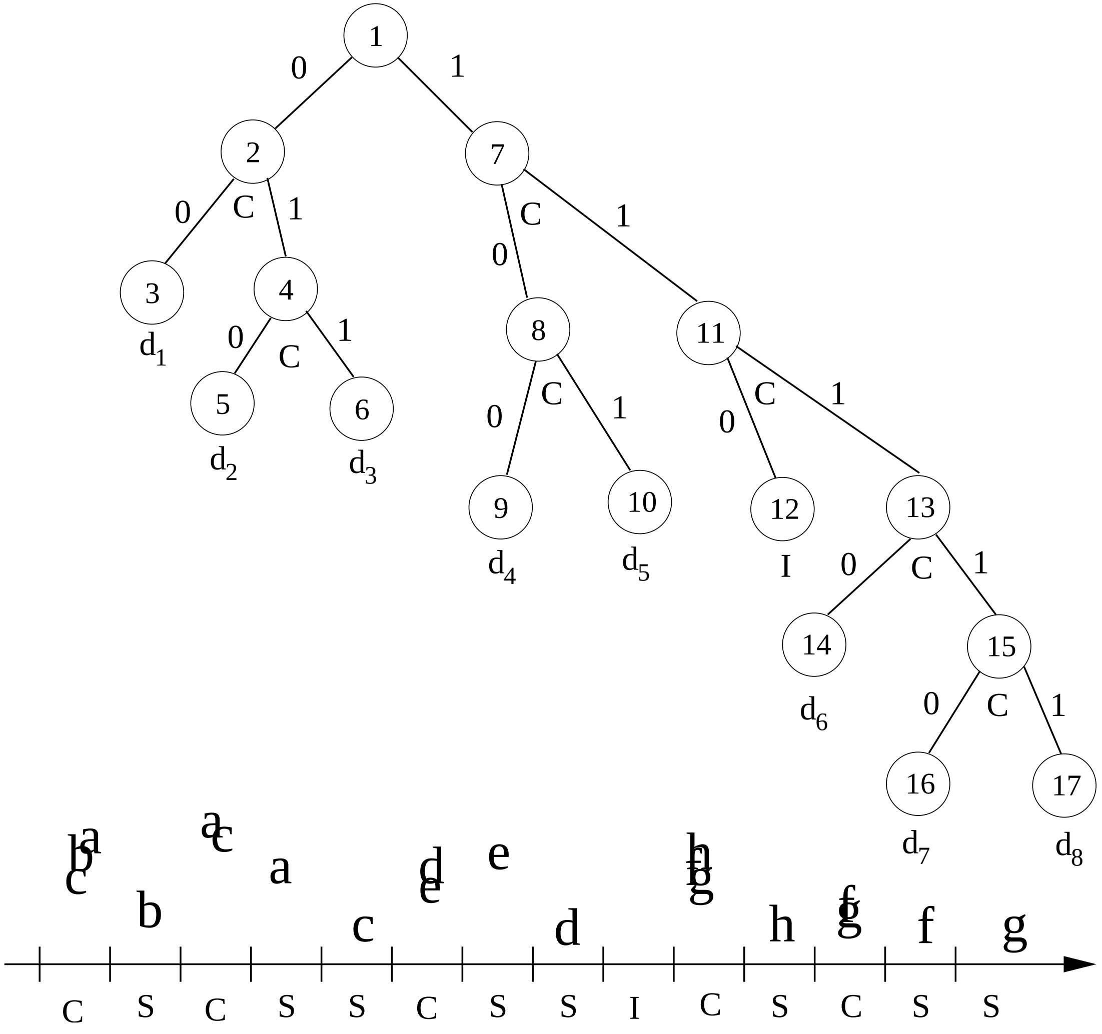
$$C = 1(n-1) + 2(n-1-2) + 3(n-1-3) + \ldots + \log(n-1)(n-1-2^{\log(n-1)-1}) = \Theta(n \log^2 n)$$

- Second strategy: each node evaluates the majority outcome of children, and encodes the number of votes: $C = \Theta(n \log n)$
- small gain but ...

Majority voting

- Again on majority voting: in *colocated* networks
- Approximated hypothesis testing is possible using *batch conflict resolution*: collisions do carry information on the number of contending nodes
- Data collection at the sink: collect and decide if $N_0 > N_1$ or viceversa
- [PZ] trade off between MAC and Application layer: let the application be aware of the Medium Access Control
- Leverage the information about the *number of transmitters*
- Stop contention as soon as the majority estimation error probability is below a certain threshold
- Still packetized solution: practically feasible

Majority voting



- Tree split algorithms; best implementations achieve almost half channel capacity
- No carrier sensing: every sensor tosses a coin
  - ▲ tail: transmit
  - ▲ head: skip the slot and toss again
- Channel has three states: I(dle),C(ollision),S(uccess)
- Need return channel to feedback transmission result

Majority voting

- Number of sensors N is unknown: just jumped into the sensor field
- Start collecting $\hat{N}_0$ and $\hat{N}_1$: eventually 2 binary split trees with p_0 and p_1
- TDMA collecting first 1s and then 0s
- n_0 number of sensor tossing 0 (they transmit): say N_{p_0}
- Bernoulli trials: $\mathbb{P}\{n_0 = N_{p_0} | N = N_0\} = \binom{N_0}{N_{p_0}} p_0^{N_0} (1 - p_0)^{N - N_0}$
- Most likely value: moda = $\lfloor (N + 1)p \rfloor$
- For small p_0, $1/p_0$ of them: $\mathbb{P}\{N = N_0 | n_0 = N_{p_0}\} \sim p_0 \, \mathbb{P}\{n_0 = N_{p_0} | N = N_0\}$
- Maximum likelihood approximate majority voting

$$\hat{N}_0 = \lfloor N_{p_0} / p_0 \rfloor \qquad \hat{N}_1 = \lfloor N_{p_1} / p_1 \rfloor$$

- [PZ] upper bound to the error probability: stop batch resolution the upper bound is below a threshold

Wireless networks as estimators

- Change of perspective: do you really need the sink node and elaborate only after data are collected there?
- Answer: no, there exist certain cases when the estimation can be made available *at each node* [Gian]
- Good news for pervasive communication/computing environments
- Using *gossip* propagation: only communicate with neighbors and propagate hop by hop (no need for end to end communication)
- Notice: no packetized communication assumed, can be done “bitwise”
- M sensors deployed: general not colocated multihop communications
- Sensor network to estimate $\mathbf{s} \in \mathbb{R}^p$
- Assumption of a linear model on the observations $\mathbf{x}_j \in \mathbb{R}^{L_j}$

$$\mathbf{x}_j = \mathbf{H}_j \mathbf{s} + \mathbf{n}_j$$

- $\mathbf{n}_j$ random vector noise with given covariance matrix Σ

Wireless networks as estimators

- Need additional hypothesis on full rank of $\mathbf{H} = [H_1^T H_2^T \dots H_M^T]^T$
- BLUE estimator $\hat{\mathbf{s}} = \mathbf{C}\mathbf{x}$, $\mathbf{C} = (\mathbf{H}^T \Sigma^{-1} \mathbf{H})^{-1} \mathbf{H}^T \Sigma^{-1}$
- Ambitious task: can you find a BLUE estimate at each sensor (if needed in infinite iterations)?
- Answer: yes and exchanging "messages" only with the neighborhood
- Two fully distributed algorithms:
  - full knowledge on $\mathbf{C}$
  - local knowledge on $\mathbf{H}_j$
- Through iterative optimizations consensus is achieved, meaning that *every sensor* converges to the $\hat{\mathbf{s}}$ [Gian]
- Notice: in order to build the algorithm, solutions can be found constraining the optimization to obtain consensus, i.e. same estimate, in sensors' neighborhood only

In-network computation

- Limit performances of sensor networks when computing is accounted for [Kum2,Kum3]
- Target: communicate a given function $f^n : \chi^n \rightarrow Y_n$ to the sink(s)
- Nodes take measures in a finite set χ and can communicate blocks of measures
- No channel errors accounted for
- [Kum2] under protocol model: important insight in improvements achievable with current technologies
- The big question: *how fast* can you sample the function of the field? Rate of computing allowed by a sensor network $R^n_{\max}$
- Proved strong dependence on:
  - topology: colocated vs multi-hop random planar
  - computed function

In-network computation

- Independence from the data arrival order: symmetric functions
- Type sensitive functions:
  - functions such that, from a given number of sensor readings on, there exist a fraction of the readings that are sufficient to discriminate output values, i.e. to state that they are different
  - sample functions: mean, median, mode
  - ex. $\chi = 1, 100$, mean function: fix the first half values, put remaining $1/2$ values either 1 or 100
- [Kum2] In colocated networks, for any possible collision free strategy,

$$R^n_{\max} = \Theta(1/n)$$

- [Kum2] In random planar networks, under suitable covering radius scaling, w.h.p.

$$R^n_{\max} = \Theta(1/\log n)$$

In-network computation

- Type threshold functions:
  - functions such that there exist a reference type vector θ such that (element-wise) larger data frequencies do not affect the reference function
  - sample functions: max,min,range
  - ex. $\max$, $\theta = \mathbf{1}$
- [Kum2] In colocated networks, for any possible collision free strategy,

$$R^n_{\max} = \Theta(1/\log n)$$

- [Kum2] In random planar networks under suitable covering radius scaling, w.h.p.

$$R^n_{\max} = \Theta(1/\log\log n)$$

- Type threshold functions are exponentially easier to compute than type sensitive functions

Location Detection

Location detection

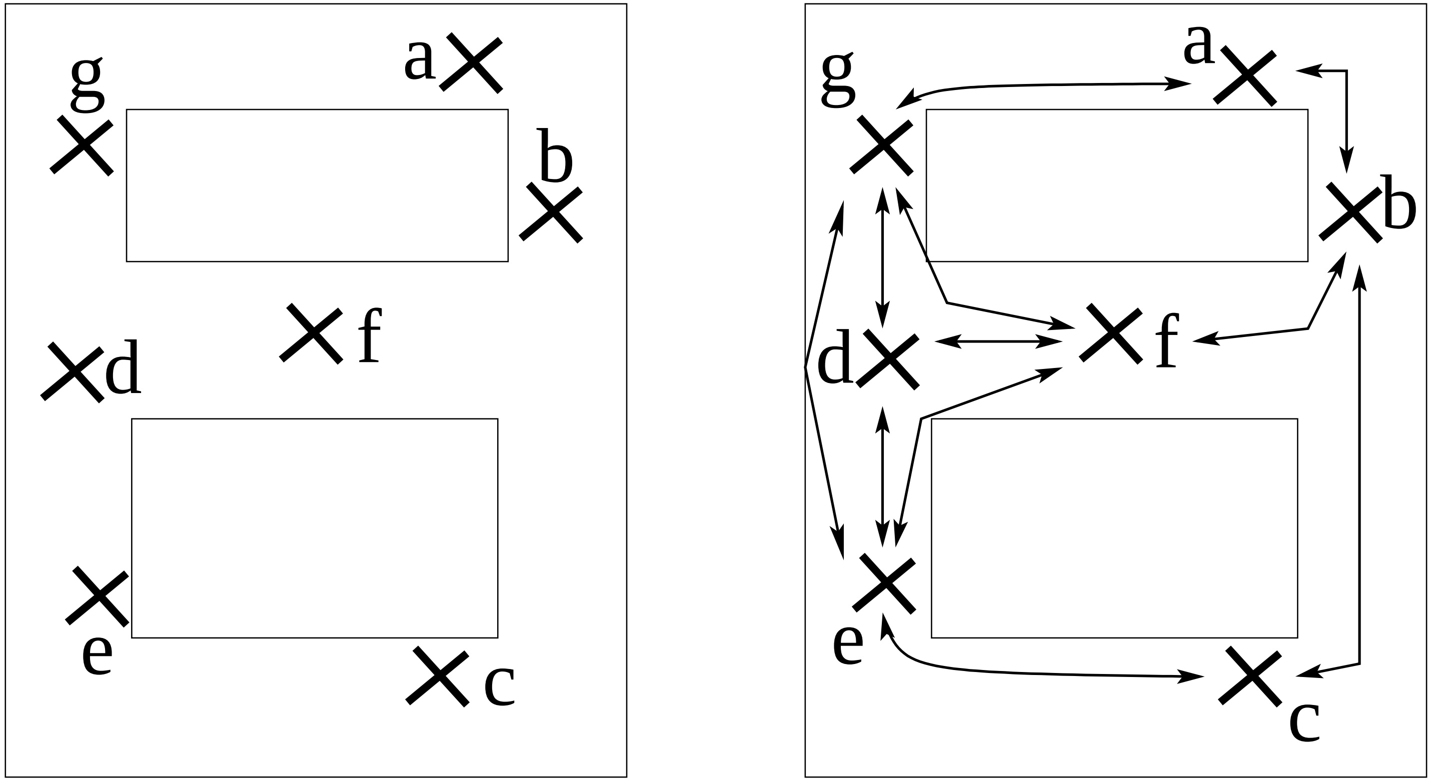
- Location detection is a very popular application of WSNs
- Existing examples of indoor location detection systems:
  - Infrared: Olivetti Research Laboratory *Active Badge* location system. A badge periodically emits a unique ID using diffused IR received by one of several receivers scattered throughout a building;
  - Ultrasound: *Active Bat* or MIT *Cricket* measure time-of-flight of ultrasound with respect to a reference signal;
  - RF propagation: Microsoft *RADAR* pre-computed SNR map and compares received signals; distances from base stations are triangulated to obtain the position;

Emergency response: robustness is a key issue, and location detection systems should prove *robust* to *multiple sensor failures*.

Location detection

- A location detection system designed through Identifying codes [Lev]
- System makes use of a robust modification of identifying codes built over an arbitrary topology,
- *location tracking*: observer transmits IDs and the system determines his location
- Let us focus on the dual *location service*: observer can determine his location
- Transmitters occupy certain positions: each transmitter periodically broadcasts a unique ID
- Observer waits T at his current position collects the ID’s from the packets he receives
- Collisions can be efficiently solved by common MAC protocols within T
- The covered locations are all those areas receiving a set of IDs (problem of identifying graphs)

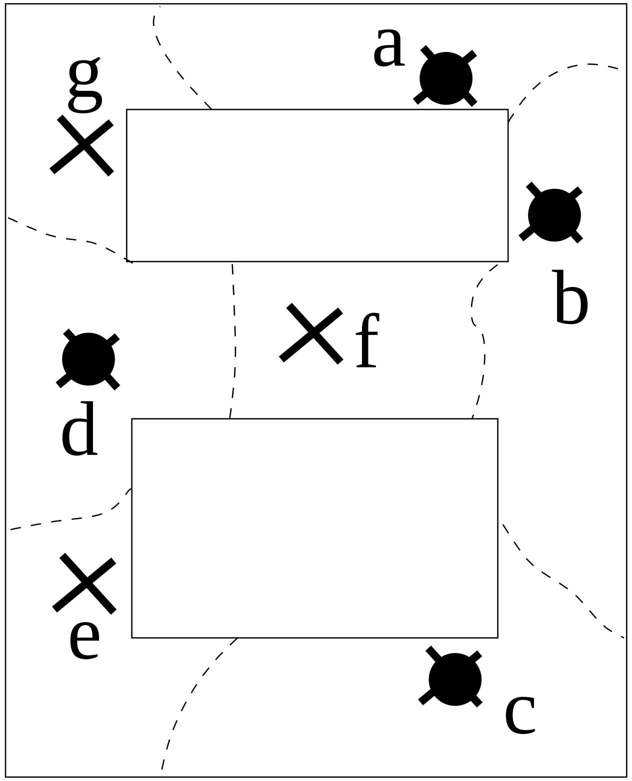
Location detection



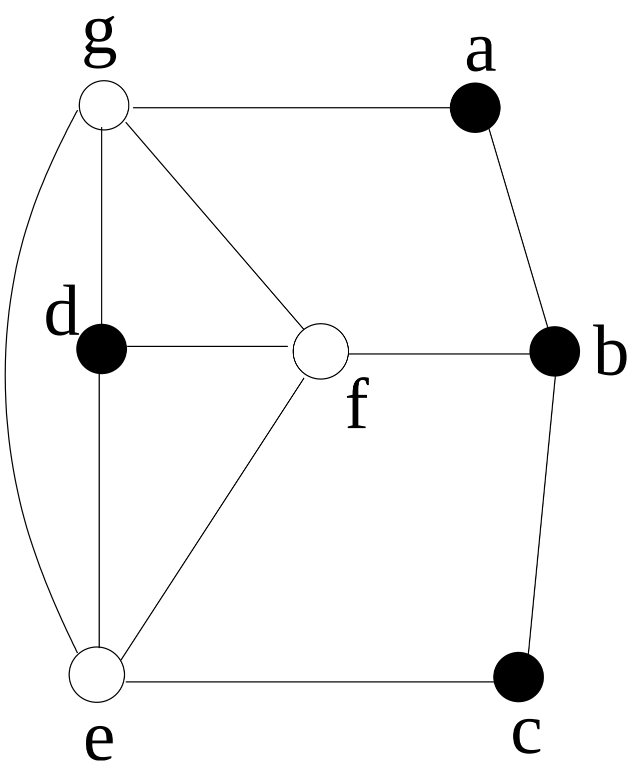
(a) Discrete Locations.

(b) Connectivity.

Location detection



(c) Sensor placement: black dots



(d) Position of the sensors on the graph.

$ID(a) = \{a, b\}$

$ID(b) = \{a, b, c\}$

$ID(c) = \{b, c\}$

$ID(d) = \{d\}$

$ID(e) = \{c, d\}$

$ID(f) = \{b, d\}$

$ID(g) = \{a, d\}$

Location detection

- Procedure: model a physical environment with a graph $G = (V, E)$,
- vertices V indicate locatable regions
- edges E connect regions with RF connectivity
- A non-empty subset $\mathbb{C} \subseteq V$ is called a *code*, its elements are called *codewords*.
- Given a code $\mathbb{C}$, the *identifying set* of a vertex $v \in V$ is defined to be

$$I_{\mathbb{C}}(v) = B(v) \cap \mathbb{C}. \tag{1}$$

- Code $\mathbb{C}$ is called an identifying code if for every $u, v \in V$

$$I_{\mathbb{C}}(u) \neq I_{\mathbb{C}}(v);$$

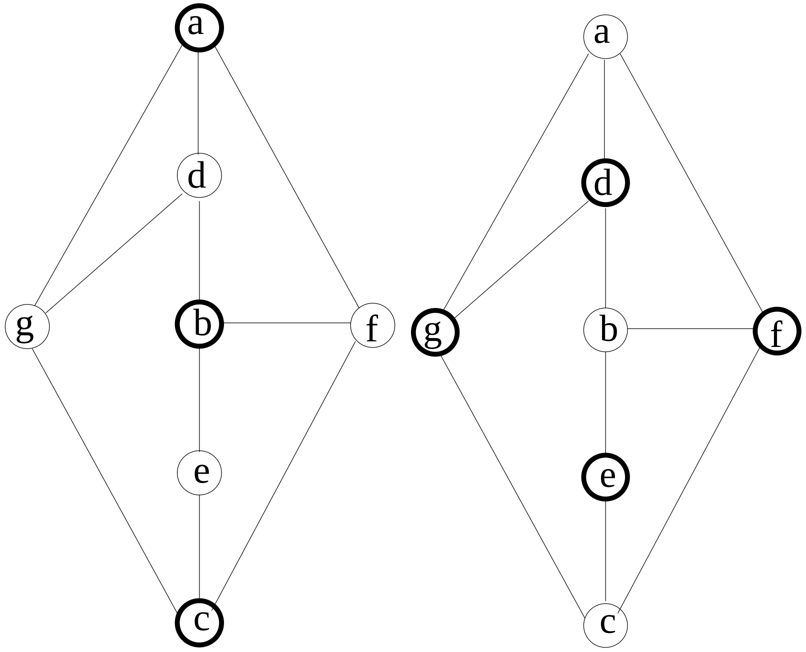
- $\mathbb{C}$ is *irreducible* if deletion of any codeword from $\mathbb{C}$ results in a code that is no longer an identifying code.
- $G = (V, E)$ is *distinguishable* if an identifying code exists for it

Location detection

- Location detection in general relies on *irregular graphs*
- Distinguishability can be enforced *a posteriori*
- Generation of an *optimal* identifying code is $\mathcal{NP}$-complete.
- Problem: *Given a distinguishable graph* $G = (V, E)$*, compute a subset* $\mathbb{C}$ *of* V *such that* $\mathbb{C}$ *is an identifying code for* G *and* $\mathbb{C}$ *is irreducible.*

```
ID-CODE($G$, $\mathbf{a}$)
$\mathbb{C} = V$
if $\mathbb{C}$ is not an identifying code
   do  EXIT
for each vertex $x \in \mathbf{a}$, taken in order
   do  $D = \mathbb{C} \setminus \{x\}$
       if $\exists\, u, v \in V$ such that $I_D(u) = I_D(v)$
             $\mathbb{C} = \mathbb{C}$
       else  $\mathbb{C} = D$
return $\mathbb{C}$
```

Location detection



- Vertices visited $\mathbf{a} = \{f, g, d, e, a, b, c\}$, then $\mathbb{C} = \{a, b, c\}$
- Conversely with $\mathbf{a} = \{a, b, c, d, e, f, g\}$, then $\mathbb{C} = \{d, e, f, g\}$
- The code $\mathbb{C}$ returned by ID-CODE is irreducible.
- Complexity is $O(|V|^2 \log |V|)$

Location detection

The case for robustness:

- Destruction of ID-transmitting vertices through the emergency agent (*e.g.* , fire, water, explosion).
- Variation of radio paths due to changes in building structure (*e.g.* , walls collapsing, furniture shifting, people moving).
- Failure of ID reception due to medium access control scheme limitations.
- Due to reflections or some other channel variations due to extreme conditions, some spurious packets may come to the observer.

Def. 1 *An identifying code* $\mathbb{C}$ *over a given graph* $G = (V, E)$ *is said to be* r-robust *if*

$$I_{\mathbb{C}}(u) \oplus A \neq I_{\mathbb{C}}(v) \oplus B$$

for all $u, v \in V$ *and* $A, B \subseteq V$ *with* $|A|, |B| \leq r$.

Location detection

- $A \oplus B = (A \cup B) \setminus (A \cap B)$
- $d_{\min}(\mathbb{C}) \triangleq \min_{u,v \in V} |I_{\mathbb{C}}(u) \oplus I_{\mathbb{C}}(v)|$
- A code $\mathbb{C}$ is r-robust if and only if

$$d_{\min}(\mathbb{C}) \geq 2r + 1.$$

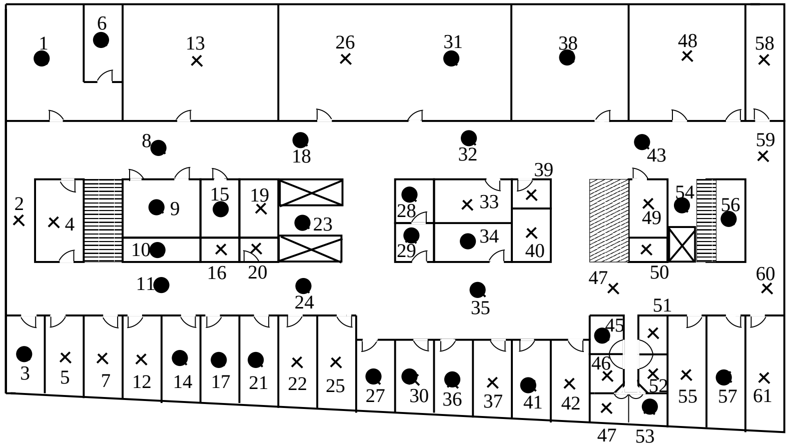
- The maximum robustness achievable by an identifying code over a graph $G = (V, E)$ is given by $d_{\min}(V)$, and is upper bounded by the sum of the degrees of any two vertices in the graph.
- Of course the problem is still $\mathcal{NP} - complete$
- Quite simple to modify the greedy algorithm presented before

Location detection

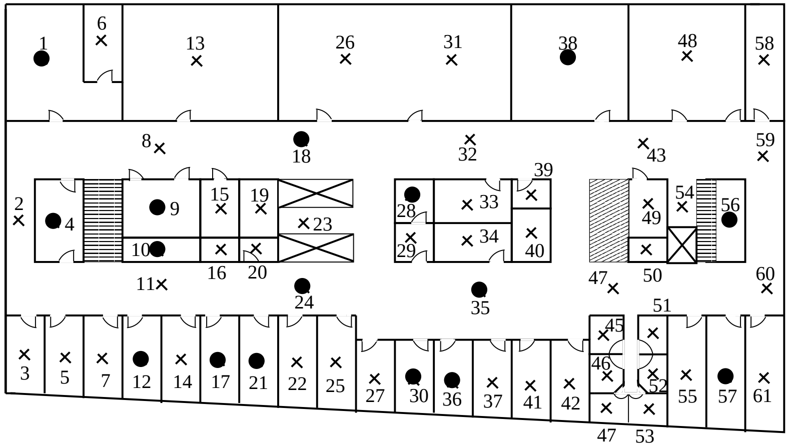
- r-ID-CODE algorithm generates r-robust identifying codes for an arbitrary graph
- Not every graph has a r-robust code
- But deletion of a node from a code decreases $d_{\min}(\mathbb{C})$
- The robust version of the algorithm:

```
r-ID-CODE($G$, $\mathbf{a}$, $r$)
$\mathbb{C} = V$
if $d_{\min}(\mathbb{C}) \leq 2r$
   do  EXIT
for each vertex $x \in \mathbf{a}$
   do  $D = \mathbb{C} \setminus \{x\}$
       if $d_{\min}(D) \leq 2r$
             $\mathbb{C} = \mathbb{C}$
       else  $\mathbb{C} = D$
return $\mathbb{C}$
```

Location detection



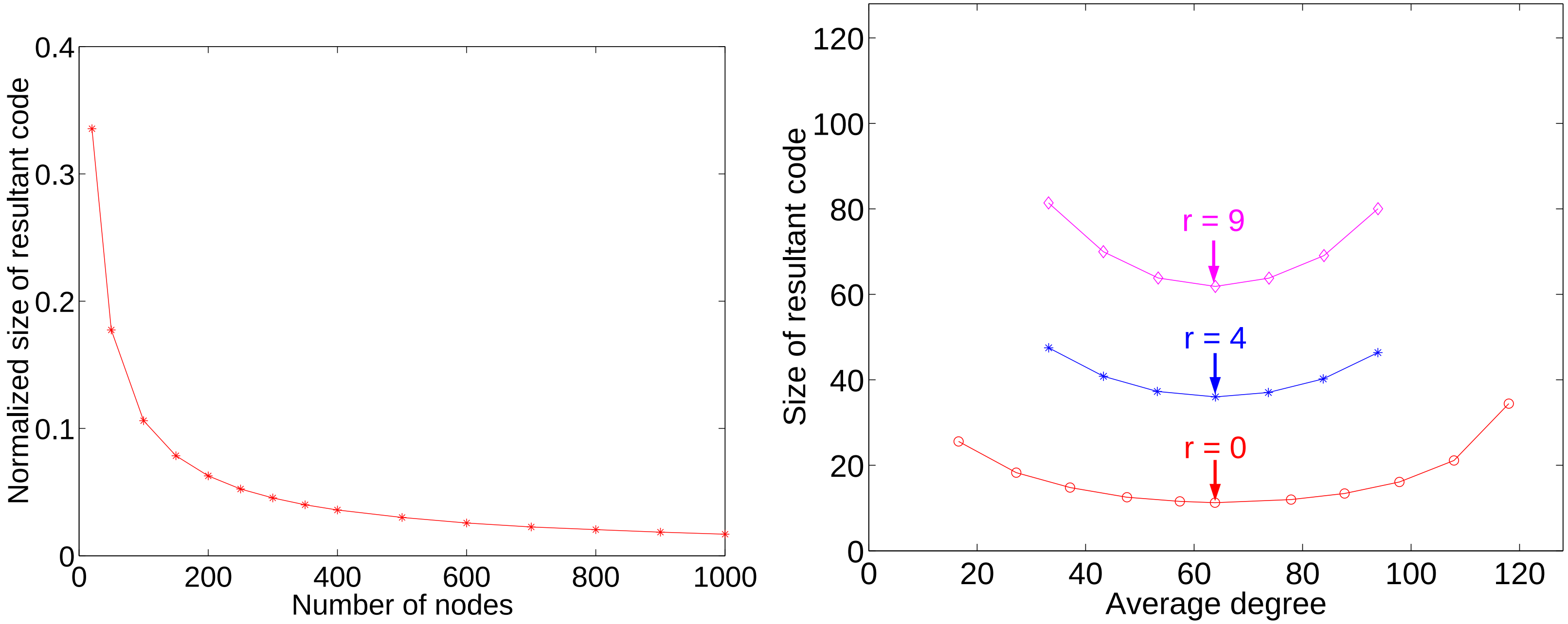
(g) 1-robust code.



(h) 0-robust code.

- 0-robust code requires 16 transmitters among the 64 points of resolution
- 1-robust code requires 32

Location detection



- Fraction of sensorized locations scales at the increase of the number of locations
- Asking for robustness costs in the *size* of the code, i.e. number of sensorized locations
- Complexity is $O(|V|^3)$

Bibliography

- [Ak] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, *“Wireless sensor networks: a survey”*, Elsevier Computer Networks, Vol 38, 2002, pp. 393-422
- [KaiserW] G. J. Pottie and W. J. Kaiser, *“Wireless integrated network sensors”*, Communications of the ACM, May 2000, Vol. 43, n0. 5
- [Es1] C. Intanagonwiwat, D. Estrin, R. Govidan and Fabio Silva, *“Impact of Network Density on Data Aggregation in Wireless Sensor Networks”*
- [Es2] C. Intanagonwiwat, D. Estrin and R. Govidan, *“Impact of Network Density on Data Aggregation in Wireless Sensor Networks”*
- [SmarDust] B. Atwood, B. Warneke, K.S.J. Pister, *"Preliminary Circuits for Smart Dust"*, Proceedings of the 2000 Southwest Symposium on Mixed-Signal Design, San Diego, California, February 27-29, 2000, pp. 87-92.
- [GuptaKumar] P. Gupta and P. R. Kumar, *“The Capacity of Wireless Networks”*, IEEE Trans. on Information Theory, Vol. 46, No. 2, March 2000.
- [DT] O. Dousse and P. Thiran, *“Connectivity vs Capacity in Dense Ad Hoc Networks”*, in the proceedings of INFOCOM 2004, Hong Kong.
- [GT] M. Glossglauser and D. Tse, *“Mobility Increases the Capacity of Ad Hoc Wireless Networks”*, IEEE/ACM trans. on Networking, Vol. 10, no. 4, August 2002.
- [GV] M. Gaspar and M. Vetterli, *“On the Capacity of Wireless NEtworks: the Relay Case”*, in the proceedings of INFOCOM 2002, New York.

- ▲ [ElGa] A. El Gamal, J. Mammen, B. Prebhakar and D. Shah, *“Throughtput-Delay Trade-off in Wireless Networks”*, in the proceedings of INFOCOM 2004, Hong Kong.
- ▲ [DFT] O. Dousse, M. Franceschetti and P. Thiran, *“On the Throughput Scaling of wireless Relay Networks”*, IEEE Transactions on information Theory, Vol. 52, No. 6, June 2006.
- ▲ [PZ] E. Bottega, P. Popowsky, M. Zorzi H. Yomo and r. Prasad, *“Hypothesis Testing over a Random Access Channel in Wireless Sensor Networks ”*, in the proceedings of EURASIP EUSIPCO 2006, Florence.
- ▲ [Gian] I. Schizas, A. Ribeiro and G. B. Giannakis “Distributed Estimation with Ad Hoc Wireless Sensor Networks”, in the proceedings of EURASIP EUSIPCO 2006, Florence.
- ▲ [Lev] M. Karpovsky, K. Chakrabarty and L. Levitin, *“A New Class of Codes for Identification of Vertices inGraphs”*, IEEE Transactions on Information Theory", Vol. 44, no. 2, 1998, pp. 599-611"
- ▲ [SFR] S. Ray, R. Ungrangsi, F. De Pellegrini, A. Trachtenberg and D. Starobinski, *“Robust Location Detection in Emergency Sensor Networks*”, in the proceedings of *IEEE INFOCOM 2003*.
- ▲ [SchPap] D. S. Scherber and H. Papadopoulos, *“Distributed Computing of Averages over Ad-hoc Networks”*, IEEE JSAC, vol. 23, No. 4, April 2005.
- ▲ [Kum2] A. Giridhar and P. R. Kumar, *Computing and Communicating Functions over Sensor Networks”*, IEEE JSAC, pp. 755-764, vol. 23, no. 4, April 2005.
- ▲ [Kum3] A. Giridhar and P. R. Kumar, *Towards a Theory of In-Network Ccomputation in Wireless Sensor Networks”*, IEEE Communications Magazine, vol. 44, no. 4, pp. 98-107, April 2006.
- ▲ [ZaDe1] A. Zanella and F. De Pellegrini, “Statistical Analysis of the Service Time in Saturated IEEE 802.11 Networks”, IEEE Communication Letters, March 2005.

- [ZaDe2] A. Zanella and F. De Pellegrini, *“Mathematical Analysis of IEEE 802.11 Energy Efficiency”*, in the proceedings of WPMC 2004.
- [KuMior] A. Kumar, E. Altman, D. Miorandi and M. Goyal, *“New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs"*, to appear in IEEE/ACM Trans. on Networking.
- [Bianchi] Giuseppe Bianchi, *“Performance Anaylsis of the IEEE 802.11 Distributed Coordination Function"*, IEEE Journal on Selected Areas in Communications, Vol. 18, Number 3, March 2000.
- [KamSurvey] J. N. Al-Karaki and A. E. Kamal, *“Routing Techniques in Wireless Sensor Networks: A Survey"*, IEEE Wireless Communications, pp. 6-28, vol. 11, no. 6, 2004.