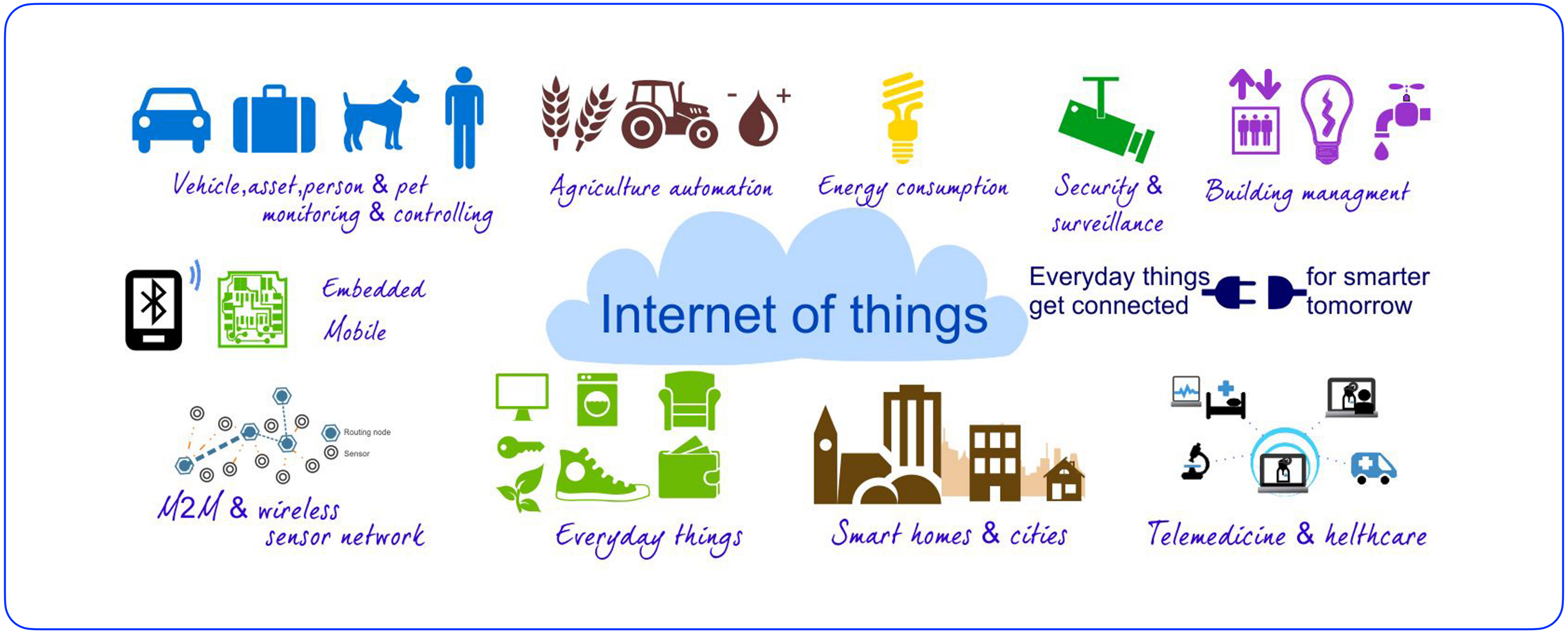
The Min-entropy of Distributed Wireless Link Scheduling Algorithms under Arbitrary Interference

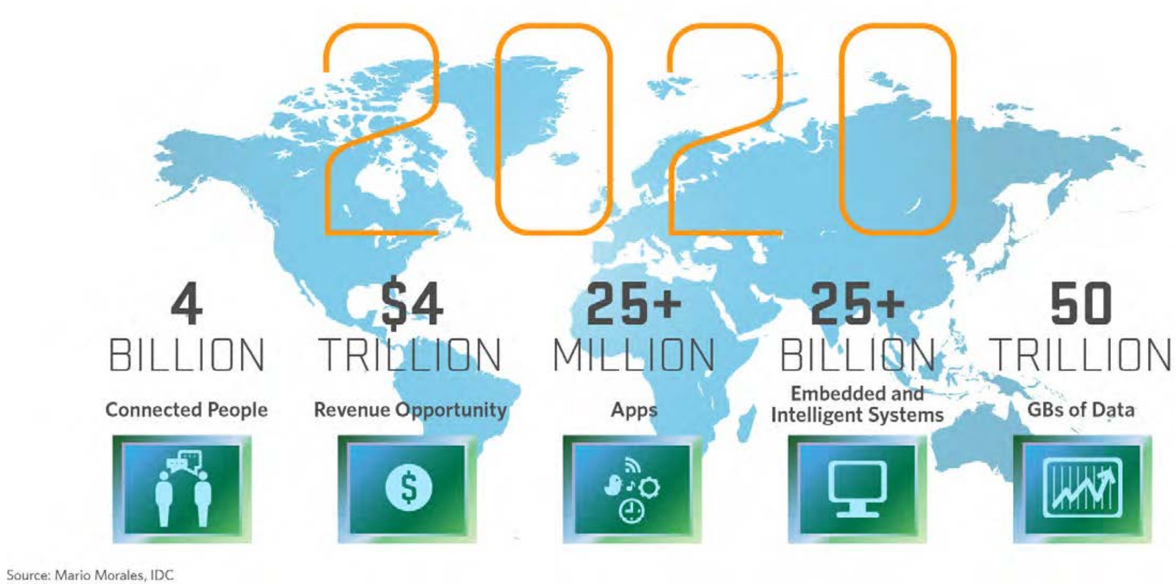
Dariusz R. Kowalski
Augusta University

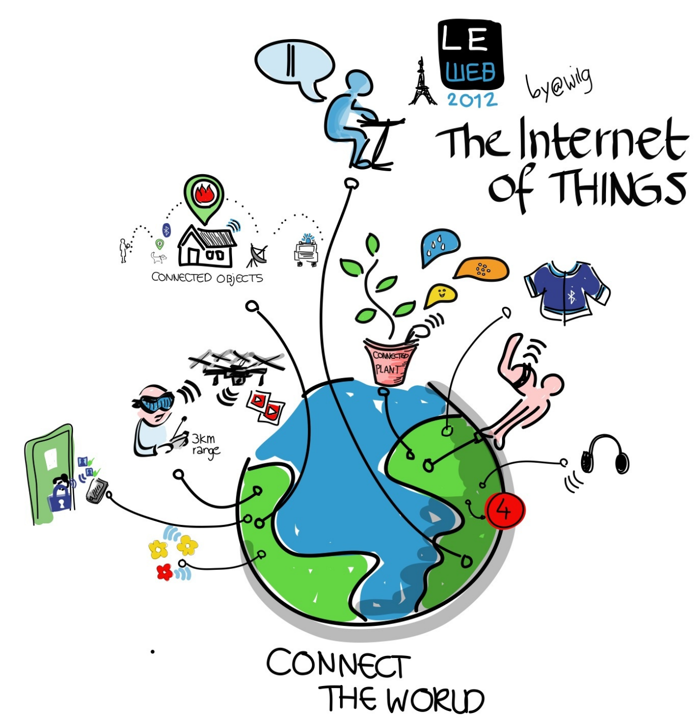
Miguel A. Mosteiro
Pace University

ISIT TAIPEI 2023

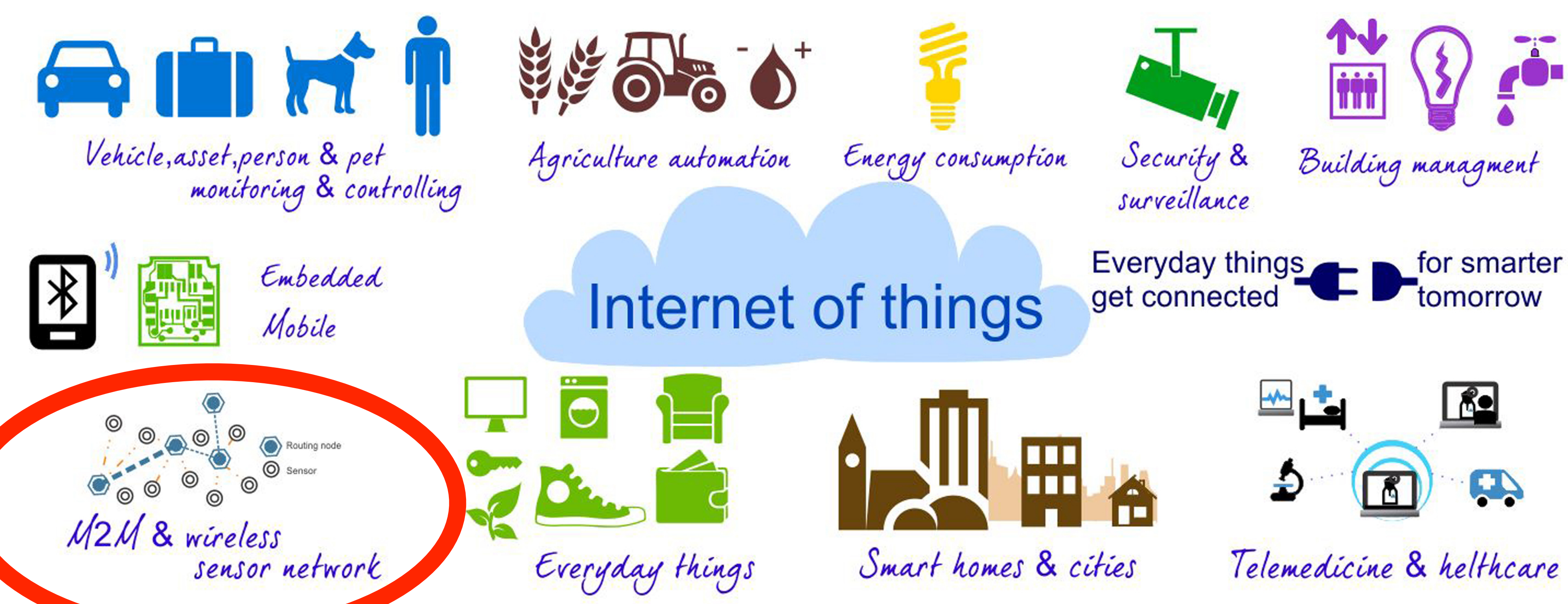
Application: the Internet of Things



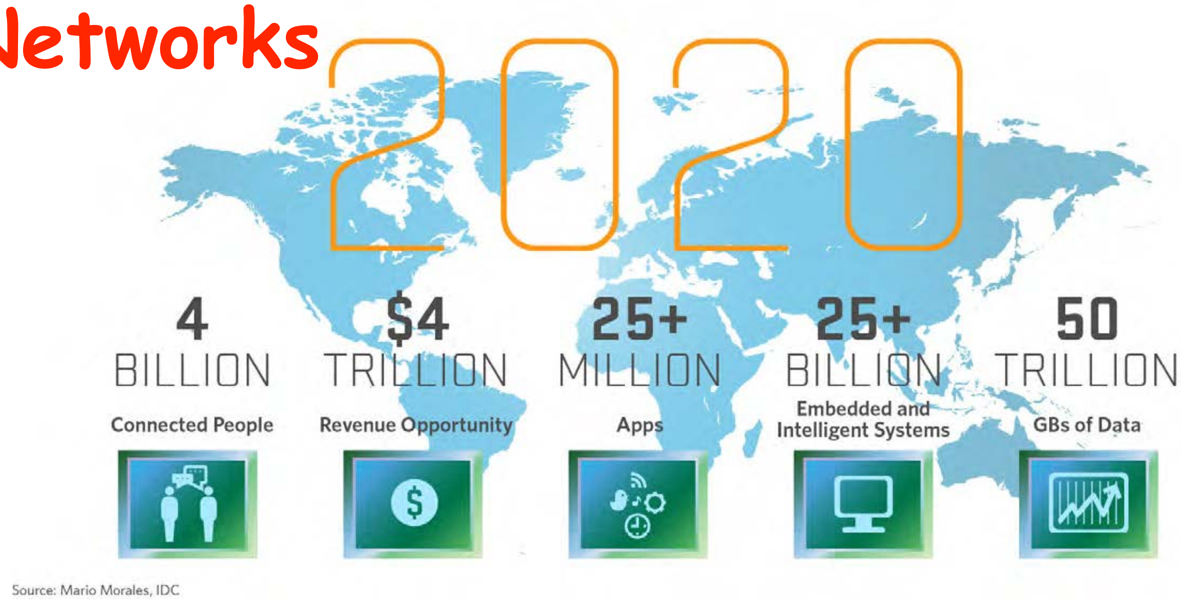


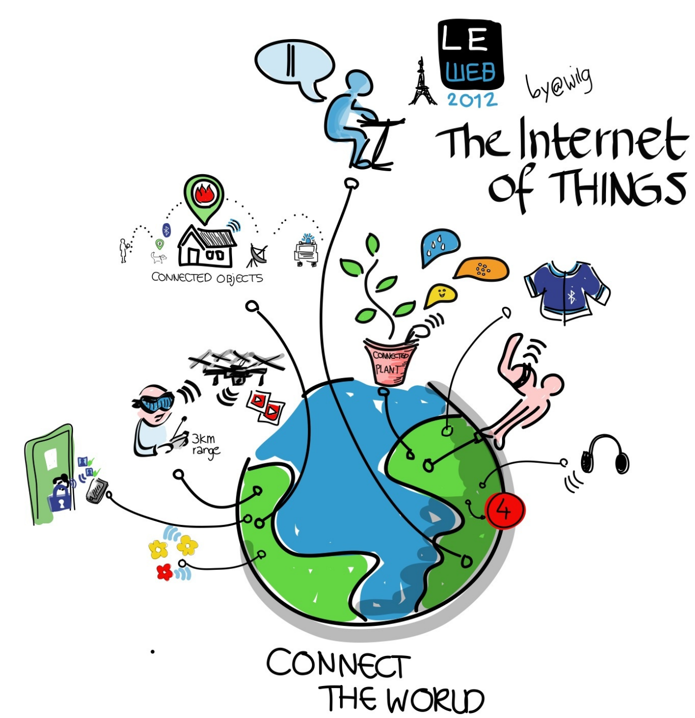


Application: the Internet of Things



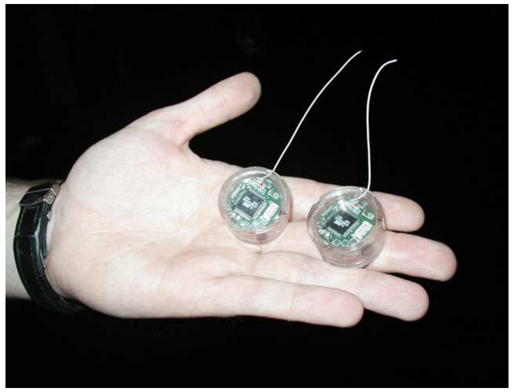
Ad-hoc Wireless Networks





Ad-hoc Wireless Networks

Example: A Sensor Network



Intel Berkeley Research Lab

Capabilities

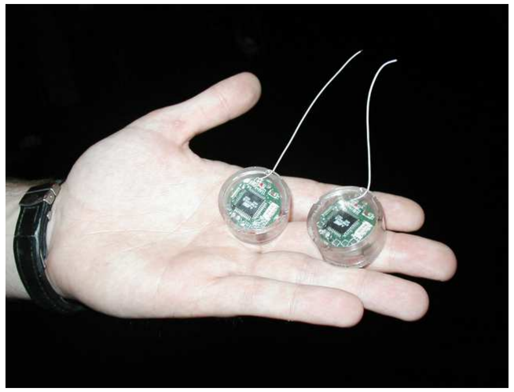
- processing
- sensing
- communication

Limitations

- range
- memory
- life cycle

Ad-hoc Wireless Networks

Example: A Sensor Network



Intel Berkeley Research Lab

Capabilities

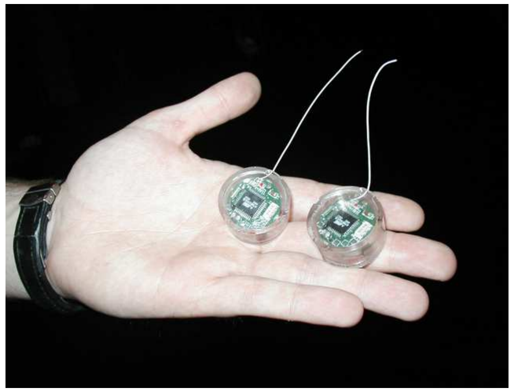
- processing
- sensing
- communication

Limitations

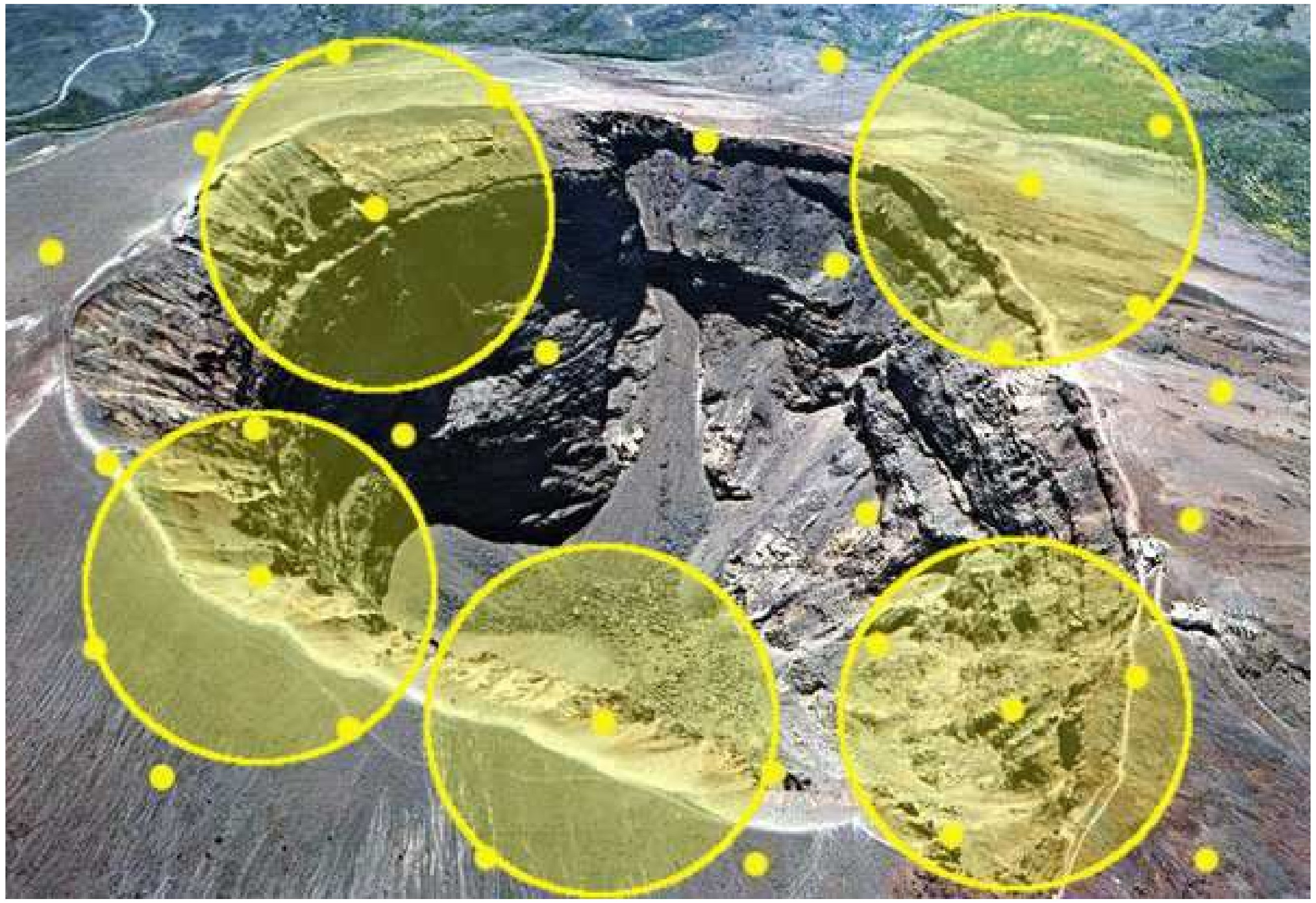
- range
- memory
- life cycle

Ad-hoc Wireless Networks

Example: A Sensor Network



Intel Berkeley Research Lab



Capabilities

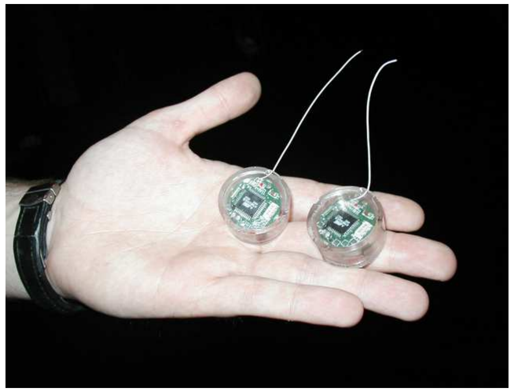
- processing
- sensing
- communication

Limitations

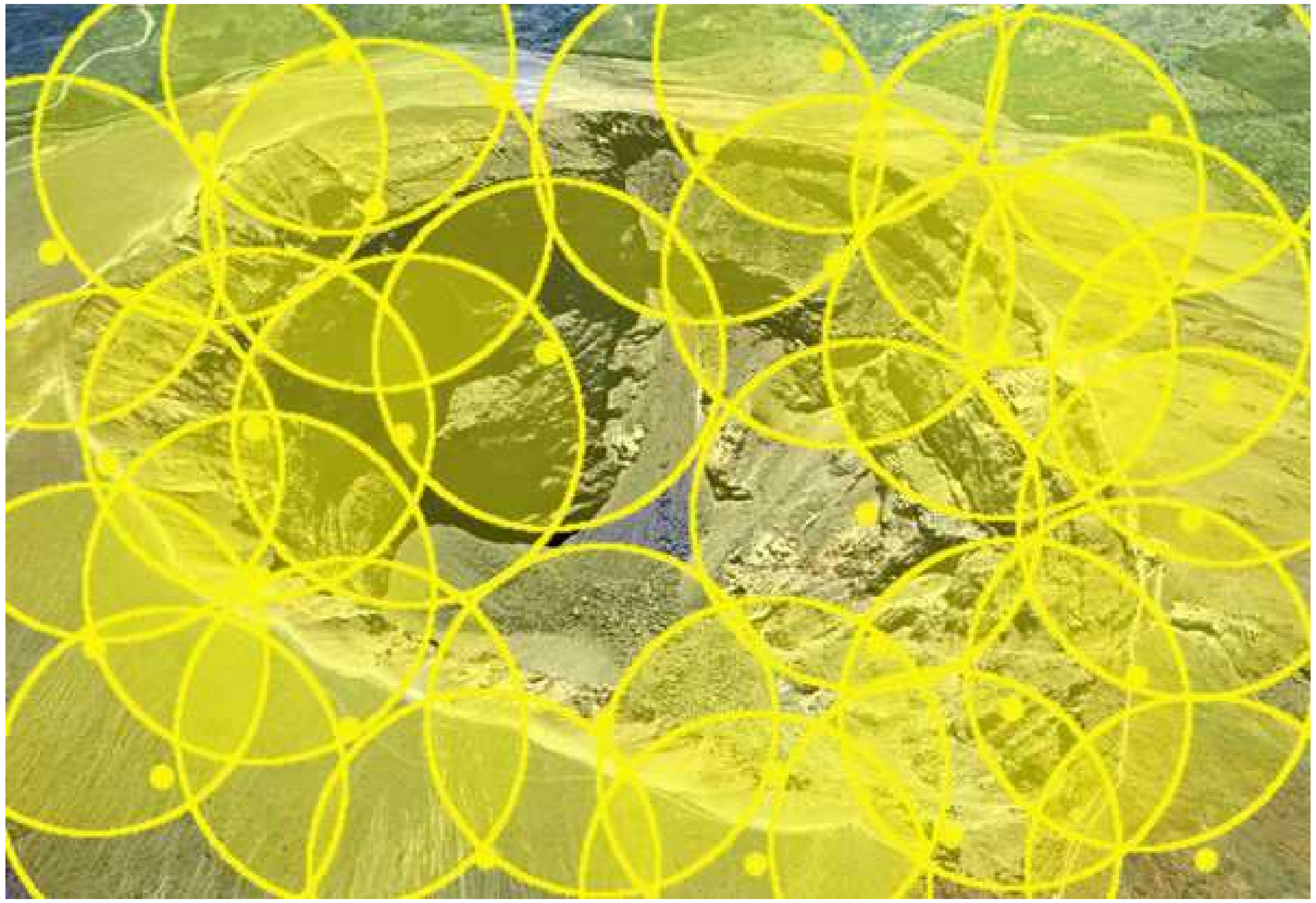
- range
- memory
- life cycle

Ad-hoc Wireless Networks

Example: A Sensor Network



Intel Berkeley Research Lab



Capabilities

- processing
- sensing
- communication

Limitations

- range
- memory
- life cycle

Models for Wireless Networks

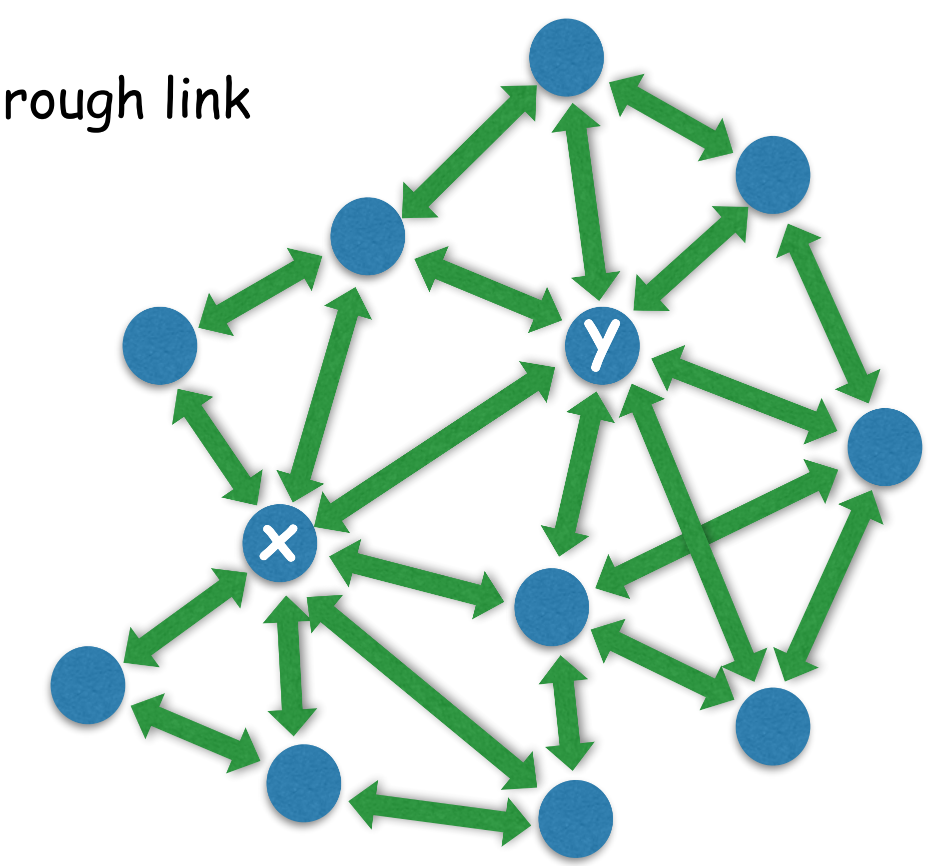
- **Topology Models :**
  - Undirected Graph
  - Unit Disk Graph
  - Time-varying Graph
- **Node Capabilities Models :**
  - Computational Resources
  - Communication Capabilities
  - Weak Sensor Model
- **Interference Models :**
  - Radio Network (RN)
  - Signal to Interference plus Noise Ratio (SINR)
  - Affectance (AFF)

Interference Models

Affectance Model [1,2,3]:

$$a((u,v),(x,y))$$

function quantifying interference of communication through link (u,v) on communication through link (x,y).



- **Collision/success:**

  For any link (x,y),

  a transmission from x is received by y at time t

  if and only if

  » x transmits at time t and

  » $\displaystyle\sum_{(u,v)\in L(t)} a((u,v),(x,y)) < 1$

$L(t) \subseteq E$: set of links whose transmitters transmit at time t

[1] Halldórsson and Wattenhofer. ICALP 2009.

[3] Fanghänel, Kesselheim and Vöcking. ICALP 2009.

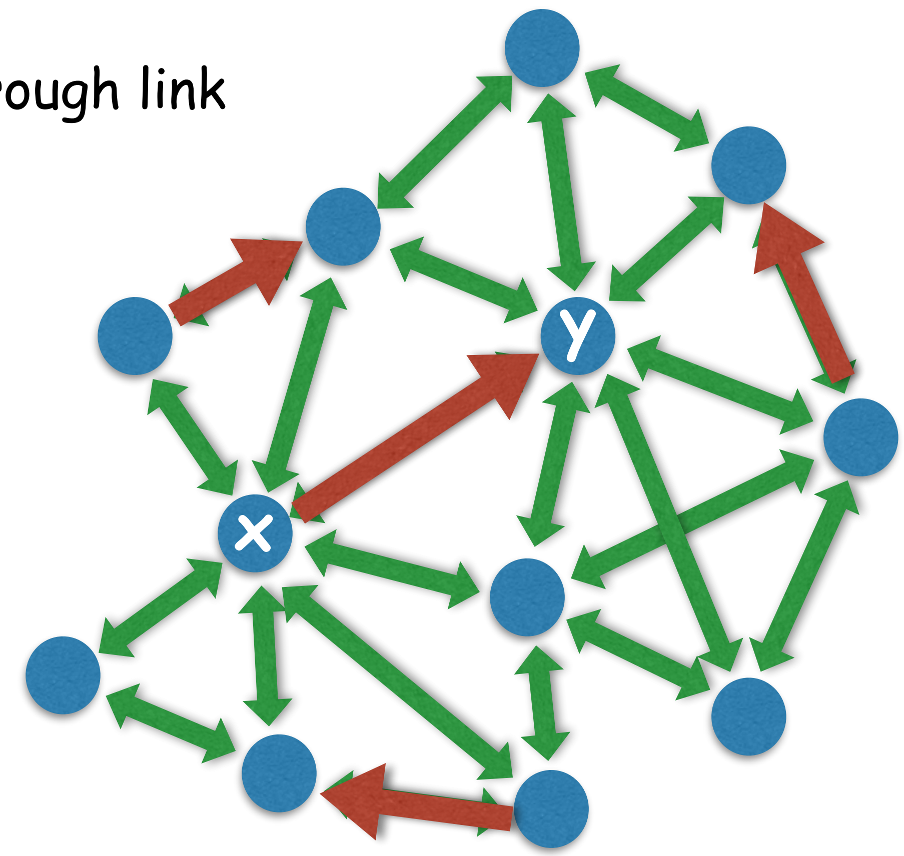
[3] Kesselheim and Vöcking. DISC 2010.

Interference Models

Affectance Model [1,2,3]**:**

$$a((u, v), (x, y))$$

function quantifying interference of communication through link (u, v) on communication through link (x, y).



- **Collision/success:**

  For any link (x, y),

  a transmission from x is received by y at time t

  if and only if

  » x transmits at time t and

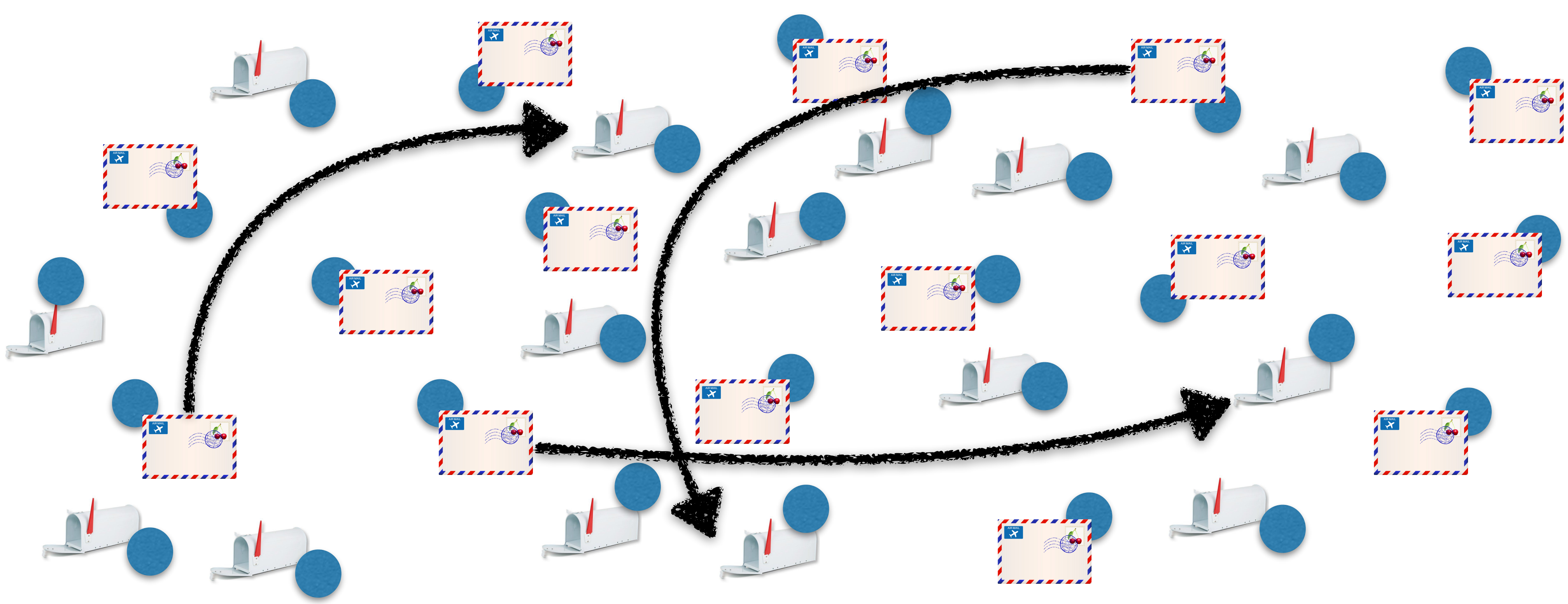
  » $\displaystyle\sum_{(u,v)\in L(t)} a((u, v), (x, y)) < 1$

  $L(t) \subseteq E$: set of links whose transmitters transmit at time t

[1] Halldórsson and Wattenhofer. ICALP 2009.
[3] Fanghänel, Kesselheim and Vöcking. ICALP 2009.
[3] Kesselheim and Vöcking. DISC 2010.

Link Scheduling Problem

- **Scenario :**
  - n network nodes called senders
  - n network nodes called receivers
  - each sender holds a message to be delivered to some receiver
  - each (sender,receiver,message) called a request
  - successful delivery of a message called a realization of the request



Link Scheduling Problem

- **Scenario :**
  - n network nodes called senders
  - n network nodes called receivers
  - each sender holds a message to be delivered to some receiver
  - each (sender,receiver,message) called a request
  - successful delivery of a message called a realization of the request
- **Conditions :**
  - realization implemented through wireless communication

    $\Rightarrow$ affectance among concurrent attempts of realization

    $\Rightarrow$ concurrent attempts may fail
  - unique node ID's, unknown to other nodes
  - time slotted in rounds of communication
- **Goal :**
  - realize all requests

Link Scheduling Problem

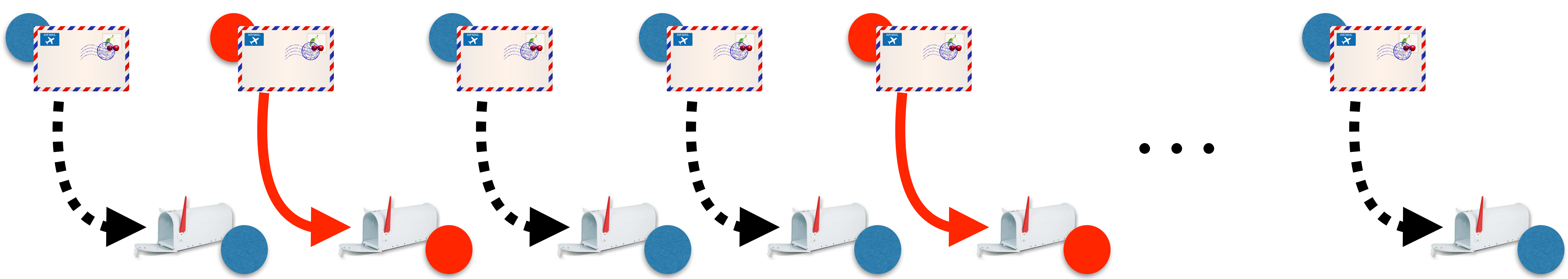
- **Input :**
  - set L of n requests

- **Output :**
  - transmissions schedule to realize all requests under arbitrary affectance

Link Scheduling Problem

- **Input** :
  - set L of n requests
- **Output** :
  - transmissions schedule to realize all requests under arbitrary affectance

e.g. realization attempts (transmissions) in round t :

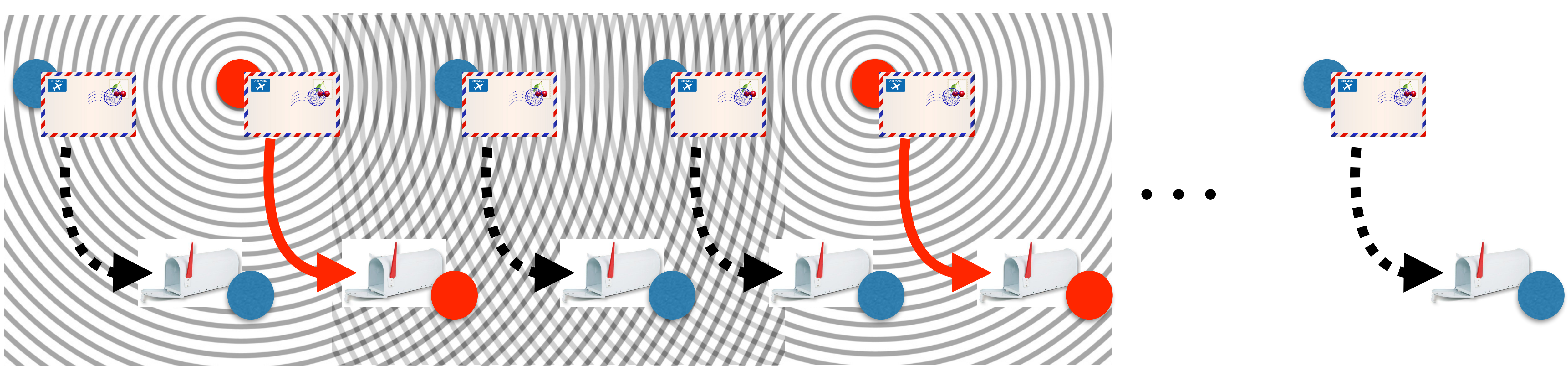


Link Scheduling Problem

- **Input** :
  - set L of n requests

- **Output** :
  - transmissions schedule to realize all requests under arbitrary affectance

e.g. realization attempts (transmissions) in round t :



affectance

Protocols Studied

- **Algorithms :**
  - distributed: each (sender,receiver) run their own algorithm, no centralized entity, ignoring messages from any other nodes
  - non-adaptive, except for switching off after realization. That is, requests are not aware of other realizations, and there are no control messages other than acknowledgements (to the transmitter only).
  - deterministic and randomized

- **Information available :**
  - each node knows only n and its own ID

Protocols Studied

- **Performance metrics :**
  - length of schedule: number of rounds to realize all requests
  - per request min-entropy . That is, the number of bits needed by the random variables used by the local algorithm used by each request.

both given as functions of n and the maximum average affectance [1]:

$$A(L) = \max_{L' \subseteq L} \left\{ \frac{1}{|L'|} \sum_{(u,v)\in L'} \sum_{(x,y)\in L'} a((u,v),(x,y)) \right\}$$

[1] Kesselheim and Vöcking. DISC 2010.

Previous Work

Distributed	Acks	Bound	power assignment p	Reference
No	No	$\frac{ALG}{OPT} \leq 12\lceil 2\tau^\alpha \rceil^2$	uniform	ICALP'09 [1]:Thm 3
Yes	No	$O(I(L)\log n)$ *whp*	$p(\ell) = cd(\ell)^\alpha$	TCS'11 [3]:Thm 5
No	Yes	$O\big(I(L) + \log^2 n\big)$ *whp*	$p(\ell) = cd(\ell)^\alpha$	TCS'11 [3]:Thm 8
(*)	No	$\Omega(I(L))$	linear	TCS'11 [3]:Thm 1
(*)	No	$\Omega\left(\frac{I(L)}{\log \frac{d_{\max}}{d_{\min}} \log n}\right)$	general	TCS'11 [3]:Thm 2
(*)	No	$\Omega\left(\frac{I(L)}{\log \frac{d_{\max}}{d_{\min}}}\right)$	general	TCS'11 [3]:Thm 4
Yes	Yes	$O\big(\overline{A}(L,p)\log n\big)$ *whp*	monotonic	DISC'10 [4]:Thm 6
(*)	Yes	$\Omega\left(\frac{\overline{A}(L,p)}{\log n}\right)$	monotonic	DISC'10 [4]:Thm 10

TABLE I

PREVIOUS BOUNDS FOR LINK SCHEDULING UNDER LESS GENERAL MODELS OF INTERFERENCE.

$\tau = 2 + \max\left\{2, \left(2^6 3\beta \frac{\alpha-1}{\alpha-2}\right)^{1/\alpha}\right\}$;

MEASURE OF INTERFERENCE $I(L) = \max_{w\in V} \sum_{(u,v)\in L} \min\{1, d(u,v)^\alpha / d(u,w)^\alpha\}$;

$\overline{A}(L,p)$ IS $A(L)$ FOR SINR WITH POWER ASSIGNMENT p;

MONOTONIC POWER ASSIGNMENT: (1) $d(\ell) \leq d(\ell') \Rightarrow p(\ell) \leq p(\ell')$ AND $\frac{p(\ell)}{d(\ell)^\alpha} \geq \frac{p(\ell')}{d(\ell')^\alpha}$, AND (2) $\frac{p(\ell)}{d(\ell)^\alpha} \geq 2\beta N$.

(*) LOWER BOUNDS ON SCHEDULE LENGTH ARE BASED ON GEOMETRY AND INTERFERENCE, REGARDLESS OF ALGORITHMS.

K-V closest work, for SINR acks

Contribution

- **We study Distributed Wireless Link Scheduling (DWLS) protocols that run under arbitrary interference.**
- **We present a novel combinatorial structure of polynomial size that guarantees that every request is realized.**
- **We present 3 DWLS protocols that trade schedule length for min-entropy.**
- **We present an affectance characteristic that takes into account acknowledgments' implementation.**

Our Results

Matches our new lower bound up to polylog

Same as K-V upper bound

	Schedule length	Min-entropy per request
Deterministic	$O(\min\{\mathbb{A}^2 \log^3 n, n\})$	0
Randomized	$O(\mathbb{A} \log n)$	$O(\log \mathbb{A} \log n)$
Parameterized	$O(\min\{(\mathbb{A}^2/W)\log^3 n, n\})$	$O(\log W \log n)$

but K-V has
$O(\overline{A} \log \overline{A} \log n)$
min-entropy

$W \leq \mathbb{A}$
$\mathbb{A} = A(L) + A(L^*)$
L^*: set of reversed requests

Deterministic DWLS

- **Algorithmic core: combinatorial structure we call $(n, \mathscr{A})$-Affectance-Direct-Link-Scheduler (AFF-DLS):**

For affectance threshold $\mathscr{A}$, *an* $(n, \mathscr{A})$*-AFF-DLS is*

a family of subsets $S_1, S_2, \ldots, S_\tau \subseteq L$ *such that*

for every request $(v_i, v_j) \in L$ *such that* $\sum_{(v_x,v_y)\in L} a((v_x, v_y), (v_i, v_j)) \leq \mathscr{A}$,

there exists $t \leq \tau$ *such that* $\sum_{(v_x,v_y)\in S_t} a((v_x, v_y), (v_i, v_j)) \leq 1$.

- **We show how each node can construct locally an AFF-DLS of length $4\mathscr{A}^2\lceil \log_{\mathscr{A}} n \rceil^2$ in poly time.**

Deterministic DWLS

<u>**In a nutshell**</u>**:**

For each $i = 1, 2, \ldots$ **until realized**

For $\log n$ **times**

Use a $(n, 2^i)$**-Aff-DLS to decide when to transmit**

If acknowledgement is received

Stop

Deterministic DWLS

__In a nutshell__:

For each $i = 1,2,\dots$ until realized

For $\log n$ times

Use a $(n,2^i)$-Aff-DLS to decide when to transmit

If acknowledgement is received

Stop

Algorithm 1: Deterministic DWLS algorithm for each request (s,r). Given locally pre-computed $(n,2^i)$-AFF-DLS, for $i = 1,\dots,\frac{1}{2}\log\frac{n}{\log^2 n}$, as in Corollary 2. S_t denotes t-th set in current $(n,2^i)$-AFF-DLS.

s gets active, r gets passive
for $i = 1,2\dots,\frac{1}{2}\log\frac{n}{\log^2 n}$ **do**
  /* Phase i: */
  for $j = 1,2\dots,\log n$ **do**
    /* Sub-phase j of phase i: */
    /* Part 1: packets */
    for $t = 1,2,\dots,length[(n,2^i)\text{-}AFF\text{-}DLS]$ **do**
      if s *is* active *and* $s \in S_t$ **then**
        s transmits *packet* to r
      if r *not* active *and gets* packet *from* s **then**
        r becomes active
    /* Part 2: acknowledgments */
    for $t = 1,2,\dots,length[(n,2^i)\text{-}AFF\text{-}DLS]$ **do**
      if r *is* active *and* $r \in S_t$ **then**
        r transmits *acknowledgement* to s
      if s *receives* acknowledgment *from* r **then**
        s gets acknowledged
    /* Part 3: successful stops */
    for $t = 1,2,\dots,length[(n,2^i)\text{-}AFF\text{-}DLS]$ **do**
      if s *is* acknowledged *and* $s \in S_t$ **then**
        s transmits *stop* to r
      if r *receives* stop *from* s **then**
        r stops
    if s *is* acknowledged **then**
      s stops
    r becomes passive

Deterministic DWLS

<u>In a nutshell</u>:

For each $i = 1,2,\ldots$ until realized

For $\log n$ times

Use a $(n,2^i)$-Aff-DLS to decide when to transmit

If acknowledgement is received

Stop

<u>Performance</u>:

- $O(\min\{\mathbb{A}^2 \log^3 n, n\})$ **rounds**
- **Min-entropy:** 0

Algorithm 1: Deterministic DWLS algorithm for each request (s,r). Given locally pre-computed $(n,2^i)$-AFF-DLS, for $i = 1,\ldots,\frac{1}{2}\log\frac{n}{\log^2 n}$, as in Corollary 2. S_t denotes t-th set in current $(n,2^i)$-AFF-DLS.

s gets active, r gets passive
for $i = 1,2\ldots,\frac{1}{2}\log\frac{n}{\log^2 n}$ **do**
  /* Phase i: */
  for $j = 1,2\ldots,\log n$ **do**
    /* Sub-phase j of phase i: */
    /* Part 1: packets */
    for $t = 1,2,\ldots,$ *length*$[(n,2^i)$-*AFF-DLS*$]$ **do**
      if s *is* active *and* $s \in S_t$ **then**
        s transmits *packet* to r
      if r *not* active *and gets* packet *from* s **then**
        r becomes active
    /* Part 2: acknowledgments */
    for $t = 1,2,\ldots,$ *length*$[(n,2^i)$-*AFF-DLS*$]$ **do**
      if r *is* active *and* $r \in S_t$ **then**
        r transmits *acknowledgement* to s
      if s *receives* acknowledgment *from* r **then**
        s gets acknowledged
    /* Part 3: successful stops */
    for $t = 1,2,\ldots,$ *length*$[(n,2^i)$-*AFF-DLS*$]$ **do**
      if s *is* acknowledged *and* $s \in S_t$ **then**
        s transmits *stop* to r
      if r *receives* stop *from* s **then**
        r stops
    if s *is* acknowledged **then**
      s stops
    r becomes passive

Randomized DWLS

In a nutshell (acks and $\mathbb{A}$ given for clarity):

For each window of $W \geq \mathbb{A}$ rounds

Choose uniformly at random a round to transmit

If acknowledgement is received

Stop

Randomized DWLS

<u>In a nutshell (acks and $\mathbb{A}$ given for clarity)</u>:

For each window of $W \geq \mathbb{A}$ rounds

Choose uniformly at random a round to transmit

If acknowledgement is received

Stop

Algorithm 2: Randomized DWLS algorithm for each request (v, w). The window size W is a parameter.

/* Algorithm for sender v */

$i \leftarrow 0$

$\delta \leftarrow$ integer chosen in $[1, W]$ uniformly at random

for *each round* $t = 1, 2, \ldots$ **do**

  if $t = iW + \delta$ **then**

    transmit to w

    if *acknowledgement is received from* w **then** stop

  if $t \equiv 0 \mod W$ **then**

    $i{+}{+}$

    $\delta \leftarrow$ integer chosen in $[1, W]$ uniformly at random

/* Algorithm for receiver w */

for *each round* $t = 1, 2, \ldots$ **do**

  if *transmission from* v *is received* **then**

    transmit acknowledgement to v

    stop

Acks given and $\mathbb{A}$ known for clarity.

Randomized DWLS

In a nutshell (acks and $\mathbb{A}$ given for clarity):

For each window of $W \geq \mathbb{A}$ rounds

Choose uniformly at random a round to transmit

If acknowledgement is received

Stop

Performance: whp

- $O(\mathbb{A} \log n)$ rounds
- Min-entropy: $O(\log \mathbb{A} \log n)$

Algorithm 2: Randomized DWLS algorithm for each request (v, w). The window size W is a parameter.

/* Algorithm for sender v */

$i \leftarrow 0$

$\delta \leftarrow$ integer chosen in $[1, W]$ uniformly at random

for *each round* $t = 1, 2, \ldots$ **do**

  if $t = iW + \delta$ **then**

    transmit to w

    if *acknowledgement is received from* w **then**

      stop

  if $t \equiv 0 \mod W$ **then**

    $i{+}{+}$

    $\delta \leftarrow$ integer chosen in $[1, W]$ uniformly at random

/* Algorithm for receiver w */

for *each round* $t = 1, 2, \ldots$ **do**

  if *transmission from* v *is received* **then**

    transmit acknowledgement to v

    stop

Acks given and $\mathbb{A}$ known for clarity.

Trading Time for Min-entropy

<u>In a nutshell</u>: Consider windows composed of $W \leq \mathbb{A}$ sub-windows.

Each sub-window composed of W' rounds.

For each window

- Choose uniformly at random a sub-window
- Use a $(n, \mathscr{A})$-Aff-DLS of length W' to decide when to transmit
- If acknowledgement is received stop

Trading Time for Min-entropy

<u>In a nutshell</u>: Consider windows composed of $W \leq \mathbb{A}$ sub-windows.

Each sub-window composed of W' rounds.

For each window

  Choose uniformly at random a sub-window

  Use a $(n, \mathscr{A})$-Aff-DLS of length W' to decide when to transmit

  If acknowledgement is received stop

<u>Performance</u>: whp

- $O(\min\{(\mathbb{A}^2/W)\log^3 n, n\})$ rounds
- Min-entropy: $O(\log W \log n)$

where $W \leq \mathbb{A}$

Open Problems

Matches our new lower bound up to polylog

Same as K-V upper bound

	Schedule length	Min-entropy per request
Deterministic	$O(\min\{\mathbb{A}^2 \log^3 n, n\})$	0
Randomized	$O(\mathbb{A} \log n)$	$O(\log \mathbb{A} \log n)$
Parameterized	$O(\min\{(\mathbb{A}^2/W)\log^3 n, n\})$	$O(\log W \log n)$

but K-V has $O(\bar{A} \log \bar{A} \log n)$ min-entropy

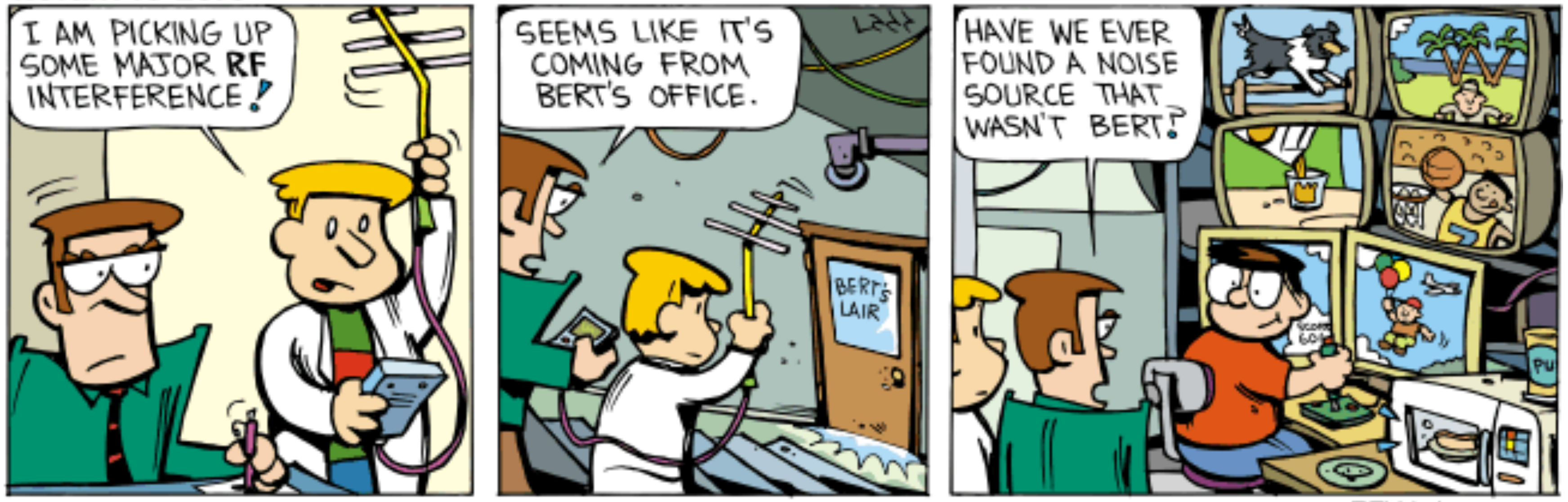
$W \leq \mathbb{A}$
$\mathbb{A} = A(L) + A(L^*)$
L^*: set of reversed requests

- reduce polylog factors?
- time-entropy lower bounds?

Thank you!

Miguel A. Mosteiro
Pace University
mmosteiro@pace.edu

Return to Zero



EEWeb.com