Engineering Emergent Behavior in Mobile Ad Hoc Networks

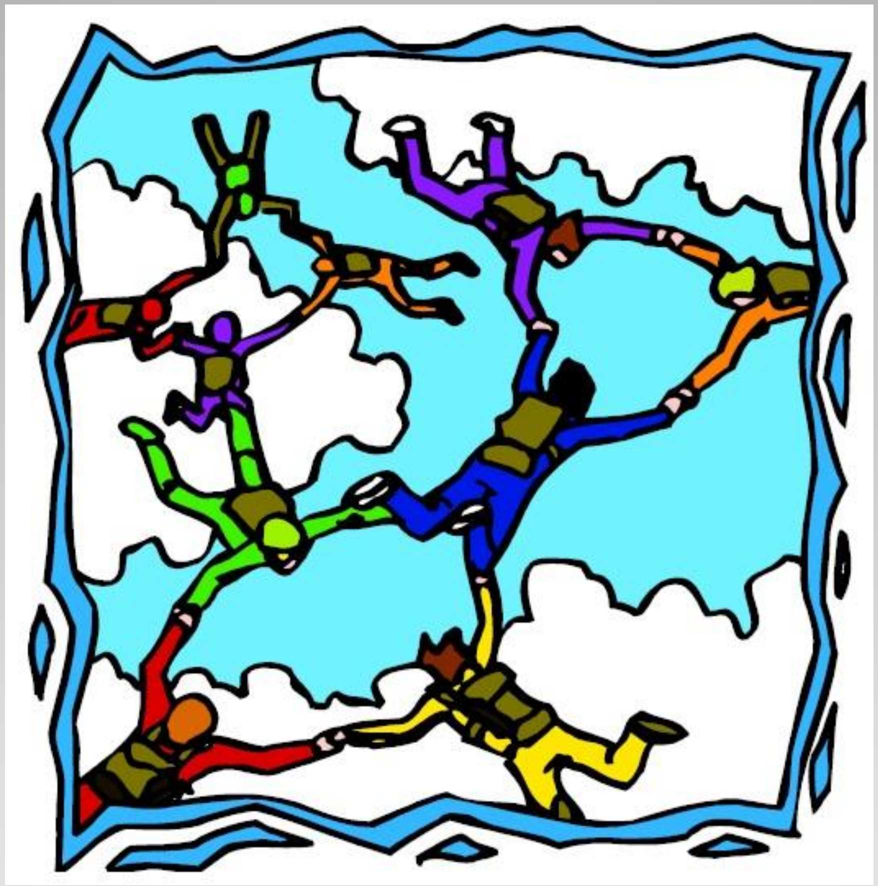
Marco A. Alzate
Universidad Distrital Francisco José de Caldas

- Introduction to Mobile Ad Hoc Networks
- Introduction to Complex Adaptive Systems
- Emergent Synchronization in MANETs
- Emergent Cooperation in MANETs
- Formal methods for engineering emergent behavior in MANETs

Schedule

- **Introduction to Mobile Ad Hoc Networks**
- Introduction to Complex Adaptive Systems
- Emergent Synchronization in MANETs
- Emergent Cooperation in MANETs
- Formal methods for engineering emergent behavior in MANETs

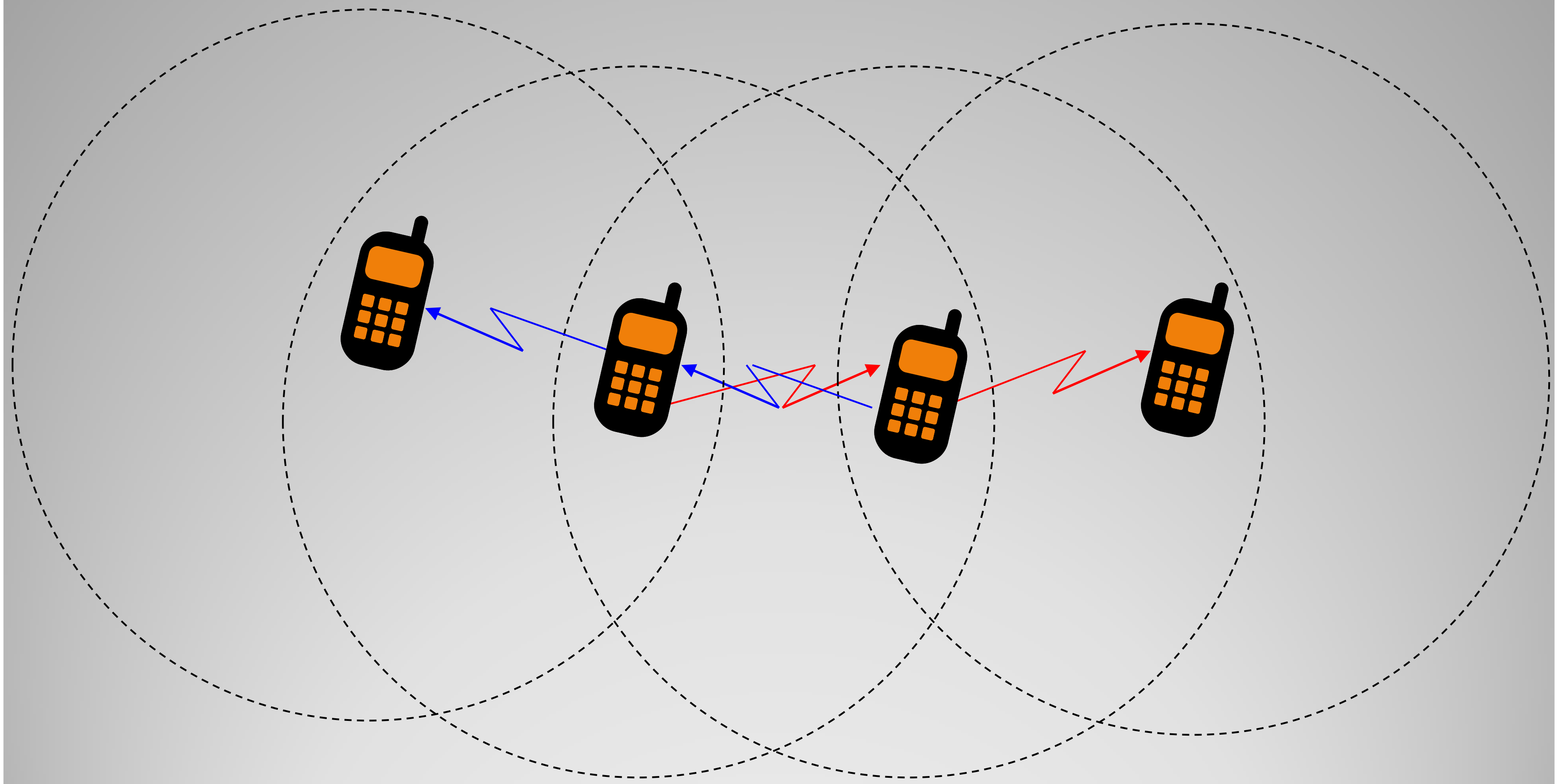
Schedule



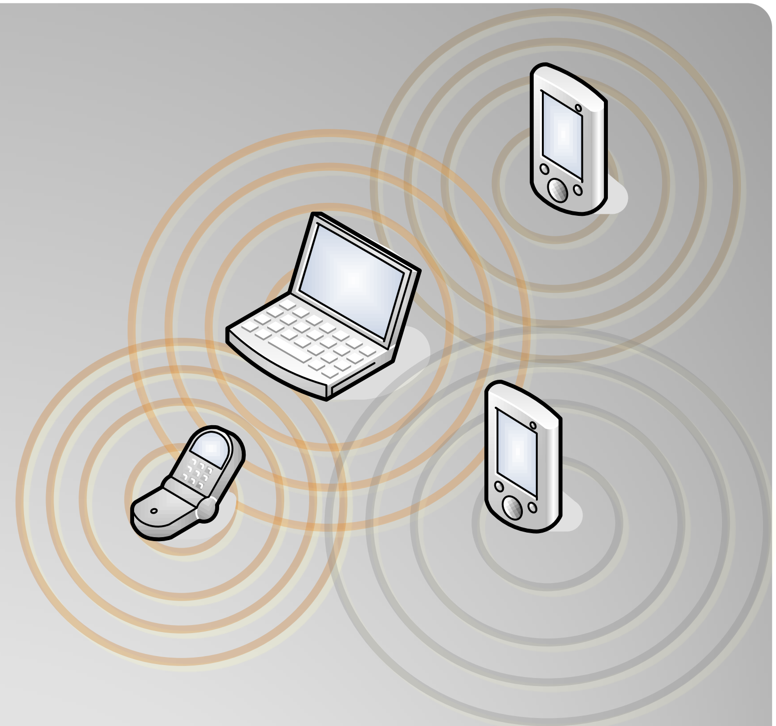
Introduction to Mobile Ad Hoc Networks

ideas
Investigación, Desarrollo y Aplicaciones en Señales

WEA

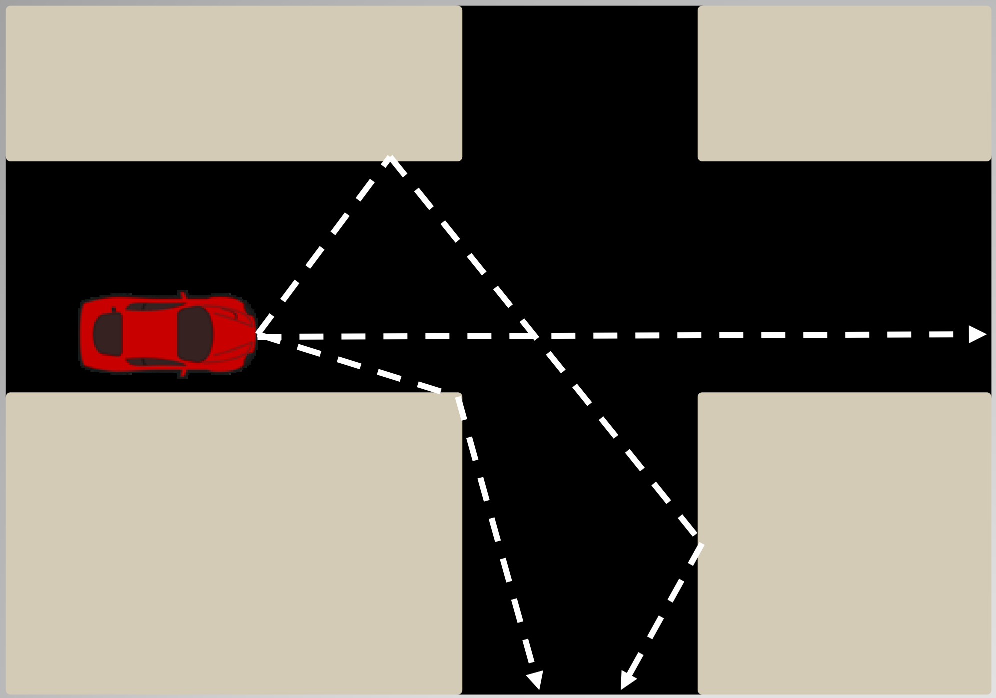


Mobile Ad Hoc Networks



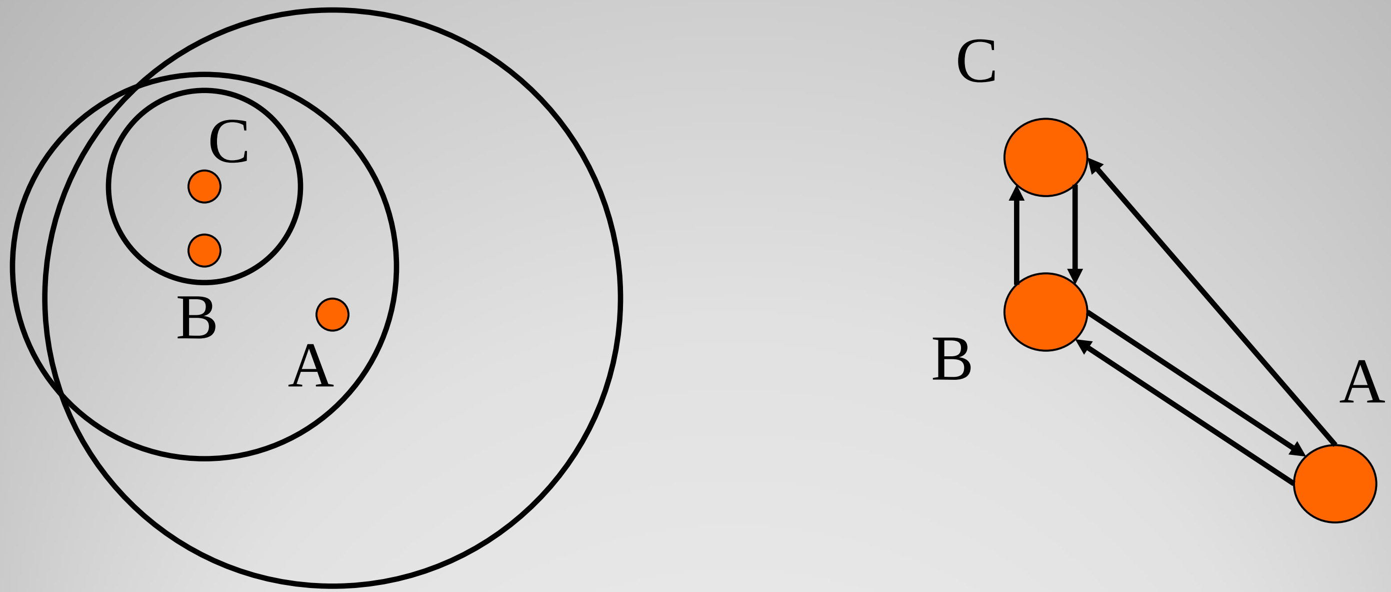
- Lack of infrastructure
- Wireless transmission
- Constant topological changes
- Mobile devices with limited resources

MANET Characteristics

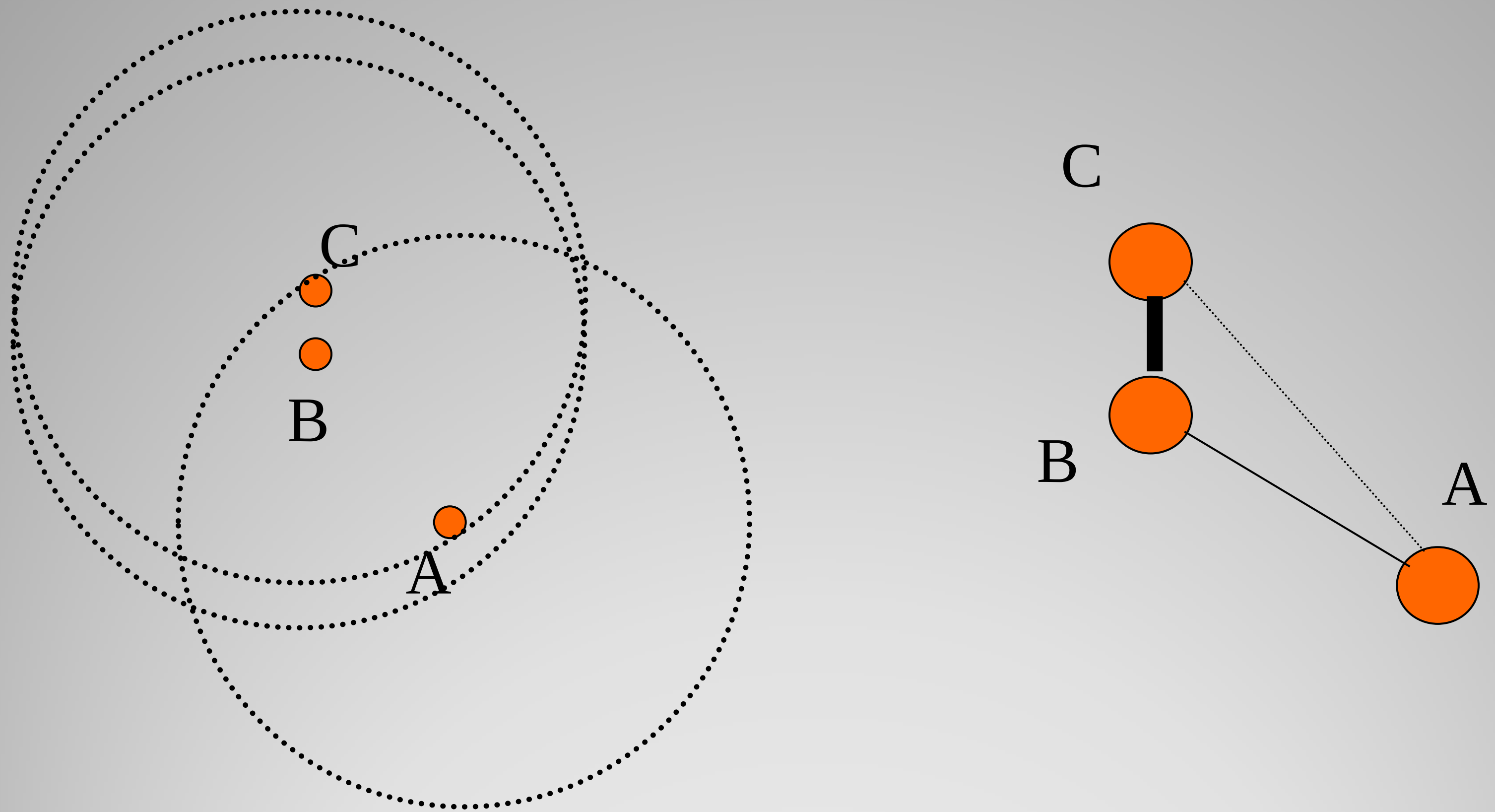


- Multipath
- Fading
- Attenuation
- Noise
- Interference
- Limited Power
- Limited Bandwidth

Wireless transmission

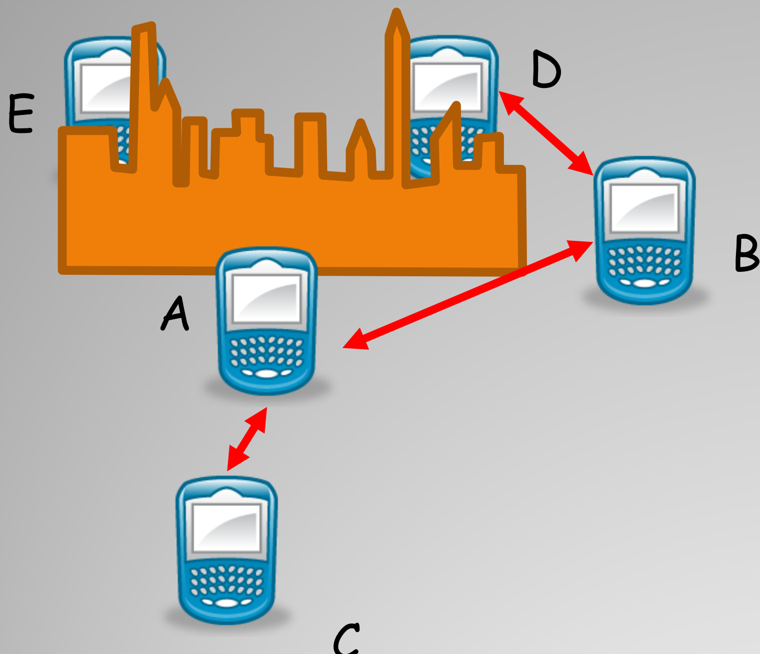


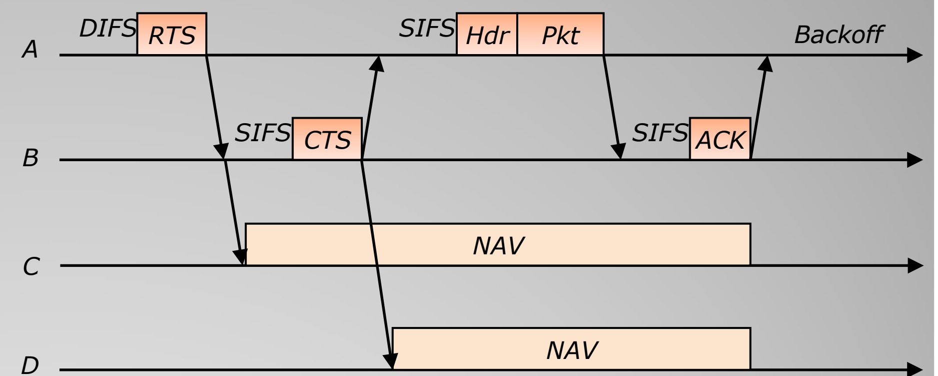
Some links can be unidirectional



Quality of links can change significantly

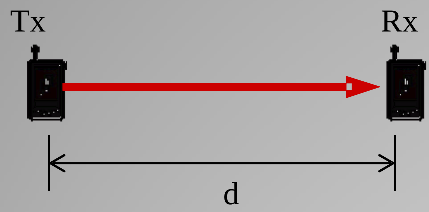
Interference

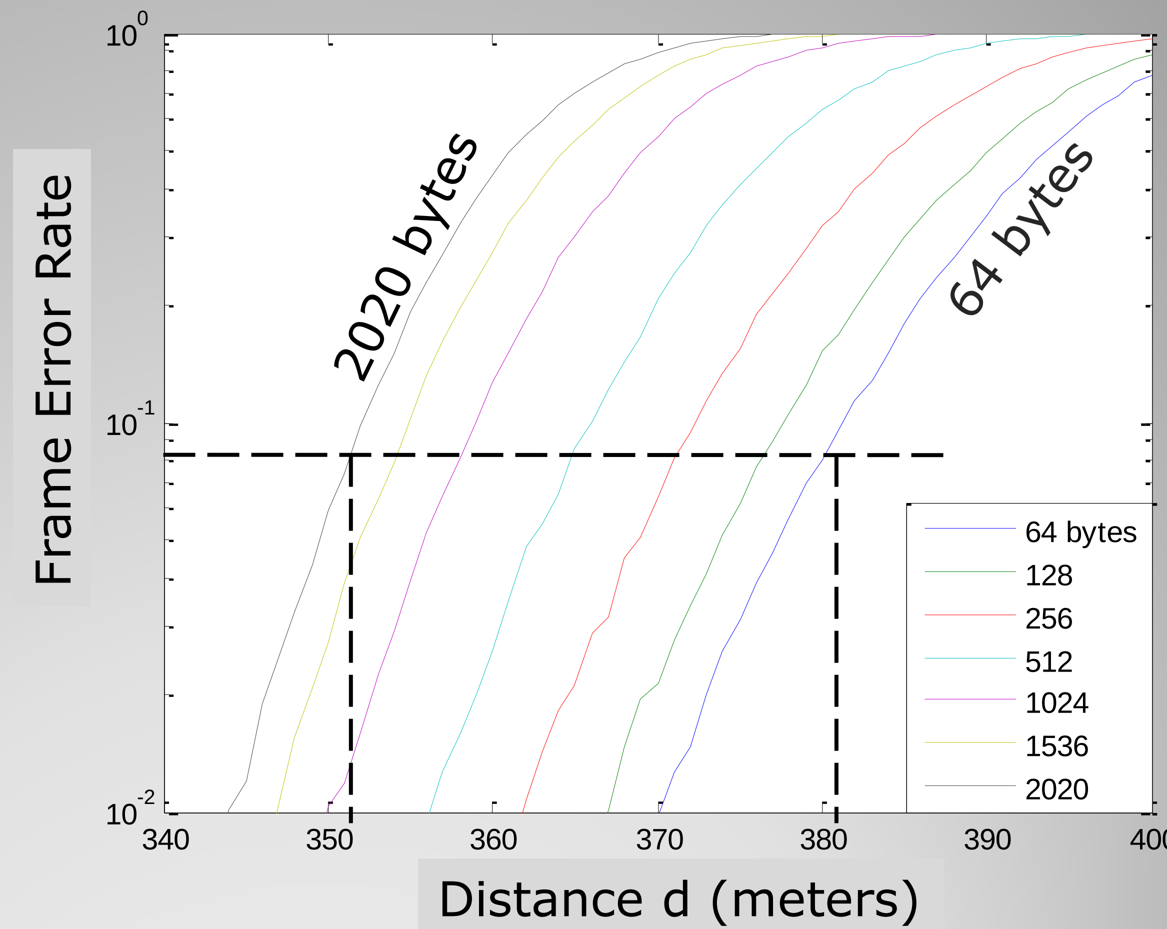




CSMA/CA (IEEE 802.11)

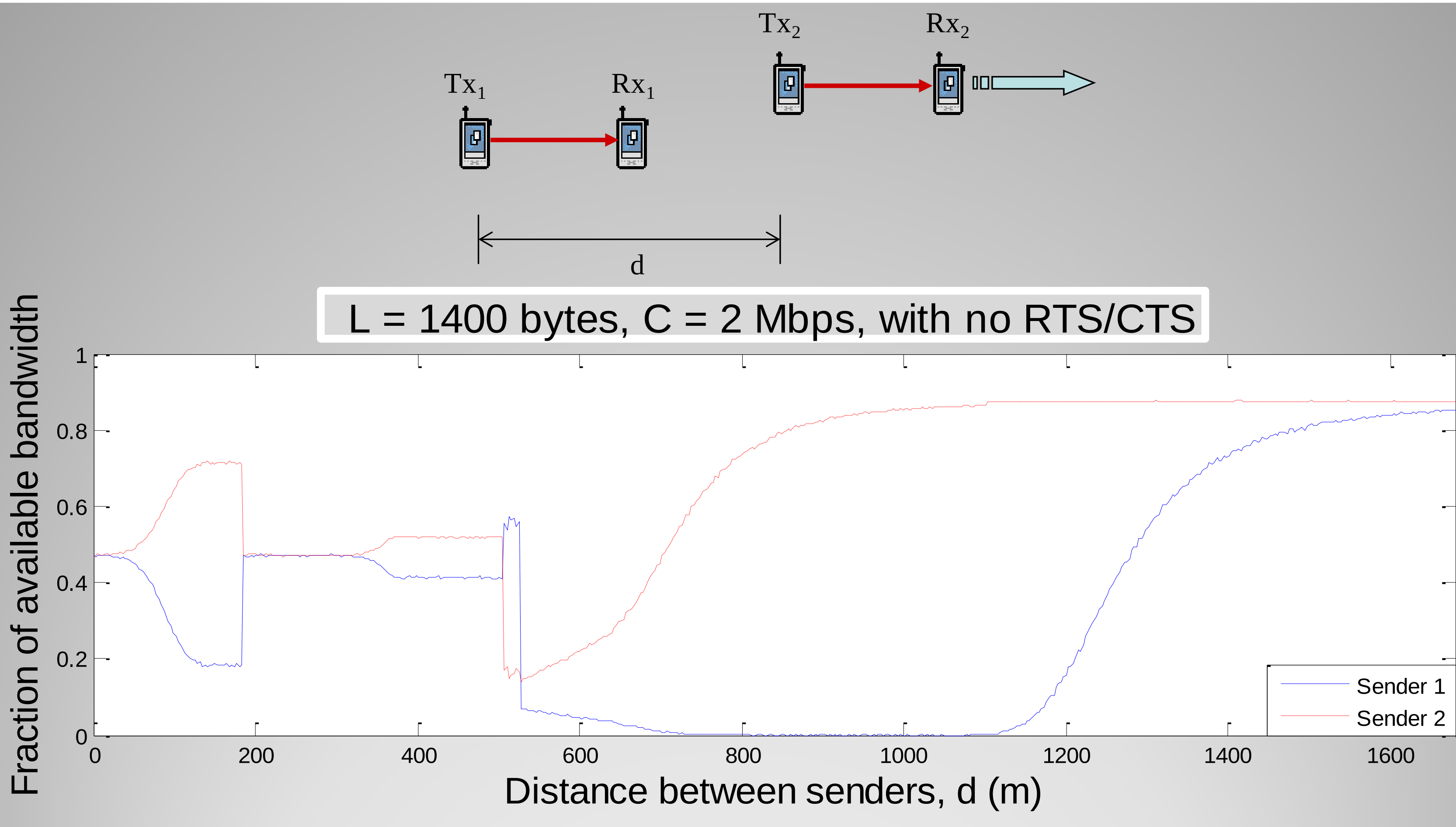
- ☑ Hidden Terminal
- ☒ Exposed Terminal





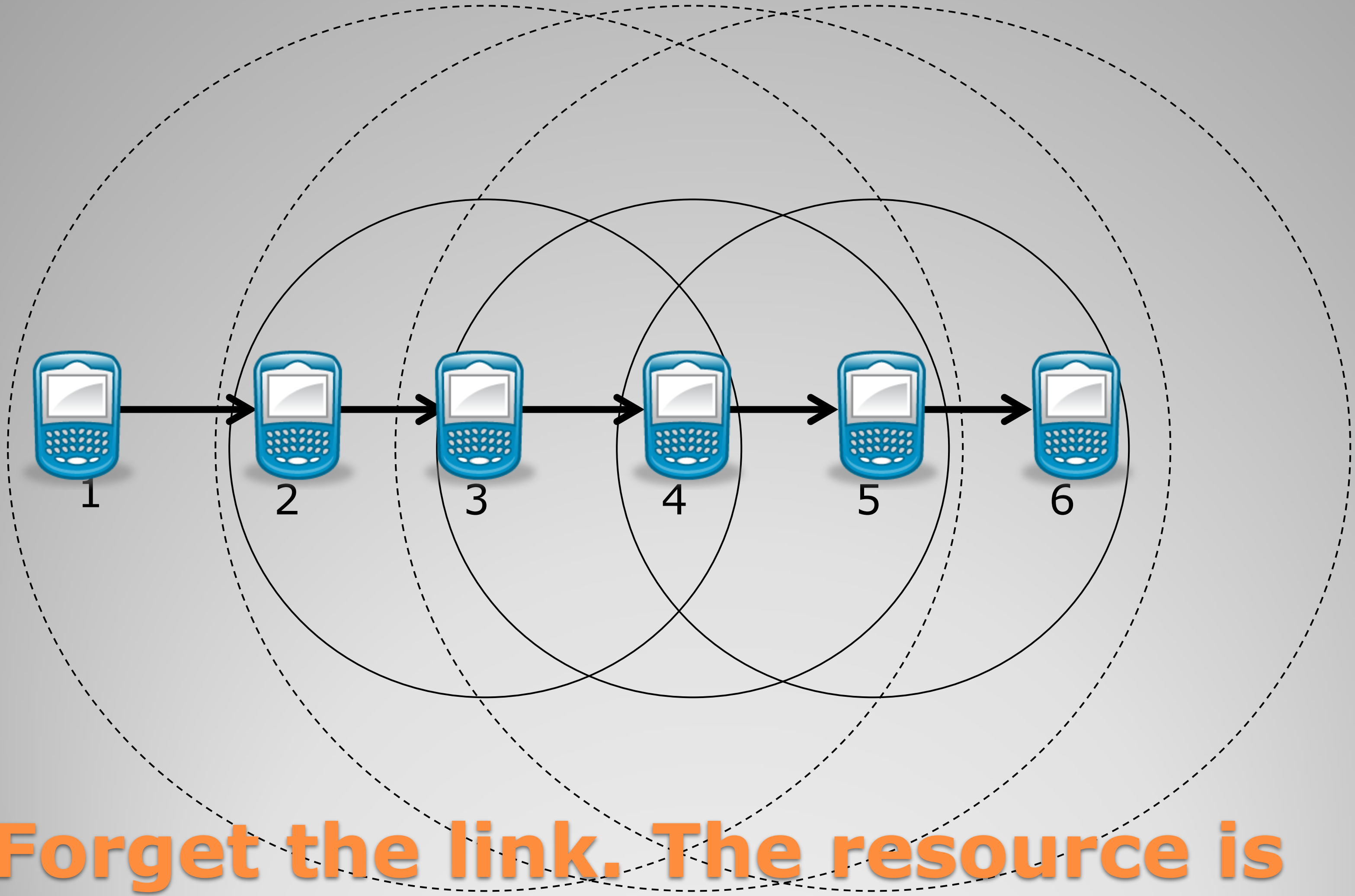
[Alzate, 2008]

What is a link in a MANET?



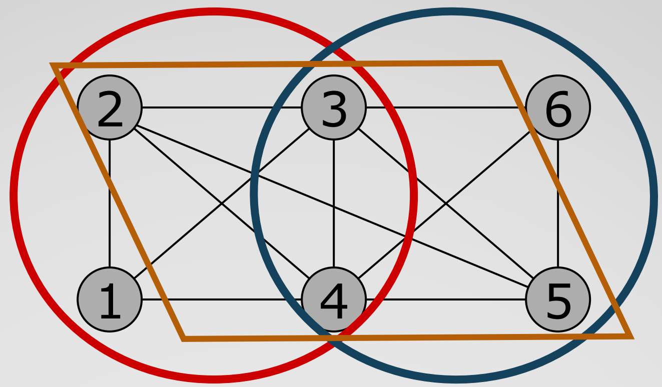
[Alzate, 2008]

What is a link in a MANET?



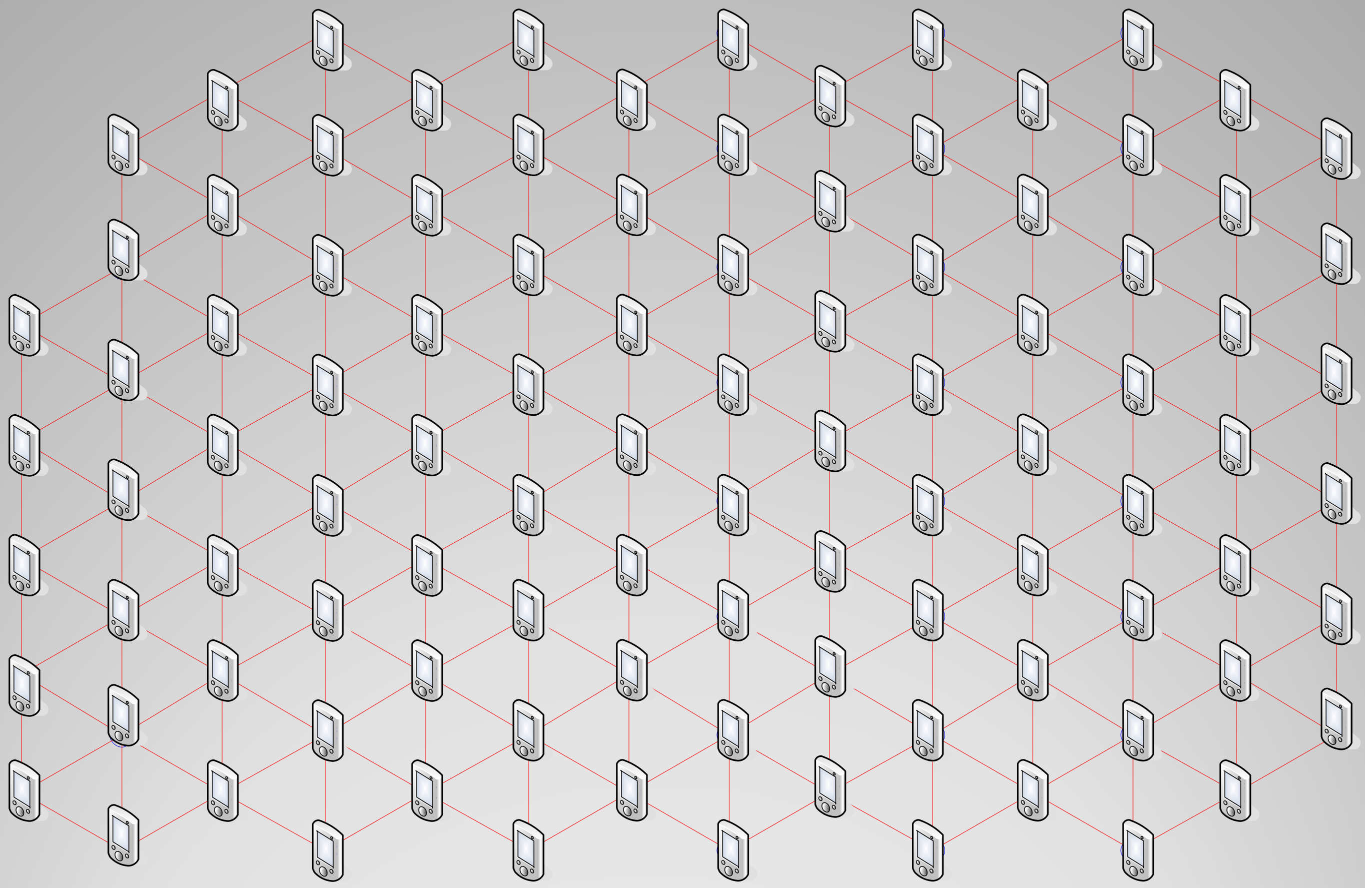
Forget the link. The resource is the "spatial channel"

Contention Graph
Spatial Channel



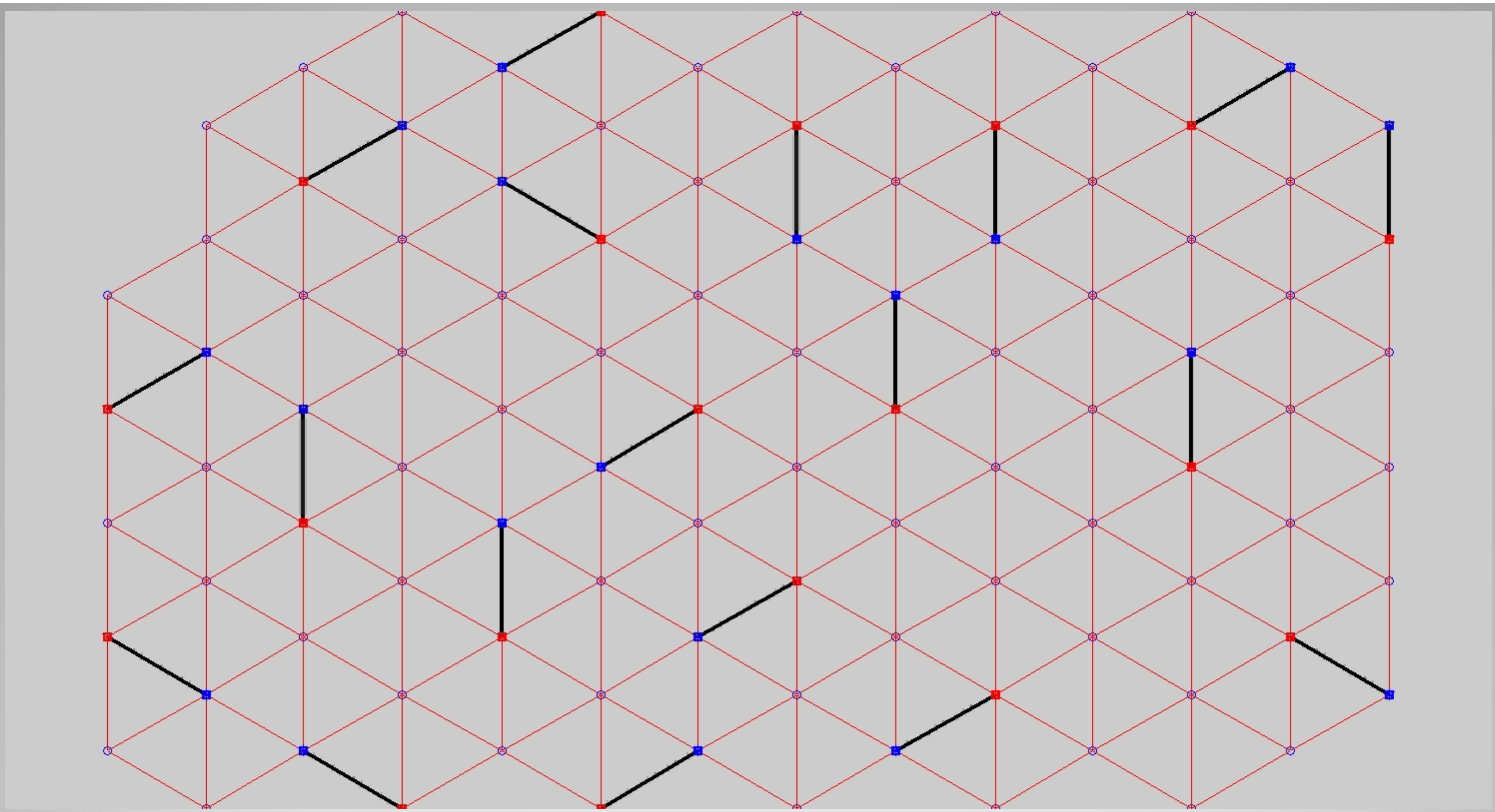
Spatial Channel: Maximal clique in the contention graph (Completely interconnected subgraph not contained within another completely interconnected subgraph)

[Alzate, 2011]



[Puerta, Aguirre, Alzate, 2010]

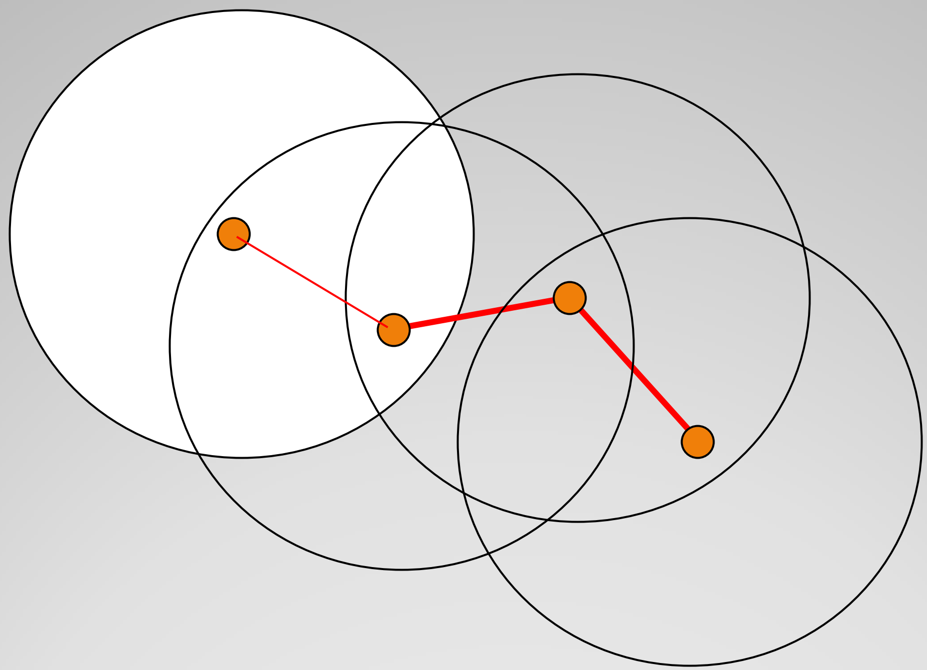
How can these links be used efficiently?



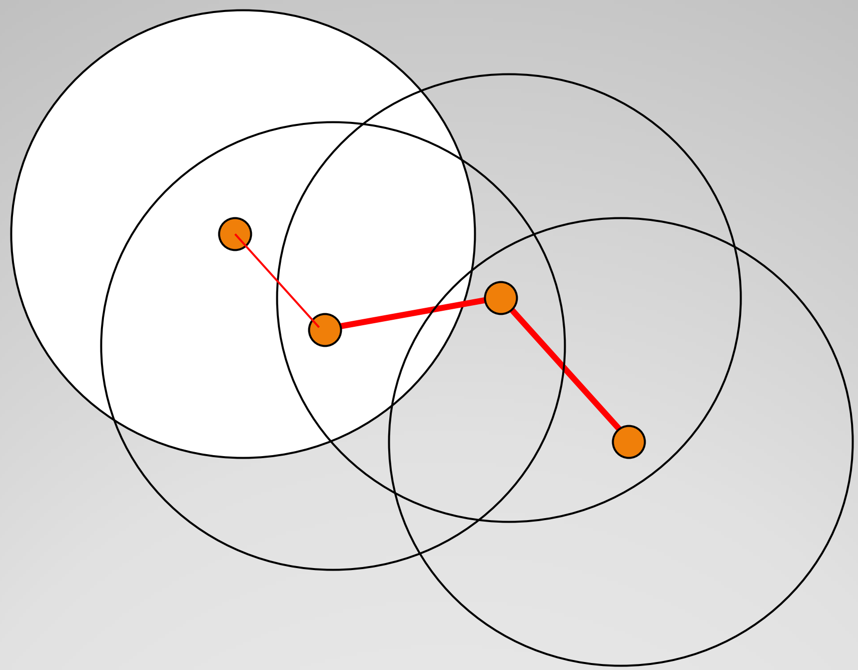
[Puerta, Aguirre, Alzate, 2010]

By Perfect scheduling…
Can they learn such scheduling?

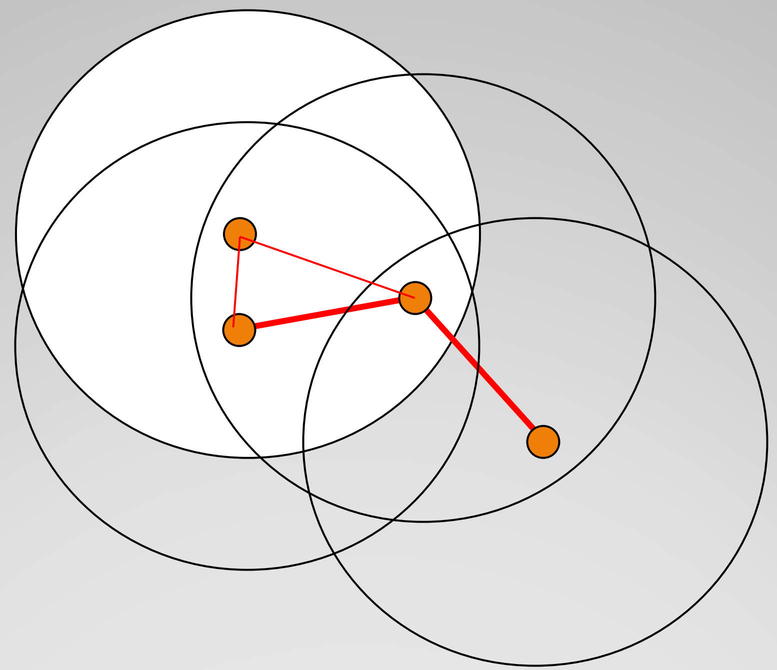
ideas
Investigación, Desarrollo y Aplicaciones en Señales



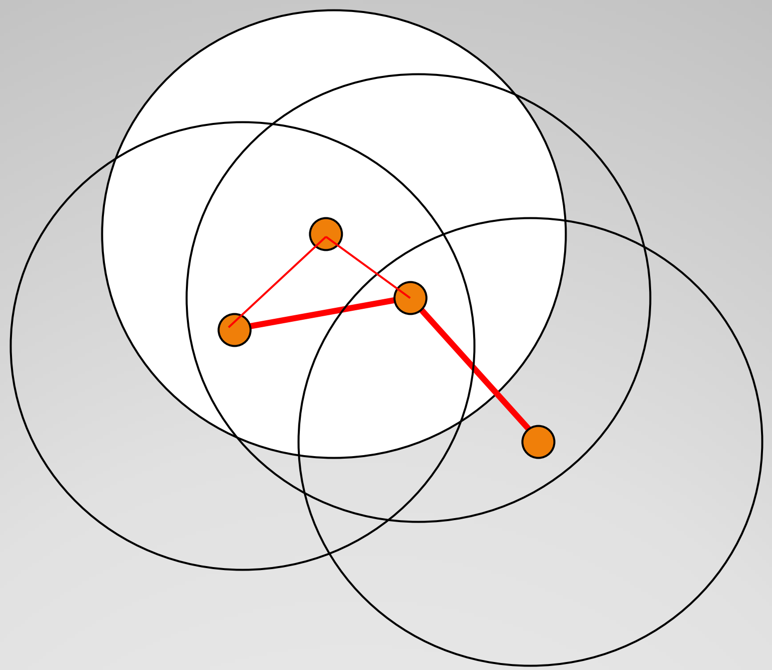
But… Mobility!



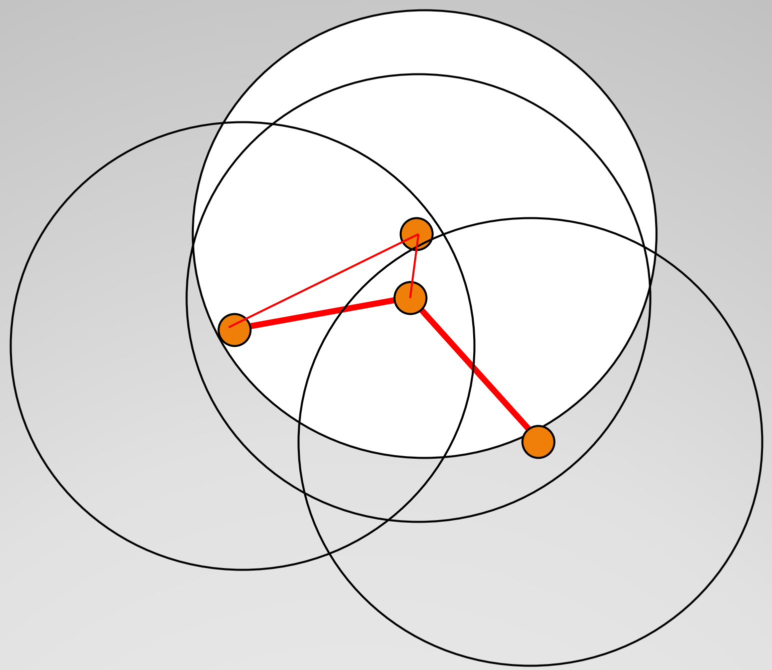
But... Mobility!



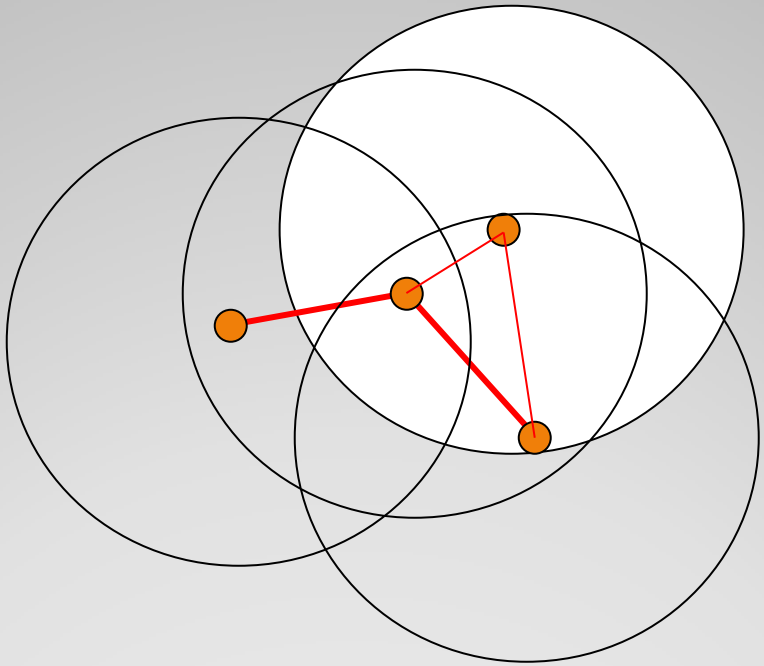
But… Mobility!



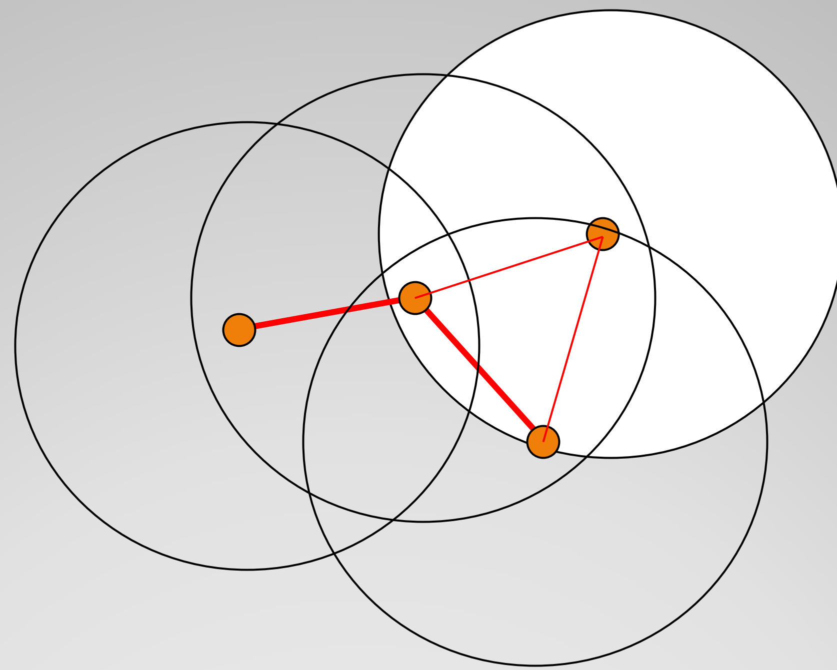
But… Mobility!



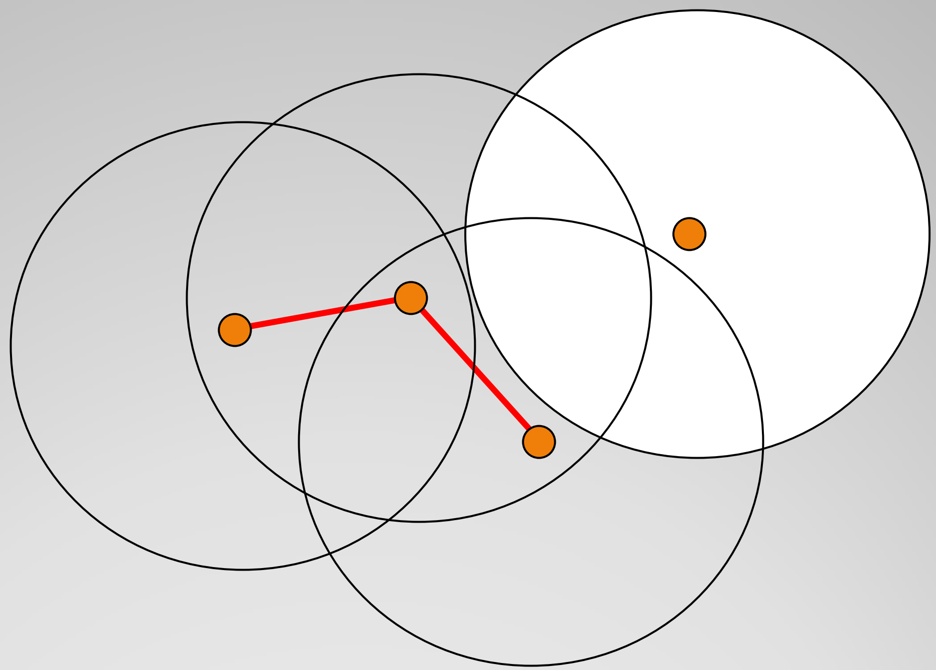
But… Mobility!



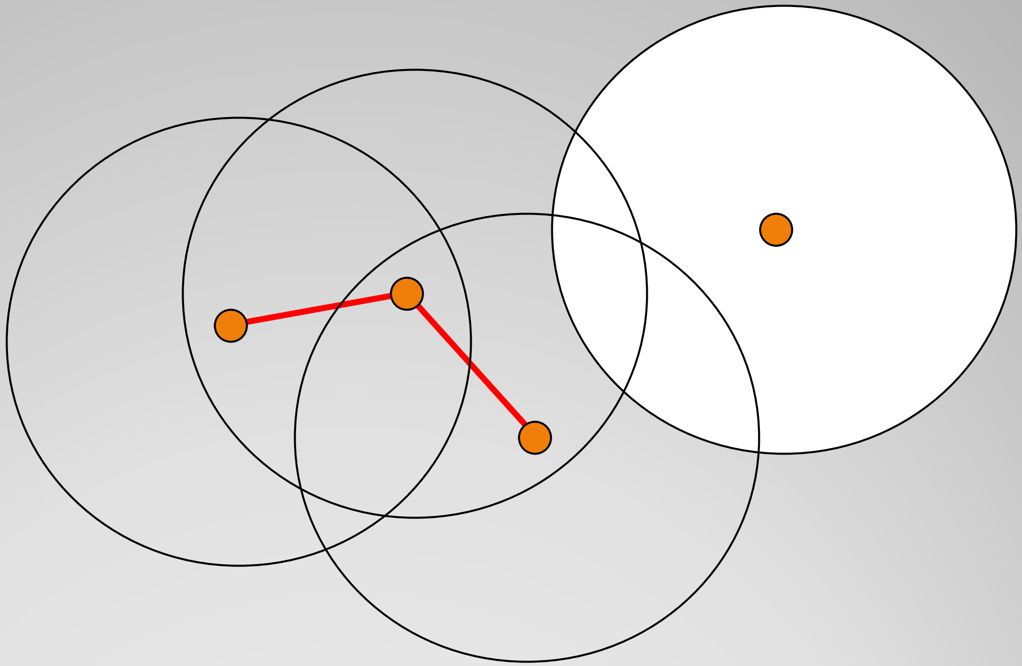
But… Mobility!



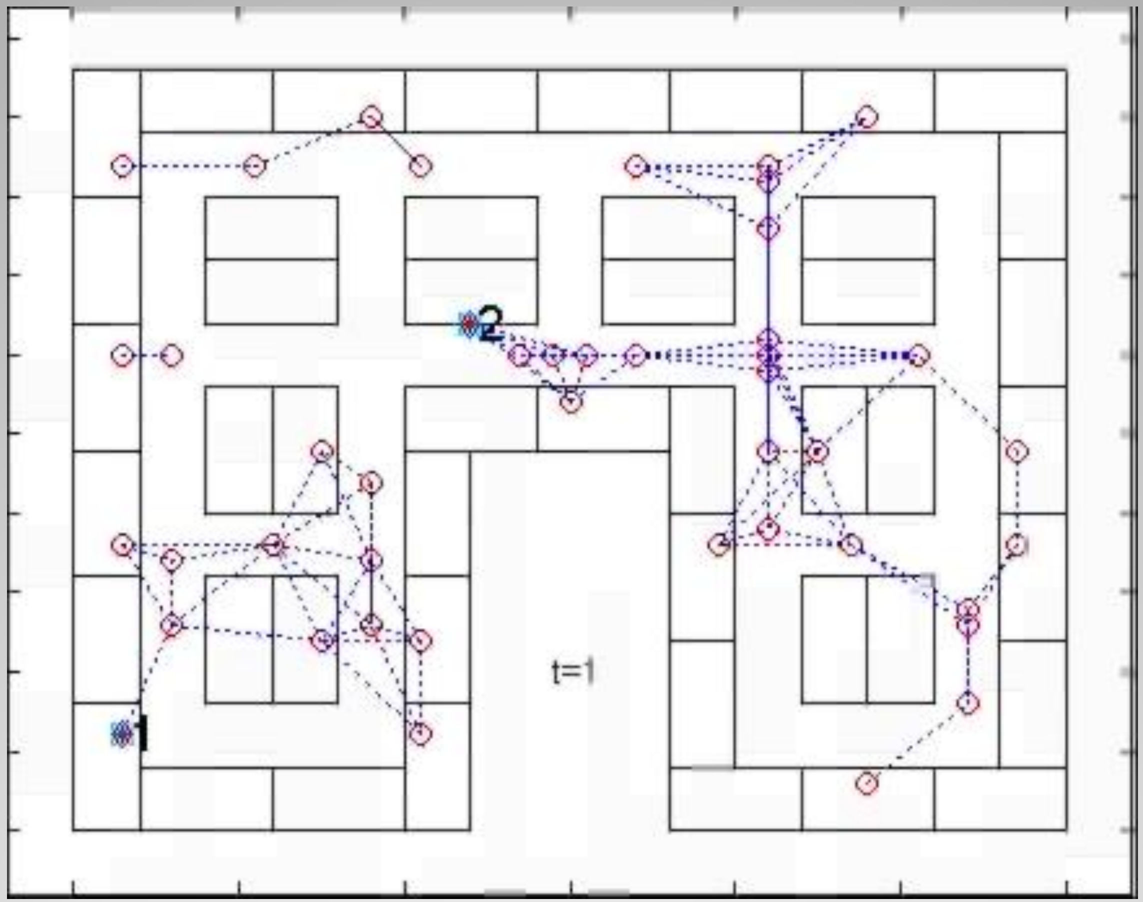
But… Mobility!



But… Mobility!

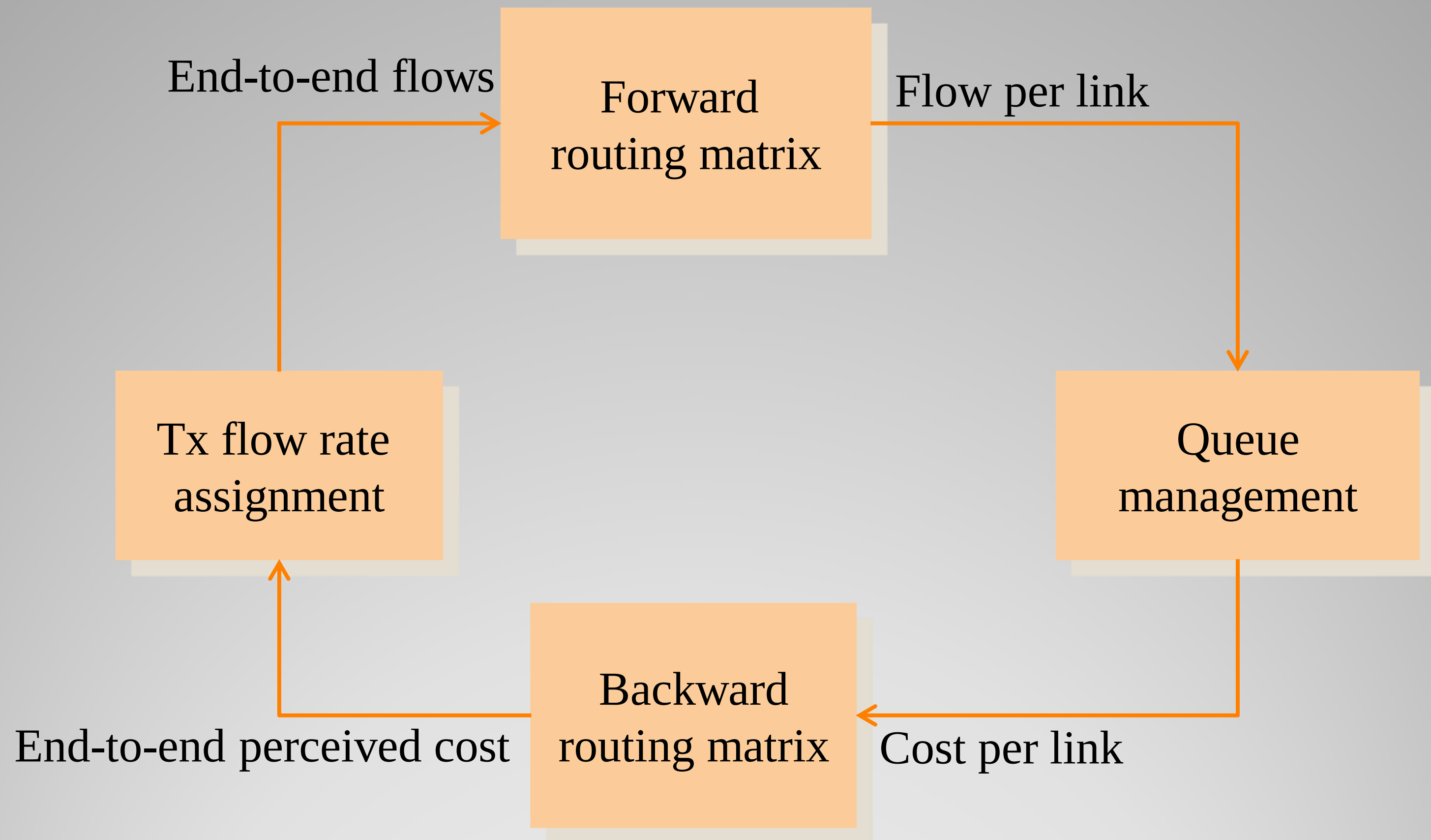


But… Mobility!

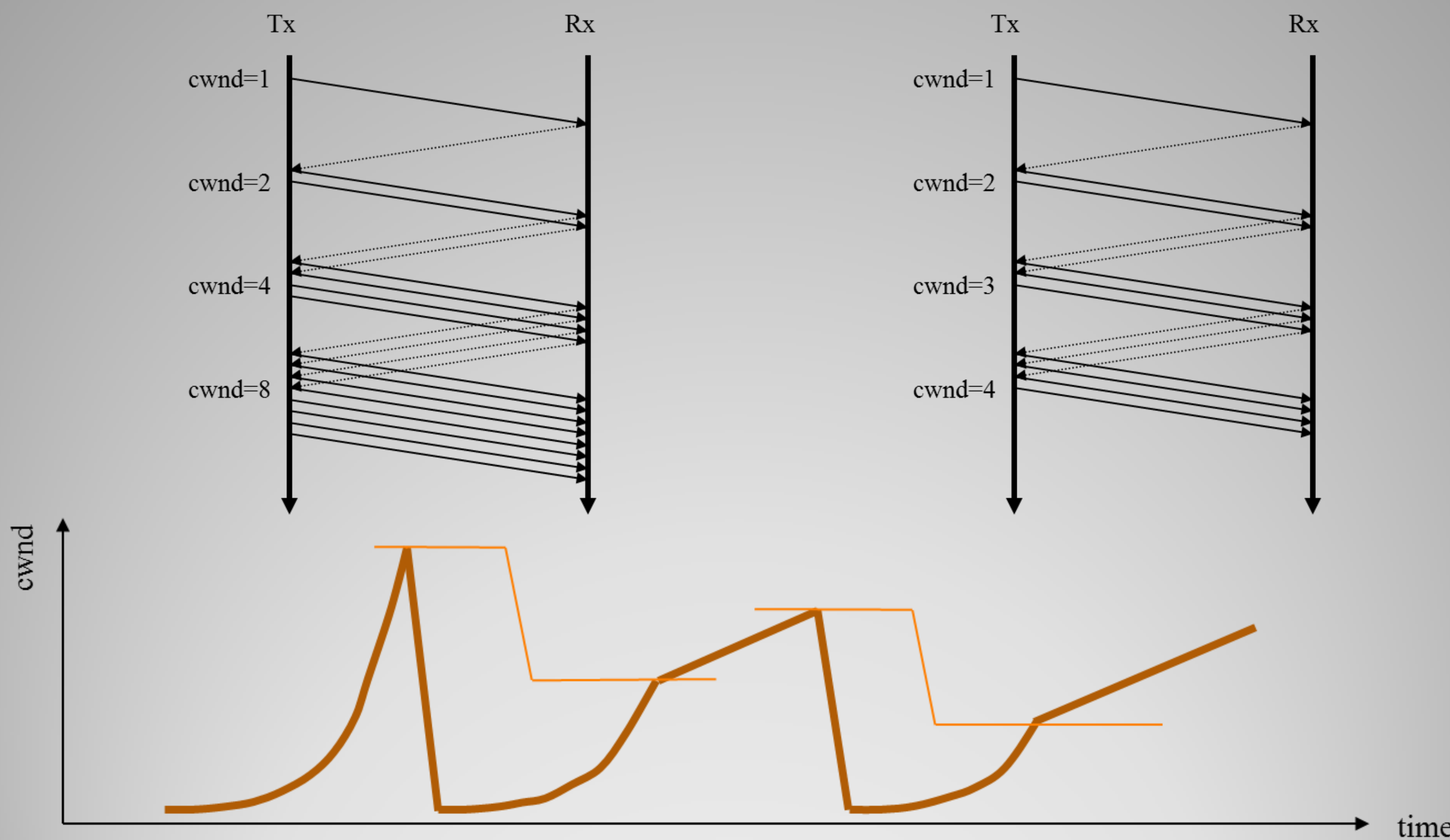


[Alzate, Baras, 2004]

Link Interaction in a MANET

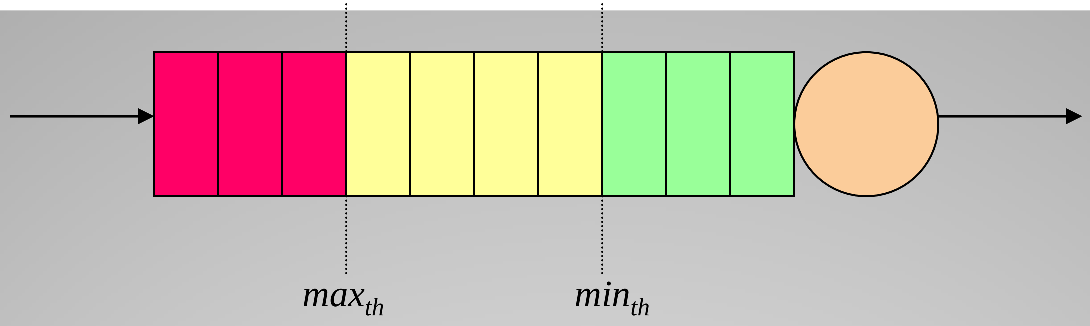


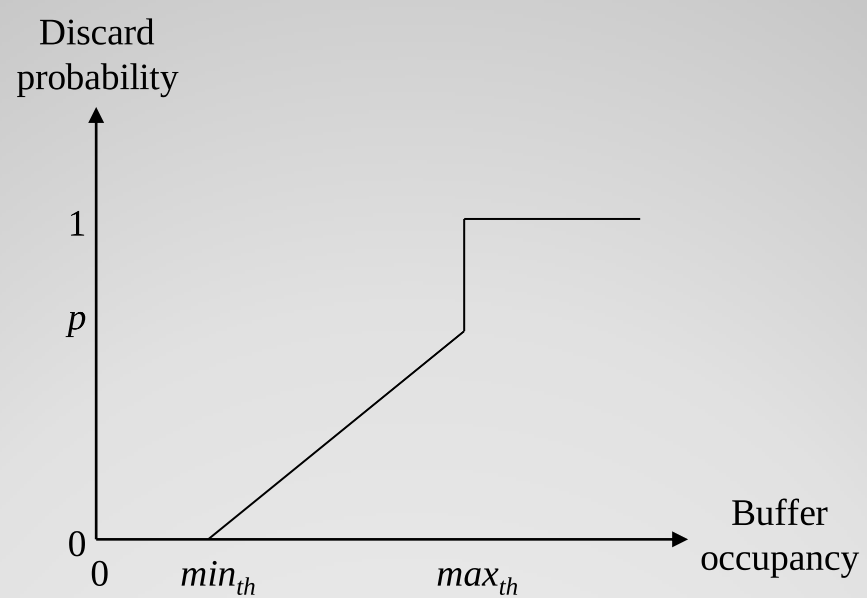
Many other unintended interactions

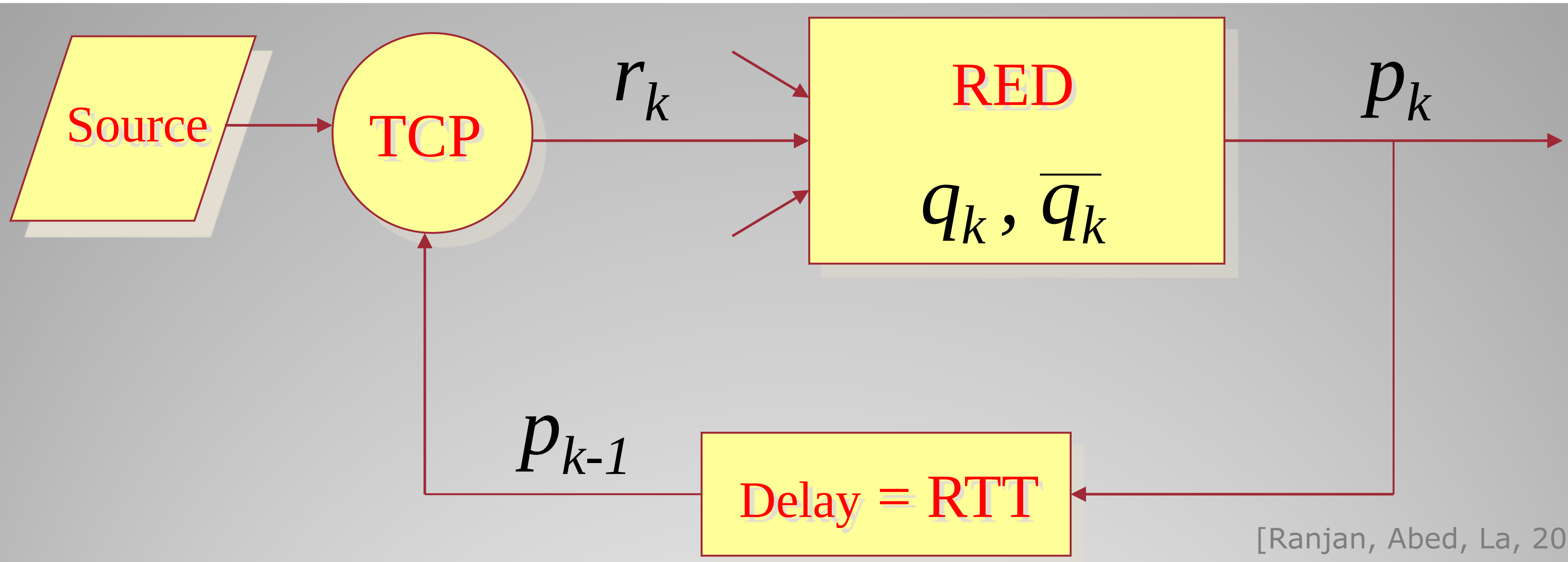


Example: TCP/IP on a wired network

RED: Random Early Detection







[Ranjan, Abed, La, 2002]

TCP: $$r_k = \frac{M}{RTT}\frac{K}{\sqrt{p_{k-1}}}$$

M: packet Size
RTT: Round Trip Time
p_k : discard probability
B: Buffer size
n : Number of TCP flows
R_0 : Minimum RTT
C : Link Capacity
K : Constant (1.25)

RED: $$q_k = \left(\frac{nr_k}{M}RTT - \frac{C}{M}R_0, B, 0\right)$$

$$\bar{q}_k = (1-w)\bar{q}_{k-1} + wq_k$$

$$p_k = \begin{cases} 0 & 0 \le \bar{q}_k < \min_{th} \\ \dfrac{\bar{q}_k - \min_{th}}{\max_{th} - \min_{th}} p_{\max} & \min_{th} \le \bar{q}_k < \max_{th} \\ 1 & \max_{th} \le \bar{q}_k \le B \end{cases}$$

TCP-RED

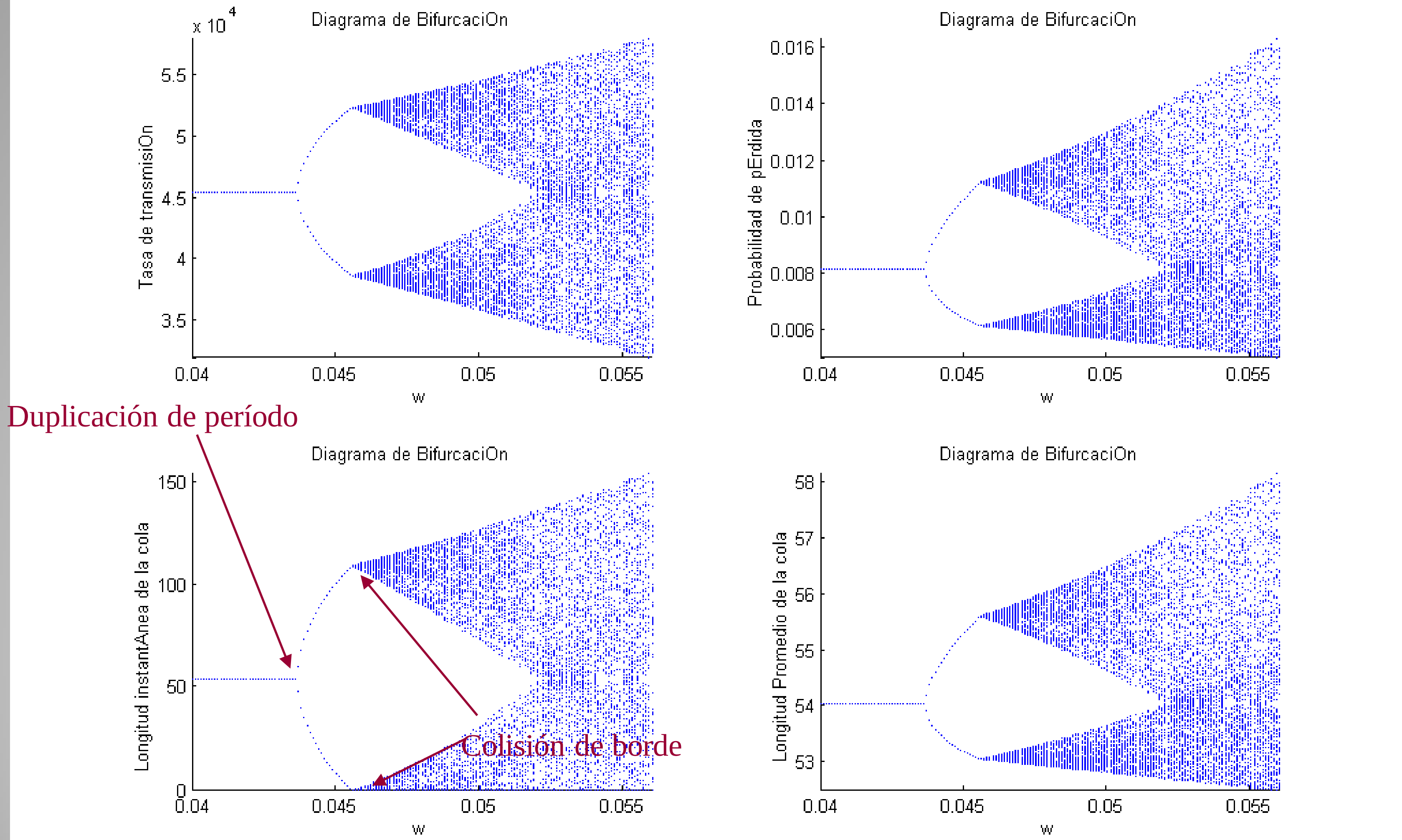
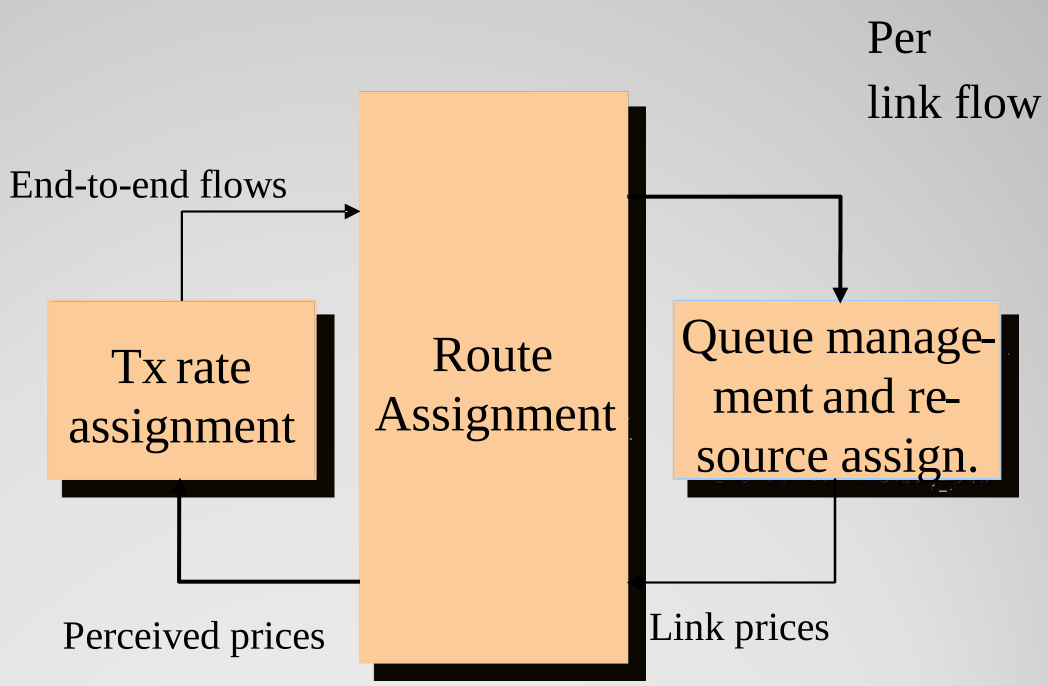
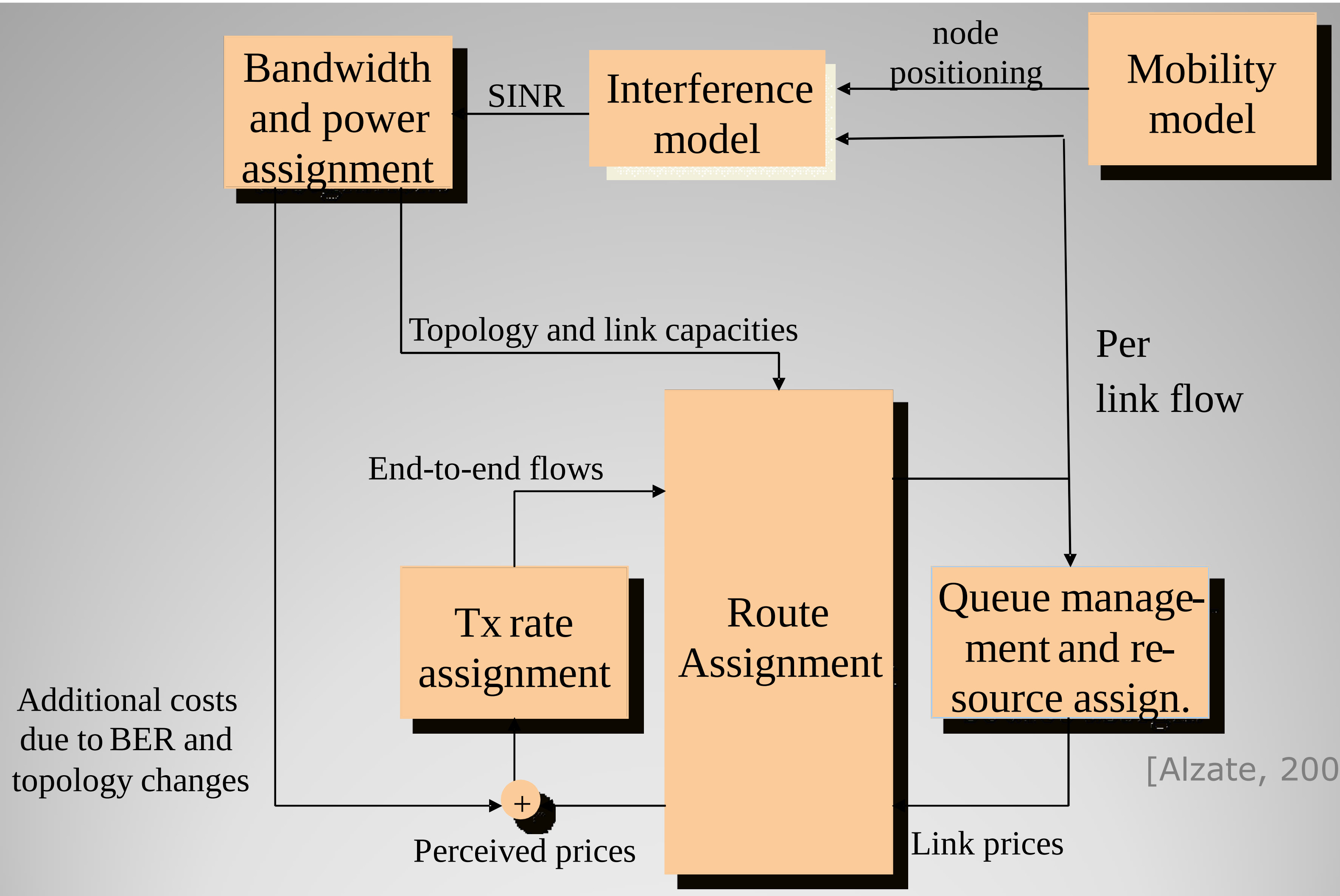


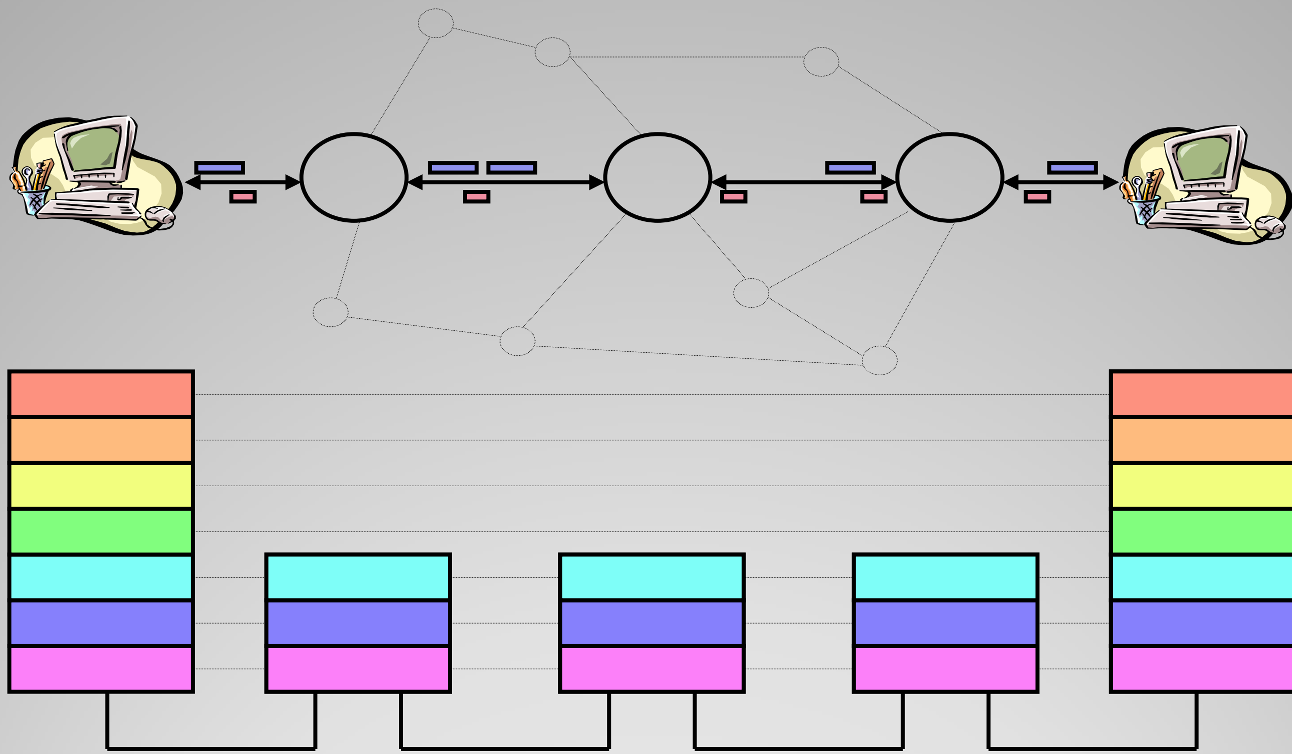
Diagrama de BifurcaciOn
x 10^4
Tasa de transmisiOn
5.5
5
4.5
4
3.5
0.04
0.045
0.05
0.055
w
Diagrama de BifurcaciOn
Probabilidad de pErdida
0.016
0.014
0.012
0.01
0.008
0.006
0.04
0.045
0.05
0.055
w
Duplicación de período
Diagrama de BifurcaciOn
Longitud instantAnea de la cola
150
100
50
0
0.04
0.045
0.05
0.055
w
Colisión de borde
Diagrama de BifurcaciOn
Longitud Promedio de la cola
58
57
56
55
54
53
0.04
0.045
0.05
0.055
w

Chaos!


Per
link flow
End-to-end flows
Tx rate assignment
Route Assignment
Queue management and resource assign.
Perceived prices
Link prices

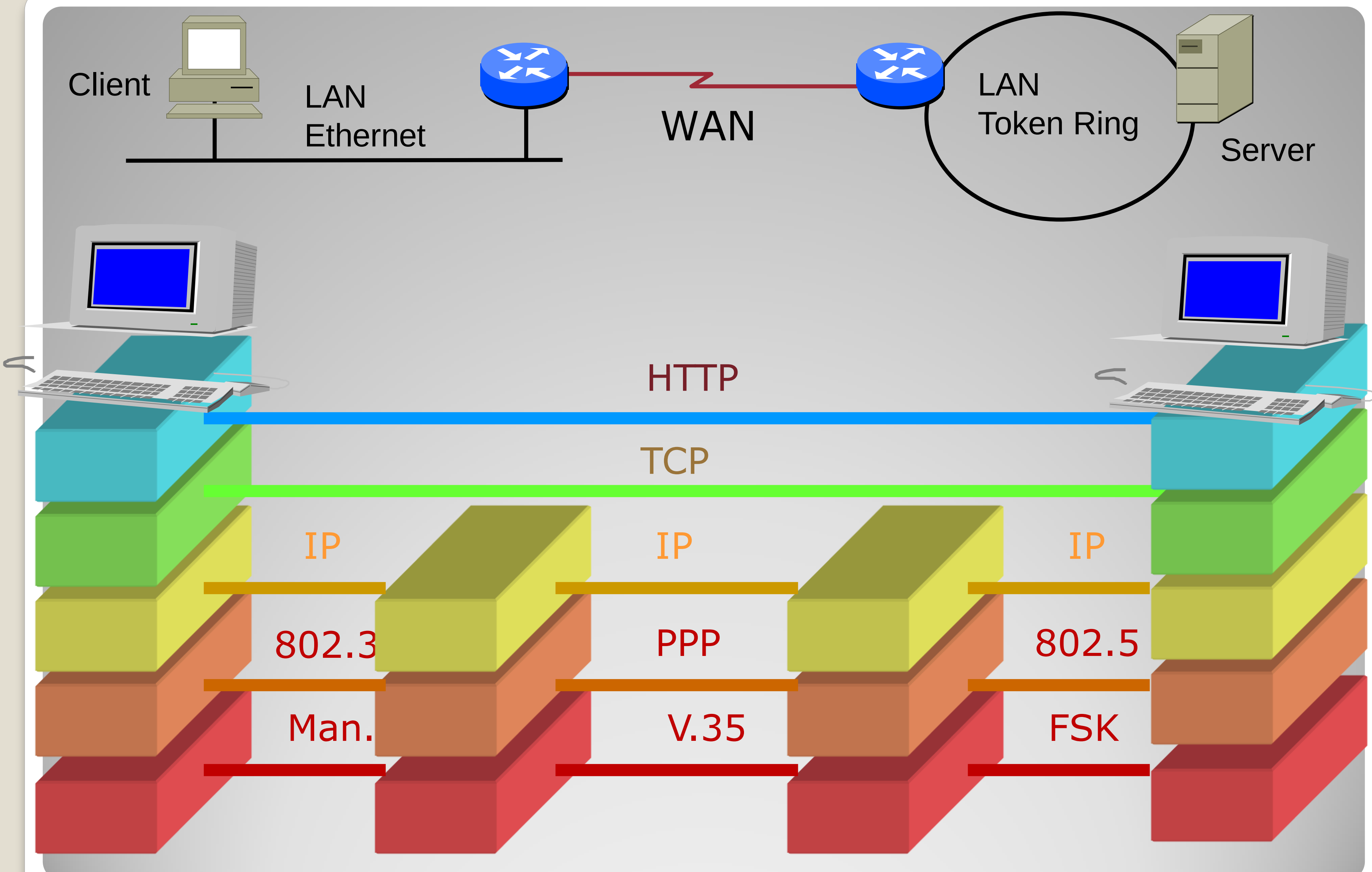

Bandwidth and power assignment
SINR
Interference model
node positioning
Mobility model
Topology and link capacities
Per link flow
End-to-end flows
Tx rate assignment
Route Assignment
Queue manage-ment and re-source assign.
Additional costs due to BER and topology changes
+
Perceived prices
Link prices

[Alzate, 2008]

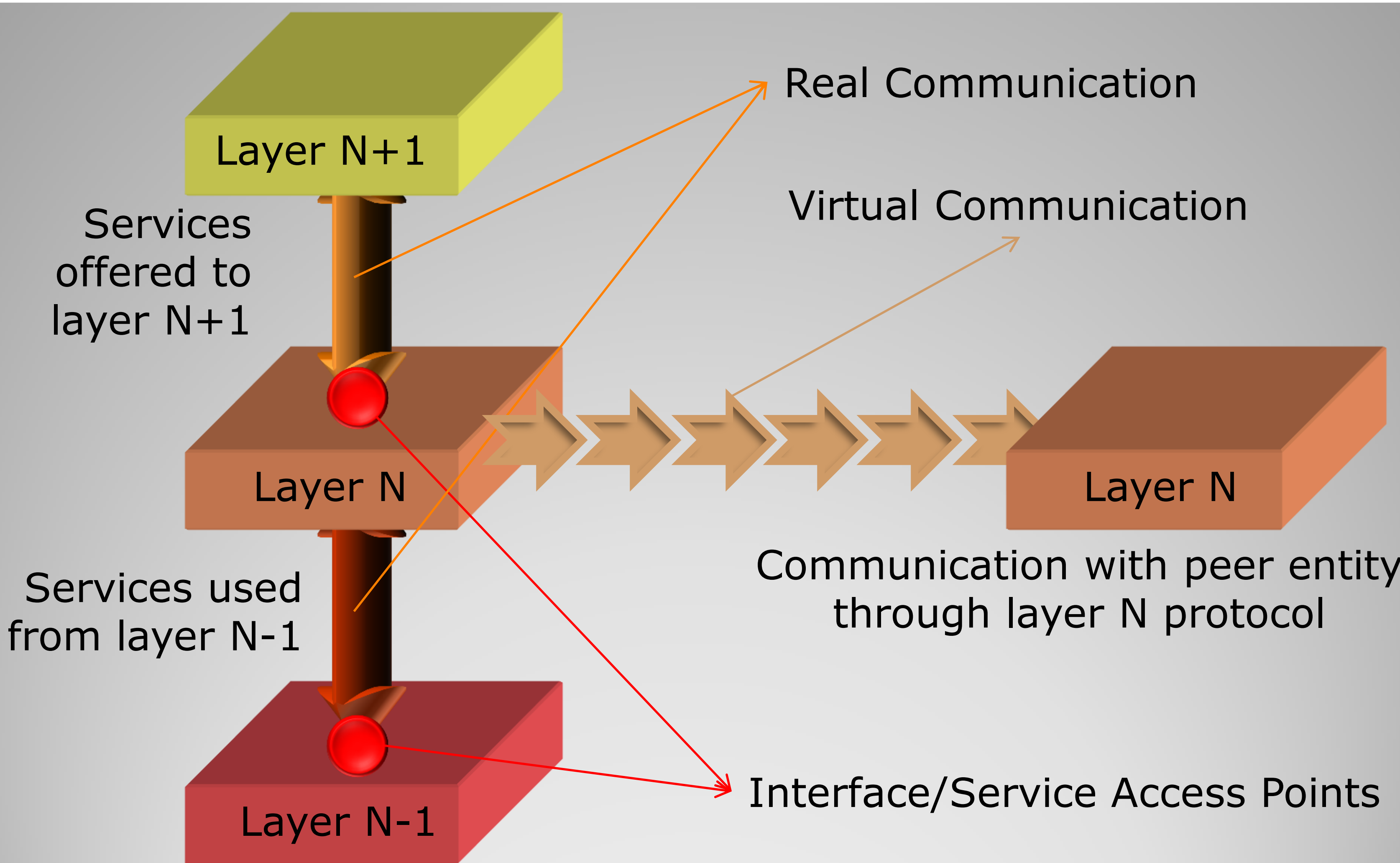


First approach: A functional decomposition in a hierarchical architecture

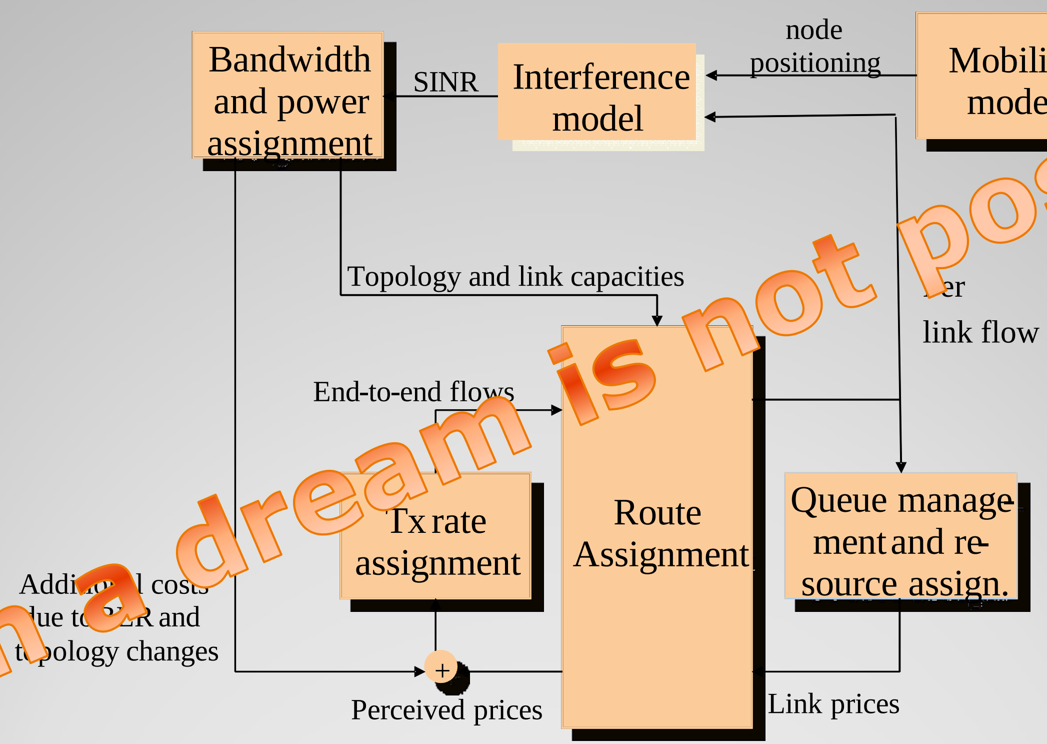
Under such complex interactions, how to design and analyze MANETs?


Client
LAN
Ethernet
WAN
LAN
Token Ring
Server
HTTP
TCP
IP
IP
IP
802.3
PPP
802.5
Man.
V.35
FSK

The main concept



"All interactions should occur only between adjacent layers, through standardized messages interchanged over clean interfaces…"

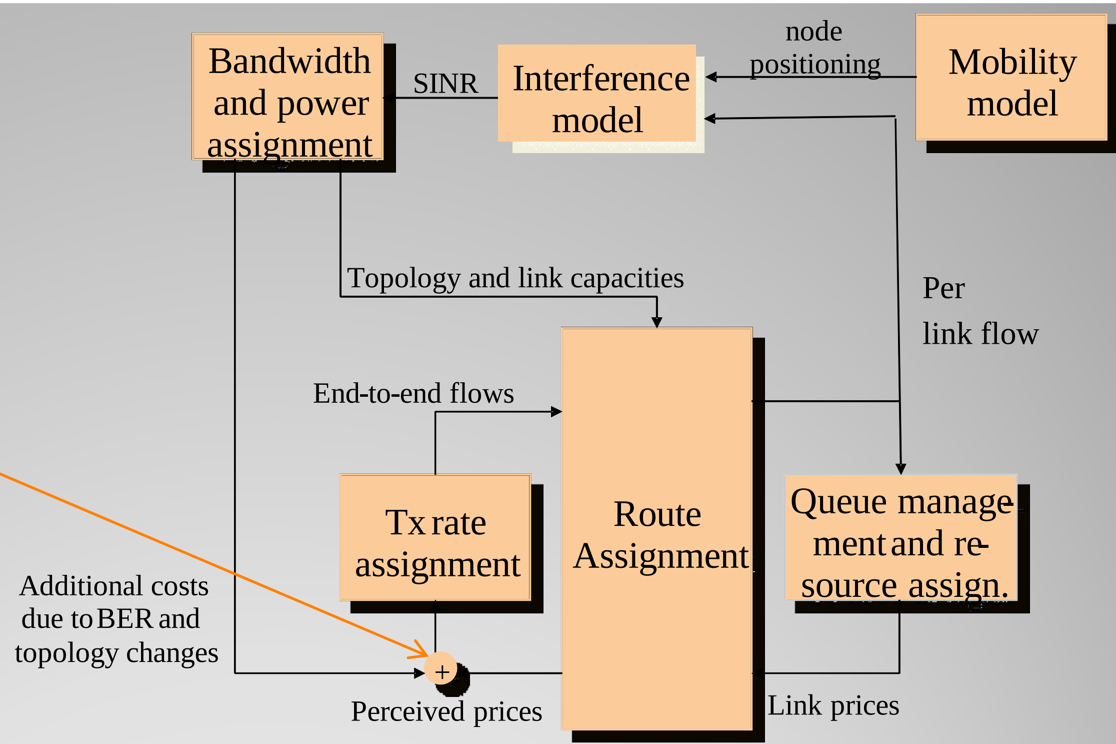


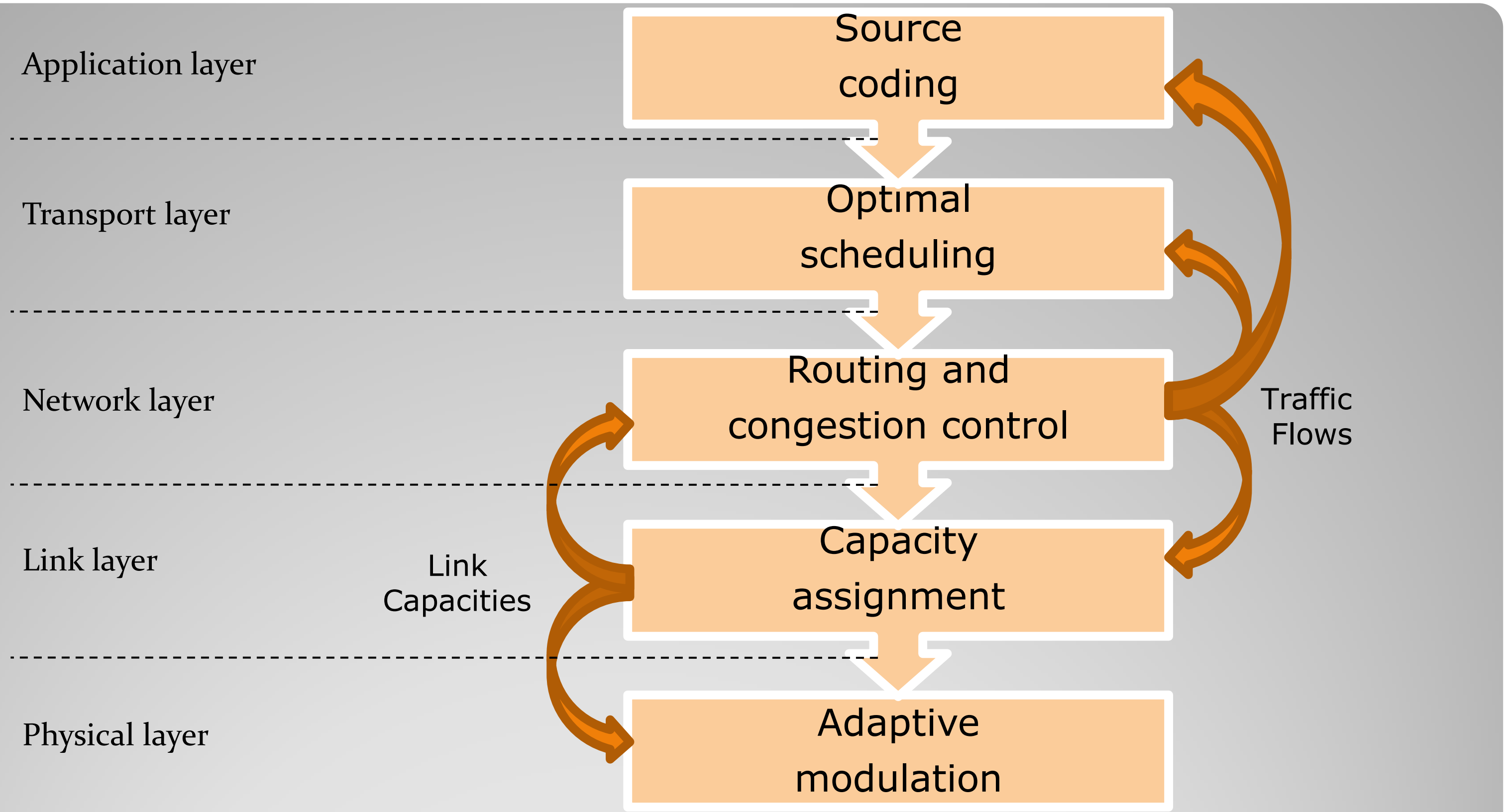
Such a dream is not possible

Functional Hierarchy

Functional Hierarchy

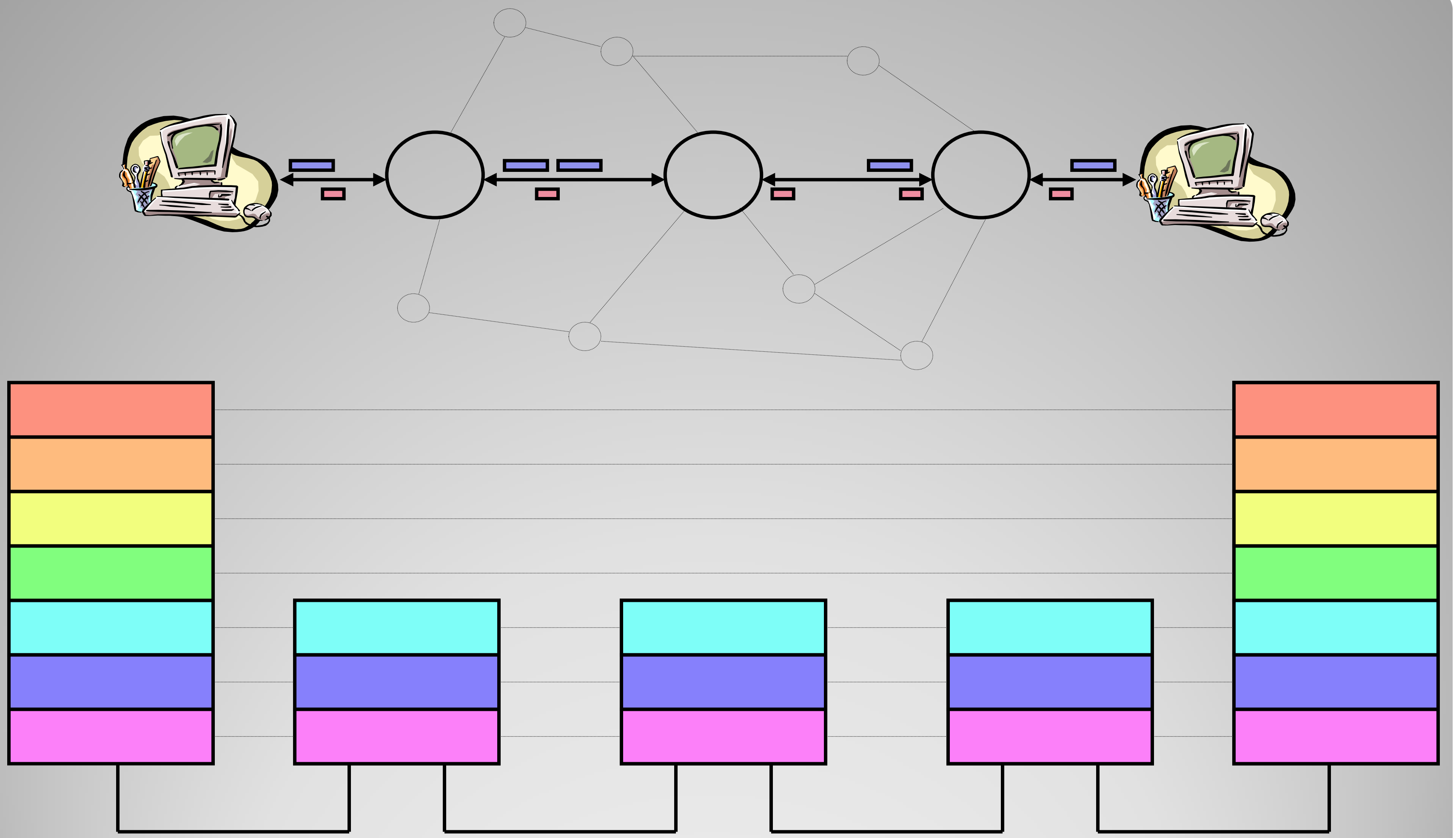
For example, TCP connections using MANET links spend excessive time in congestion avoidance and/or slow start procedures triggered by packet losses… due to transmission errors!!



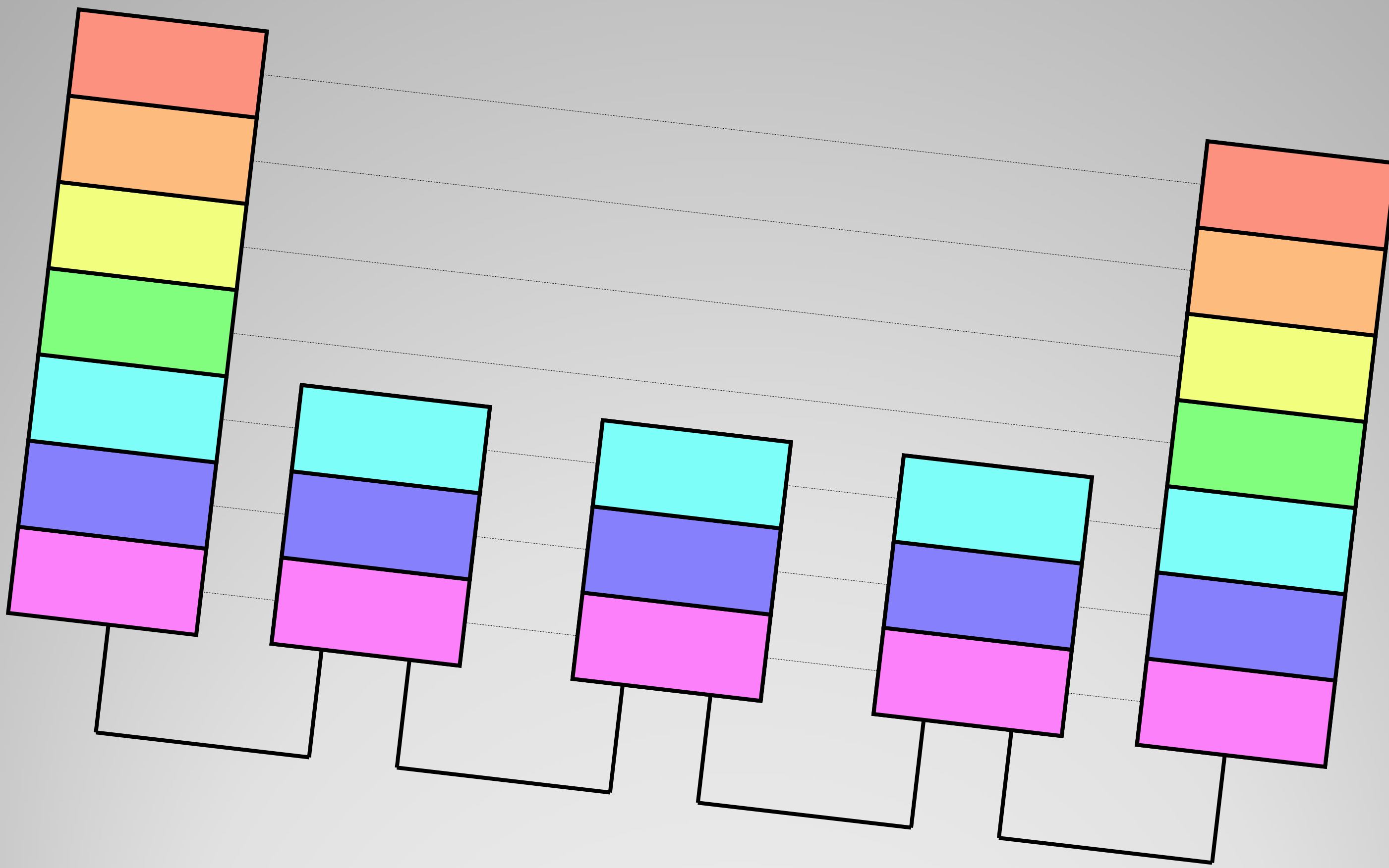


Second approach: Cross-layer design

Under such complex interactions, how to design and analyze MANETs?

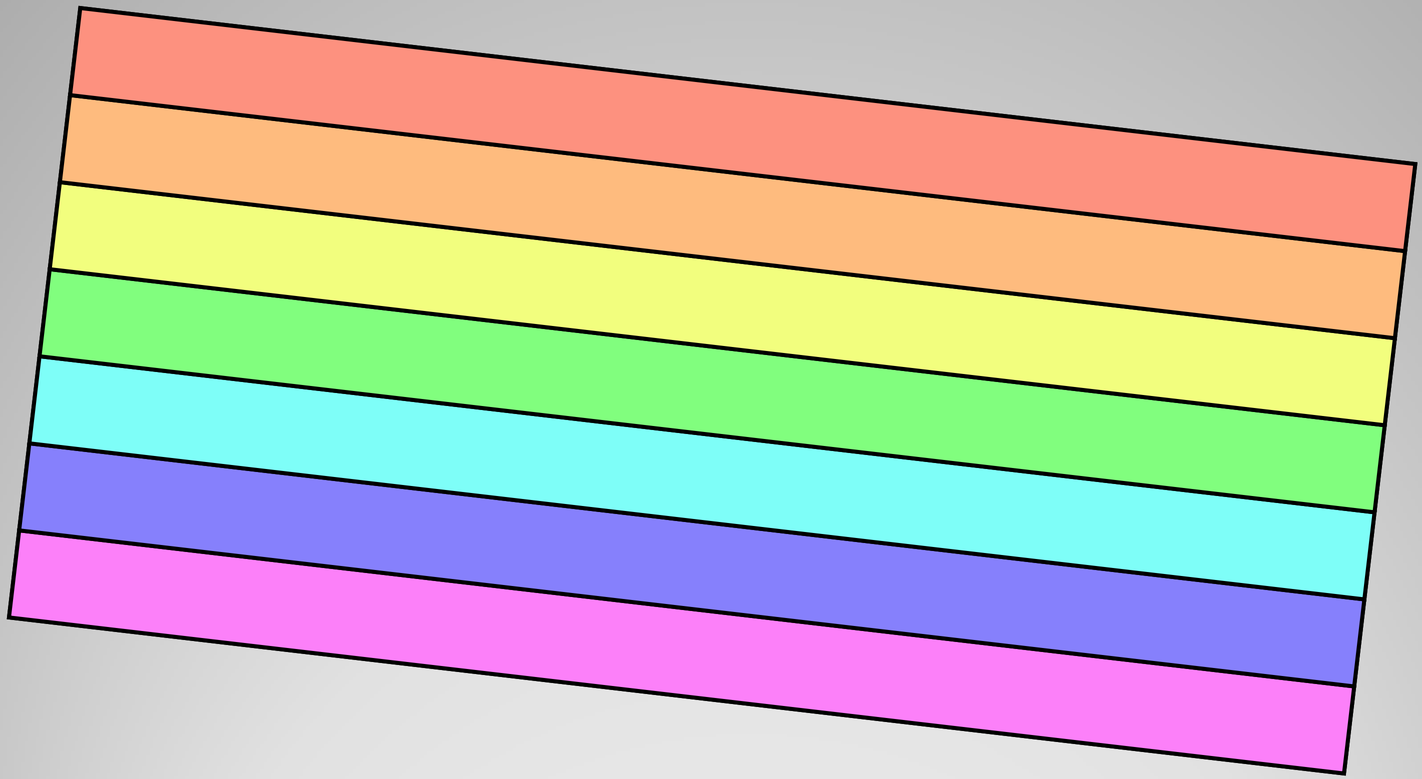


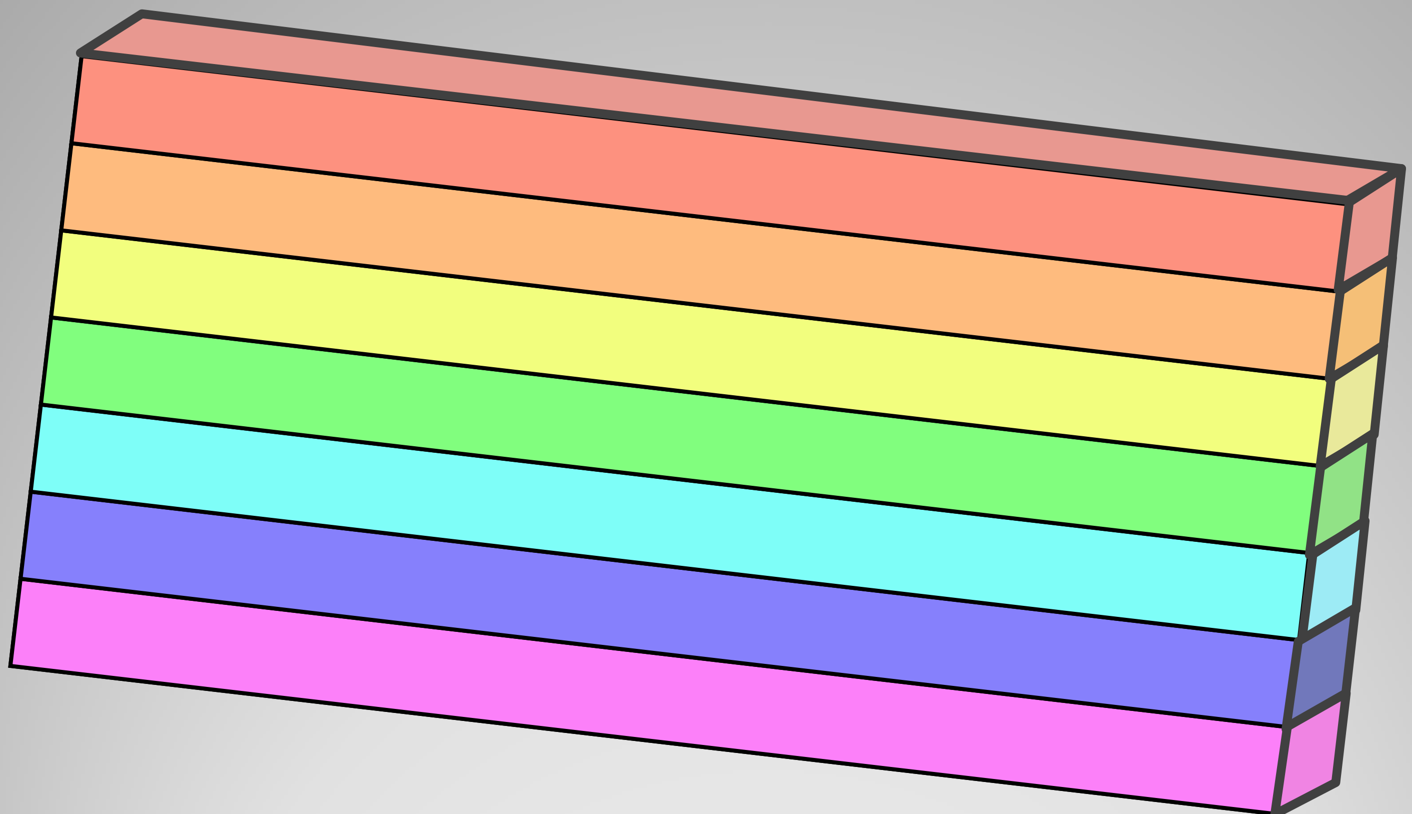
CrossLayer Design



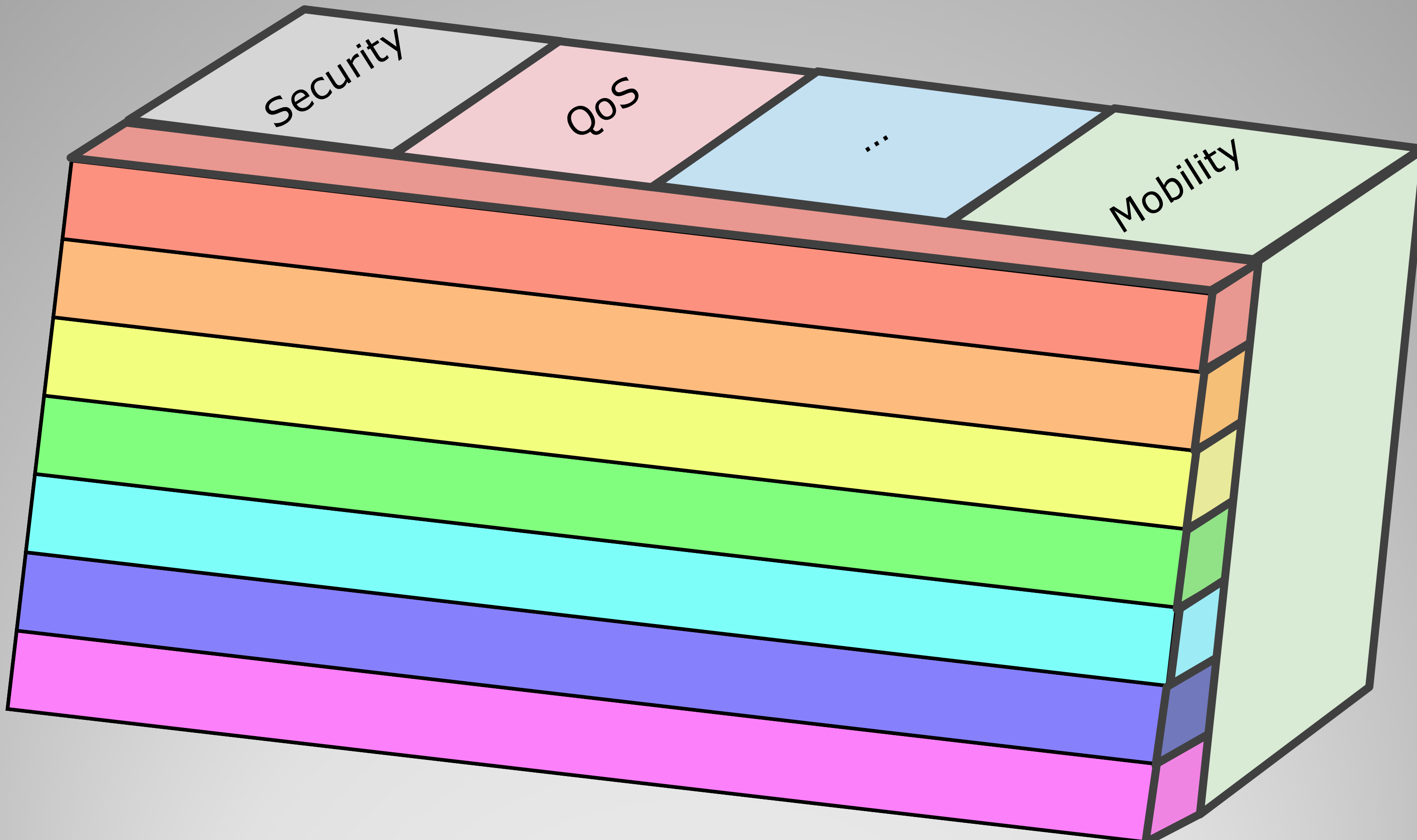
CrossLayer Design

CrossLayer Design





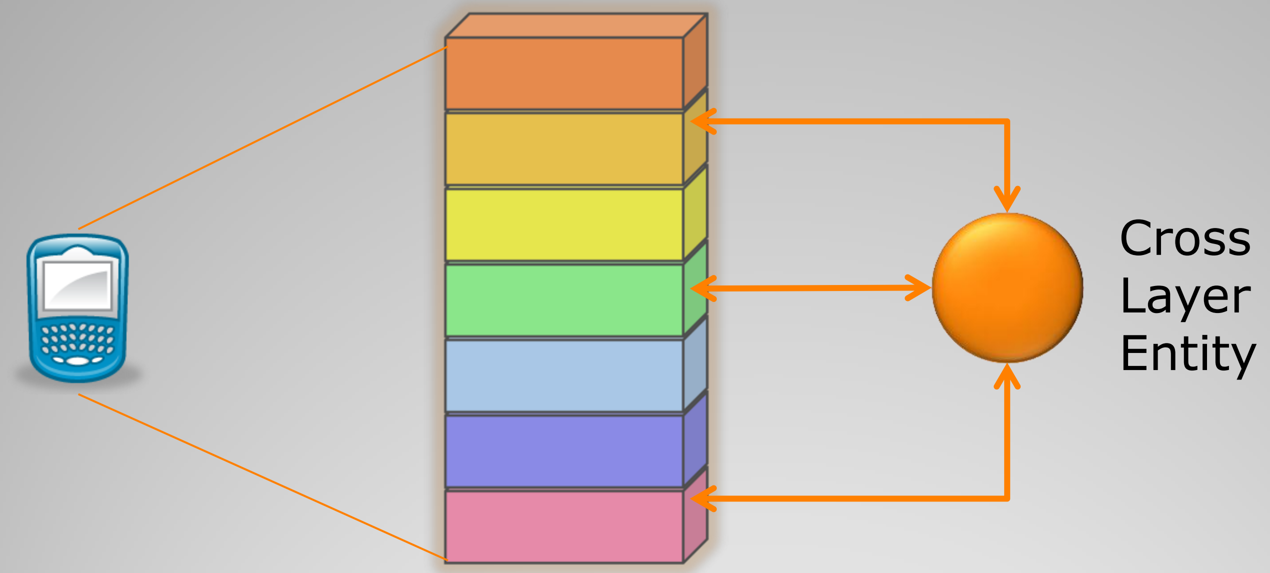
CrossLayer Design



Transversal coordination planes for specific CLD functions

CrossLayer Design

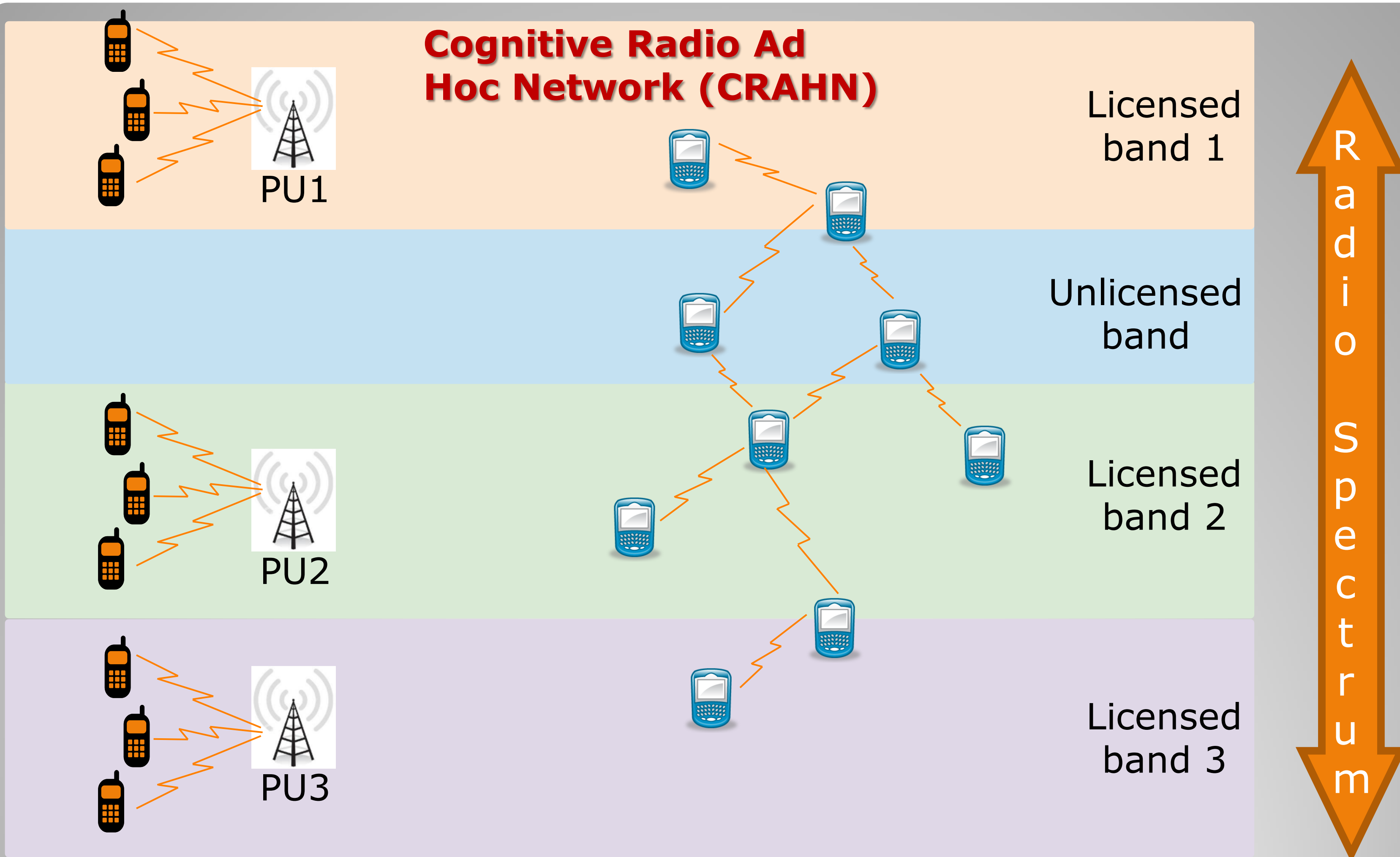
CrossLayer Design



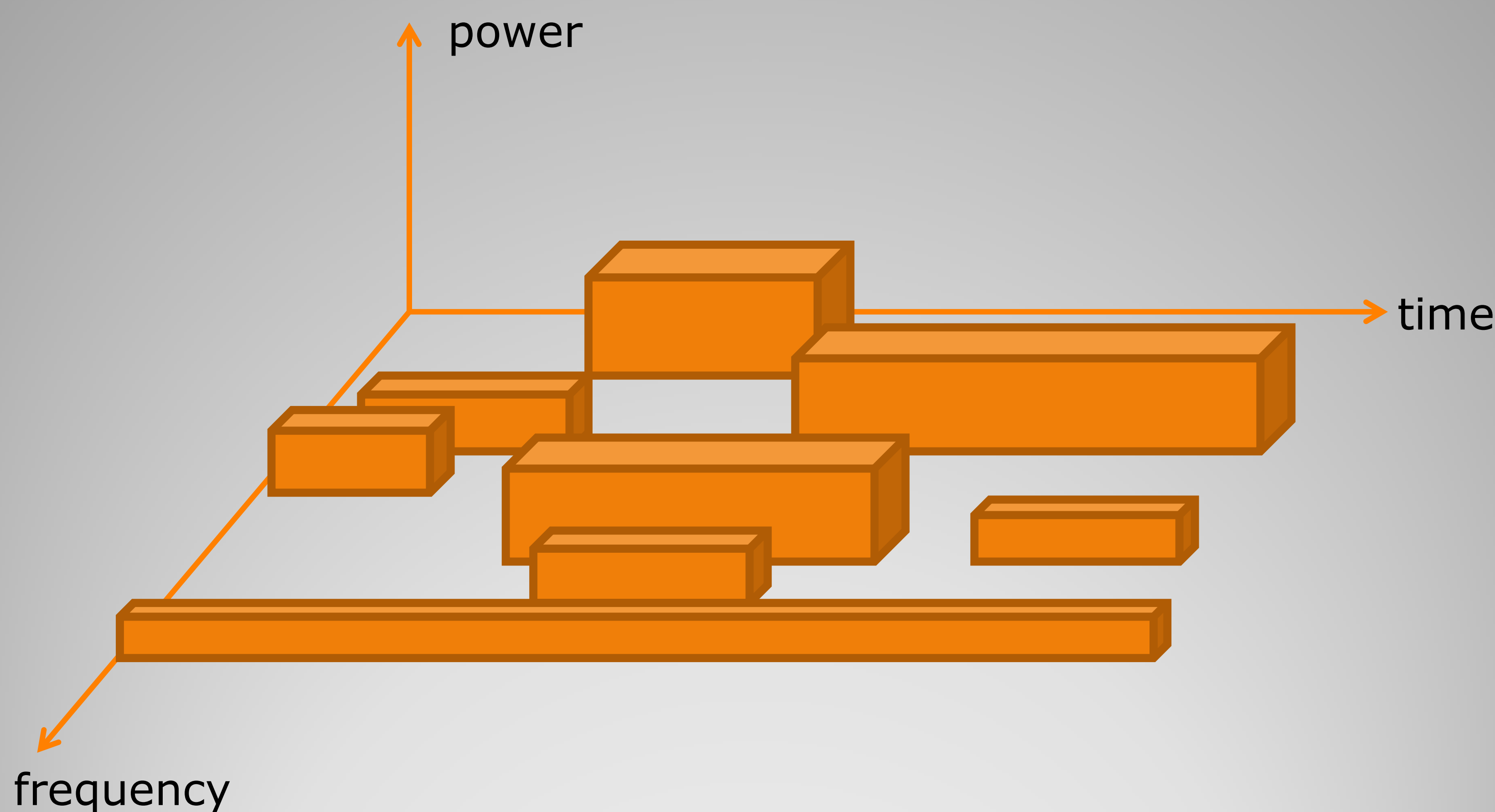
CLD allows communication to take place even between nonadjacent layers through additional entities introduced into the system's architecture

Potential difficulties

1. Modular architecture, tested an innumerable number of times, provides the necessary abstraction to understand the system. CLD do not.
2. A modular design accelerates the development, since designers can focus in specific subsystems, ensuring interoperability. CLD do not.
3. CLD could create additional unintended interactions among layers, with possible instability effects.
4. While modular architecture facilitates the standardization, CLD needs a new design for each new application.
5. With CLD it is not possible to redesign isolated parts of the system.
6. A little abuse of the CLD principles could lead to a “spaghetti” design that hinders innovation and maintenance.

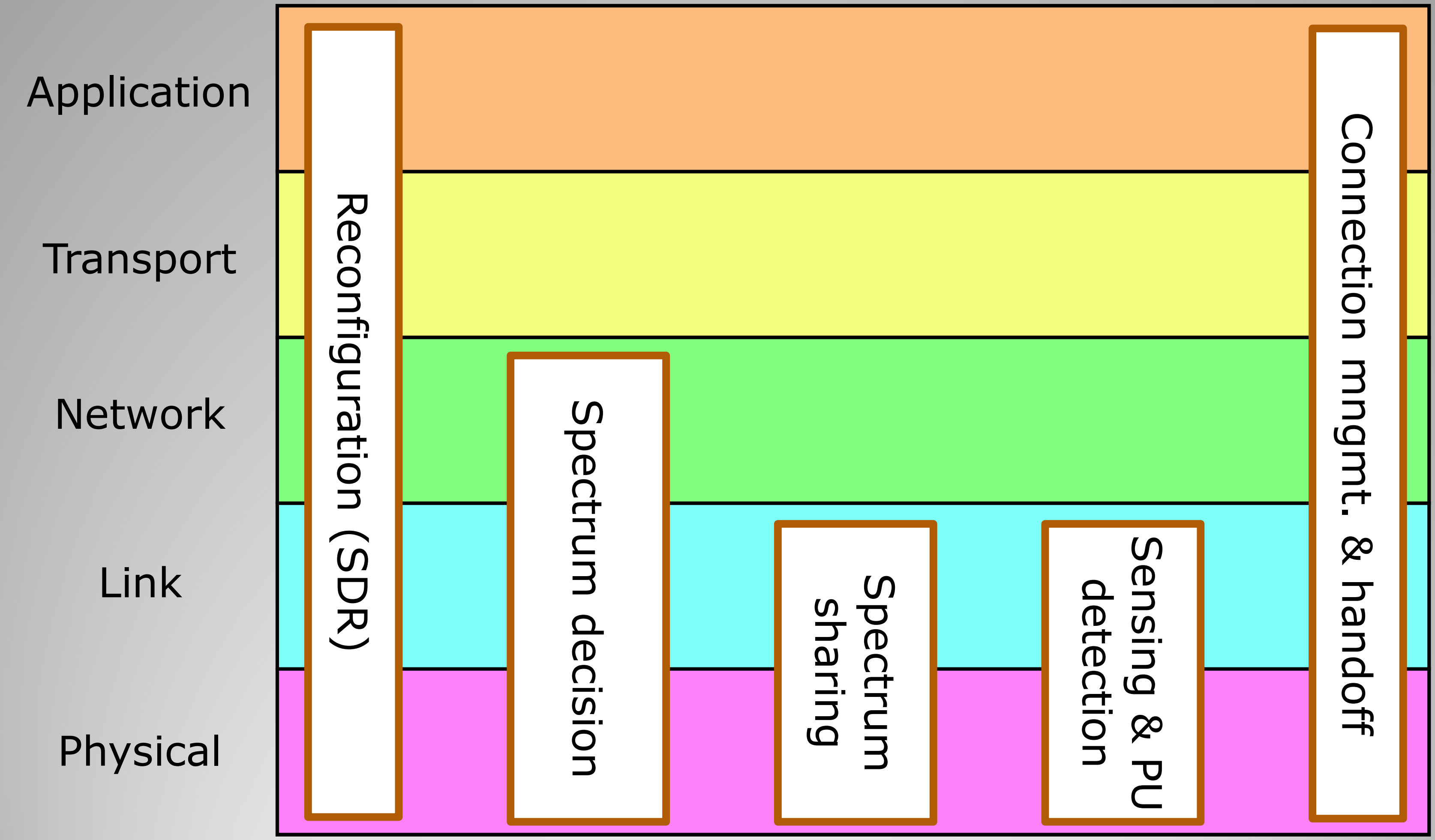


Under such complex interactions, how to design and analyze MANETs?



Spectrum sensing/access/sharing

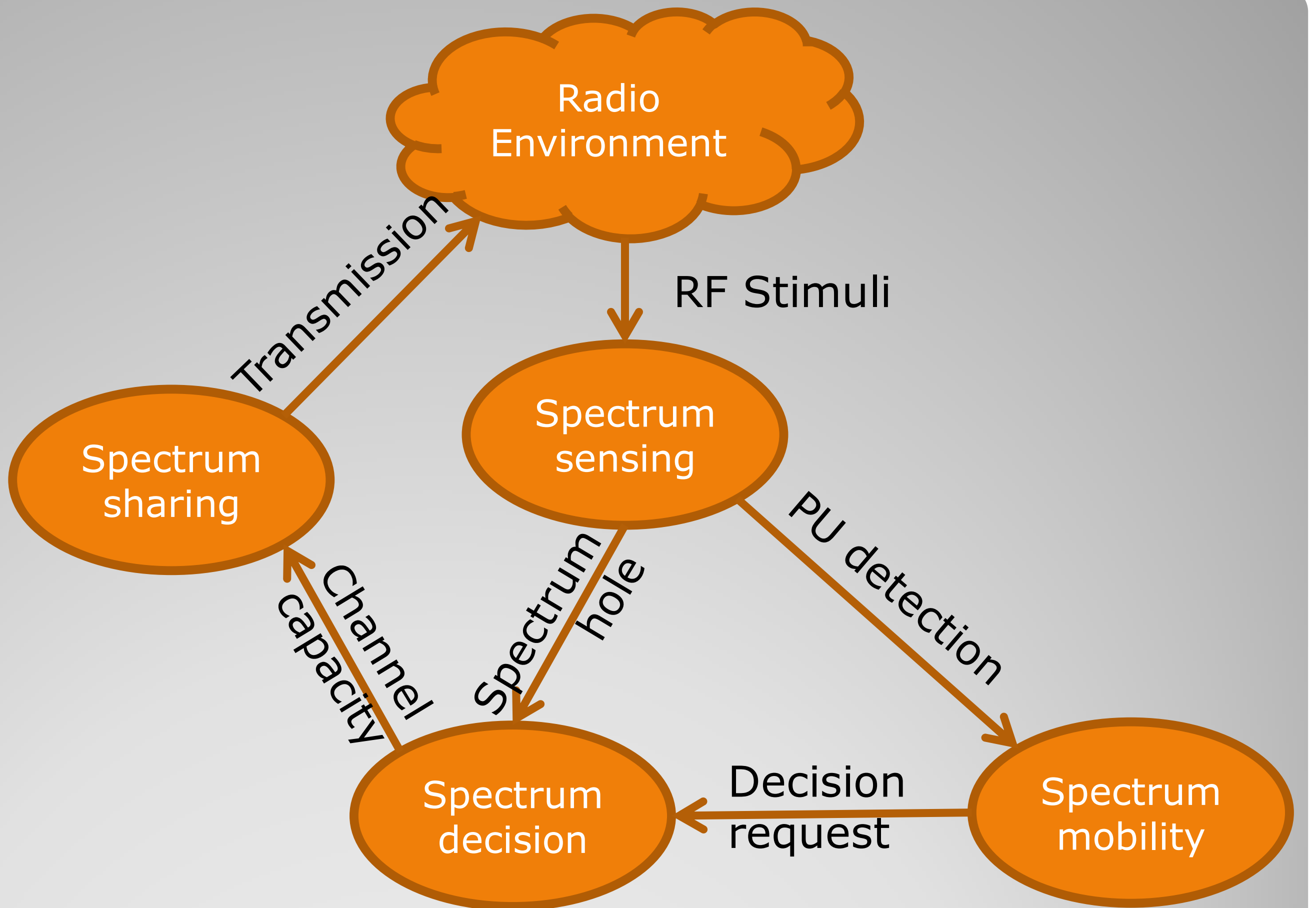
CRAHN



CRAHN : Cross Layer Design

CRAHN: Complex adaptive system

- **Sensing**
- **Learning**
- **Adapting**



[Akyildiz, Lee, Chowdhury, 2009]

- Introduction to Mobile Ad Hoc Networks
- **Introduction to Complex Adaptive Systems**
- Emergent Synchronization in MANETs
- Emergent Cooperation in MANETs
- Formal methods for engineering emergent behavior in MANETs

Schedule

Complex Systems Theory

- There is no a widely accepted definition (nor a narrowly accepted one), but we can listen to some experts:

- George Cowan founded the Santa Fe Institute in 1984 (http://www.santafe.edu/)
  - "Founded for multidisciplinary collaborations in the physical, biological, computational, and social sciences, attempting to uncover the mechanisms that underlie the deep simplicity present in our complex world. Understanding of complex adaptive systems is critical to addressing key environmental, technological, biological, economic, and political challenges."

- There is no a widely accepted definition (nor a narrowly accepted one), but we can listen to some experts:

- Stephen Wolfram began the Center for Complex Systems Research at the University of Illinois, in 1986 (http://www.ccsr.uiuc.edu/)
  - "Studies systems that display adaptive, self-organizing behavior and systems that are usually characterized by a large throughput, such as turbulent flow, lightning, and the flow of information through the internet. To describe these complex systems, we develop models and techniques drawn from nonlinear dynamics and chaos, neural nets, cellular automata, artificial life, and genetic algorithms."

Complex Systems Theory

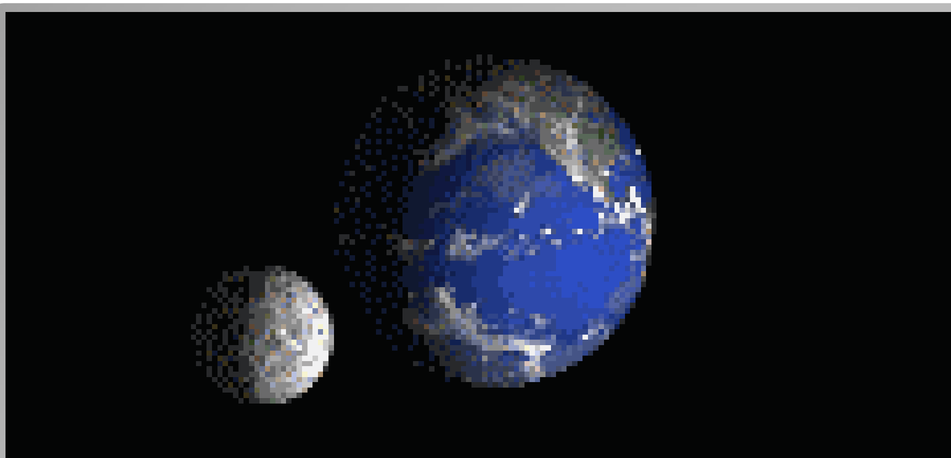
- There is no a widely accepted definition (nor a narrowly accepted one), but we can listen to some experts:
- Yaneer Bar-Yam founded the New England Complex Systems Institute in 1992 (http://necsi.org/)
  - "Studies how parts of a system give rise to its collective behaviors, as well as how the system interacts with its environment. By using mathematics to focus on pattern formation, and the question of parts, wholes and relationships, the field of complex systems cuts across all the disciplines of science, as well as engineering, management, and medicine."

Complex Systems Theory

- There is no a widely accepted definition (nor a narrowly accepted one), but we can listen to some experts:

- Springer-Verlag launches its Complexity series in 2004 (www.springer.com/complexity)
  - “Complex Systems are systems that comprise many interacting parts with the ability to generate a new quality of macroscopic collective behavior the manifestations of which are the spontaneous formation of distinctive temporal, spatial or functional structures. Models of such systems can be successfully mapped onto quite diverse "real-life" situations like the climate, the coherent emission of light from lasers, chemical reaction-diffusion systems, biological cellular networks, the dynamics of stock markets and of the internet, earthquake statistics, freeway traffic, the human brain, or the formation of opinions in social systems, to name just some of the popular applications. Although their scope and methodologies overlap somewhat, one can distinguish the following main concepts and tools: self-organization, nonlinear dynamics, synergetics, turbulence, dynamical systems, catastrophes, instabilities, stochastic processes, chaos, graphs and networks, cellular automata, adaptive systems, genetic algorithms and computational intelligence.”

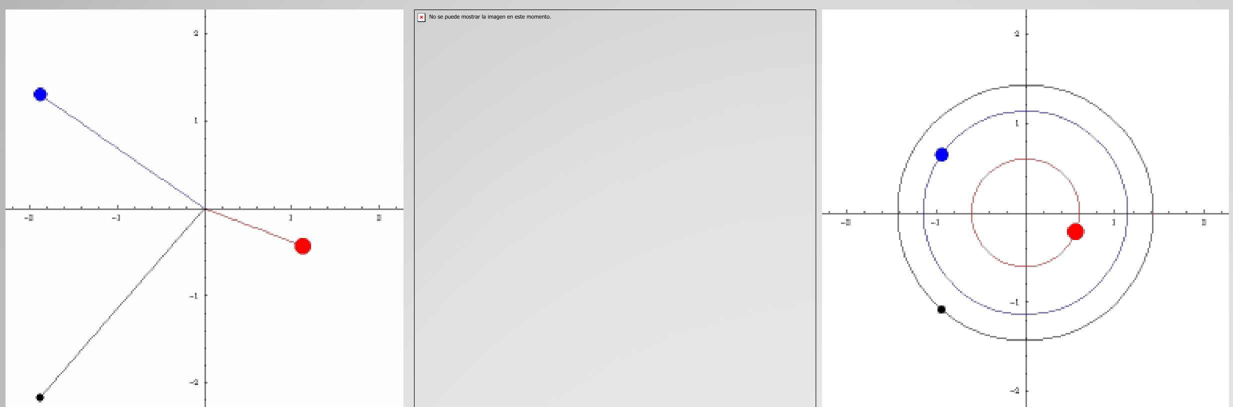
Complex Systems Theory



Two masses attract each other with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them:

Simple elements, simple interactions, but…

http://bestanimations.com/Earth&Space/Moon/Moon2.html



…Complex Emergent Behavior

http://www.scholarpedia.org/article/Three_body_problem

Complex Systems Theory

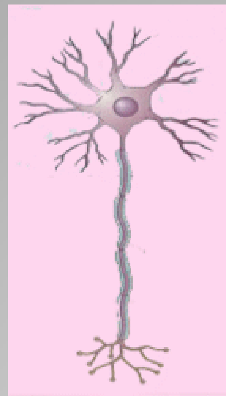
Simple components
Simple Interactions

An isolated ant is very “simple”:
- Deposits pheromones
- Follows pheromone tracks
- Superposes random walk

Complex Systems Theory

Complex emergent self-organized behavior

An ant colony easily solves highly complex problems (NP-complete):
- Shortest path
- Sorting



Simple components
Simple Interactions

An isolated neuron is very “simple”:
- Processes input stimuli
- Generates output stimuli

Complex emergent self-organized behavior

A human brain is capable of highly complex behaviors:
- Learning
- Generalizing
- Consciousness
- Intuition

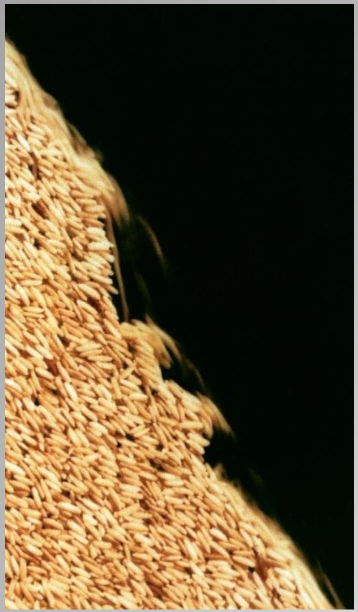
Complex Systems Theory

Simple components
Simple Interactions

Complex emergent self-organized behavior

Complex Systems Theory

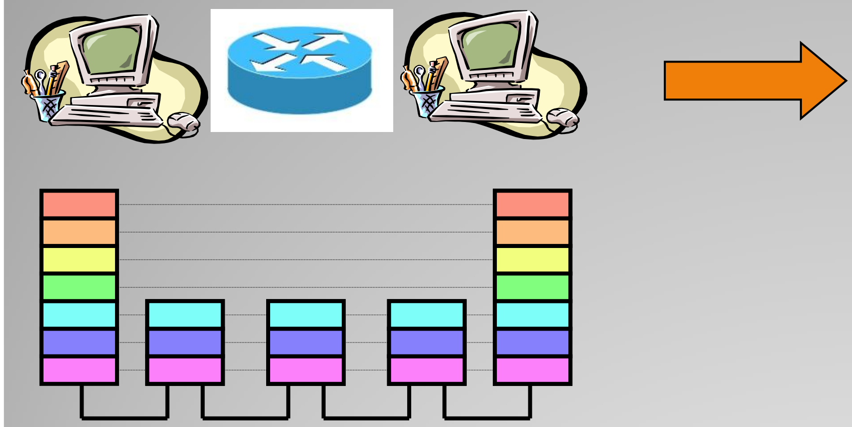
ideas
Investigación, Desarrollo y Aplicaciones en Señales



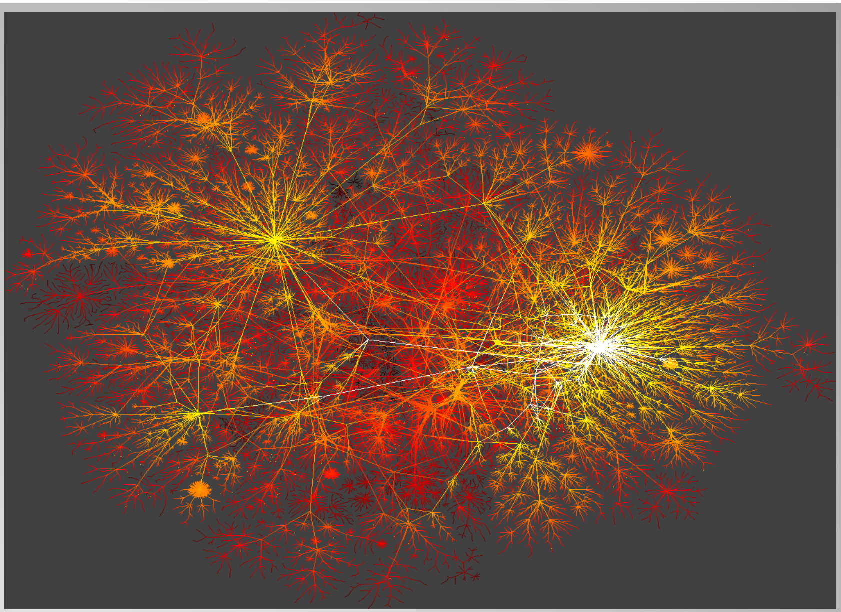
Simple components
Simple Interactions

Complex emergent self-organized behavior

Complex Systems Theory



Simple components
Simple Interactions



Complex emergent self-organized behavior

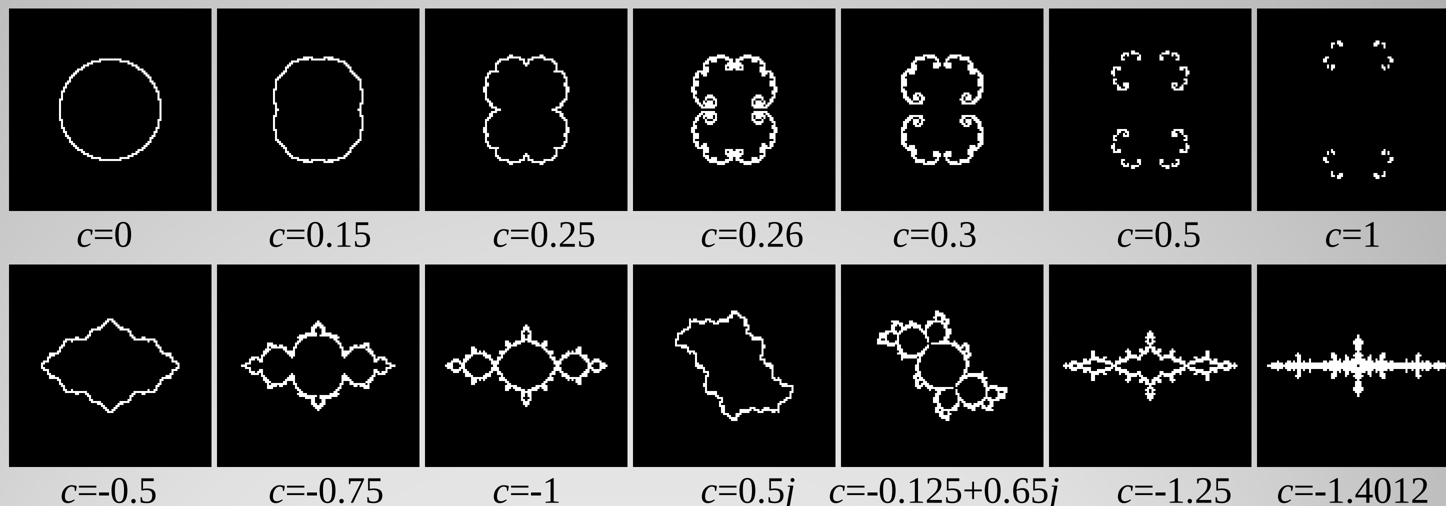
- Fractal traffic
- Scale free topologies
- Potentially chaotic dynamics
- Self-organization at edge of congestion
- ...

Complex Systems Theory

$$x_{n+1} = x_n^{\,2} + c$$

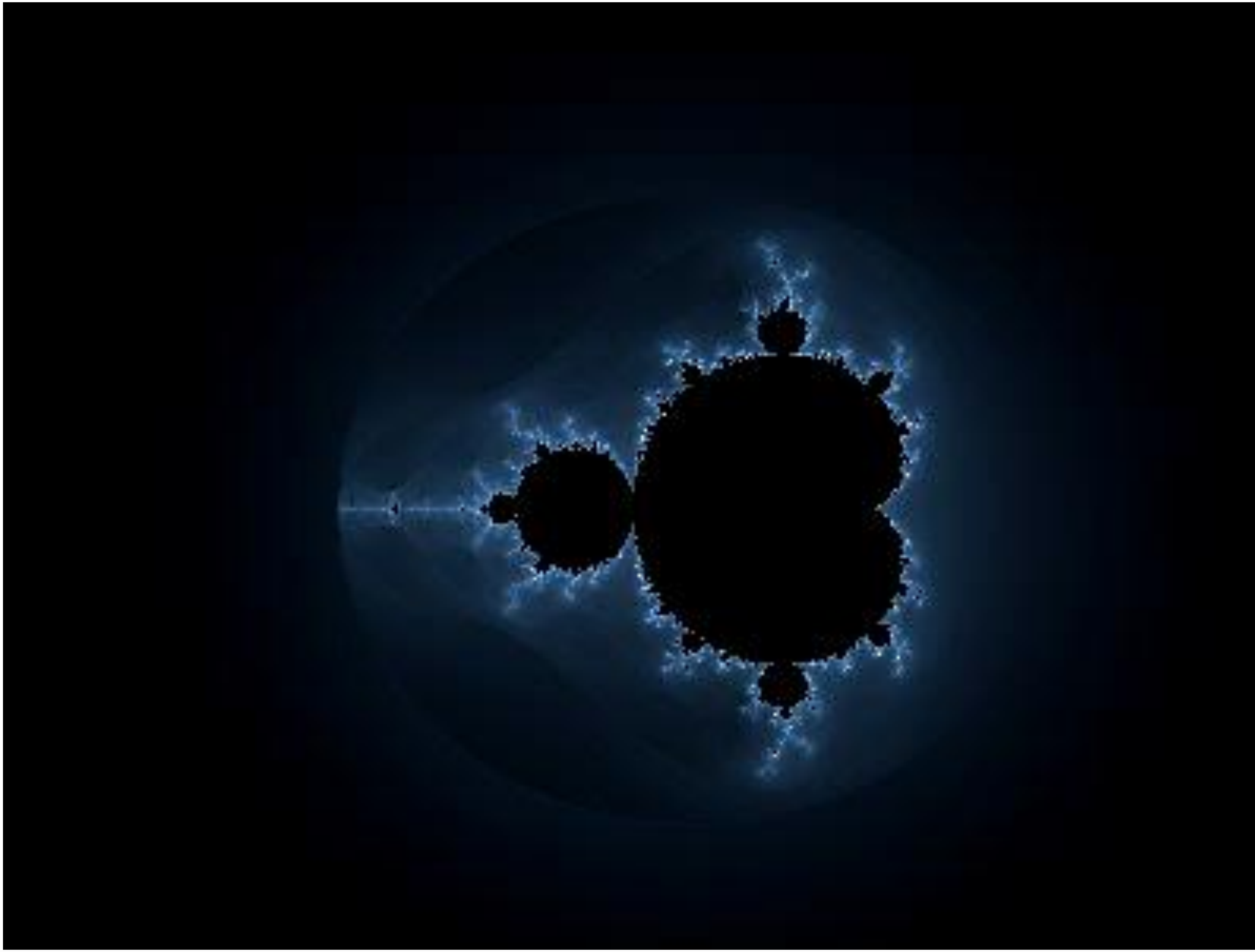
Simple components

Simple Interactions

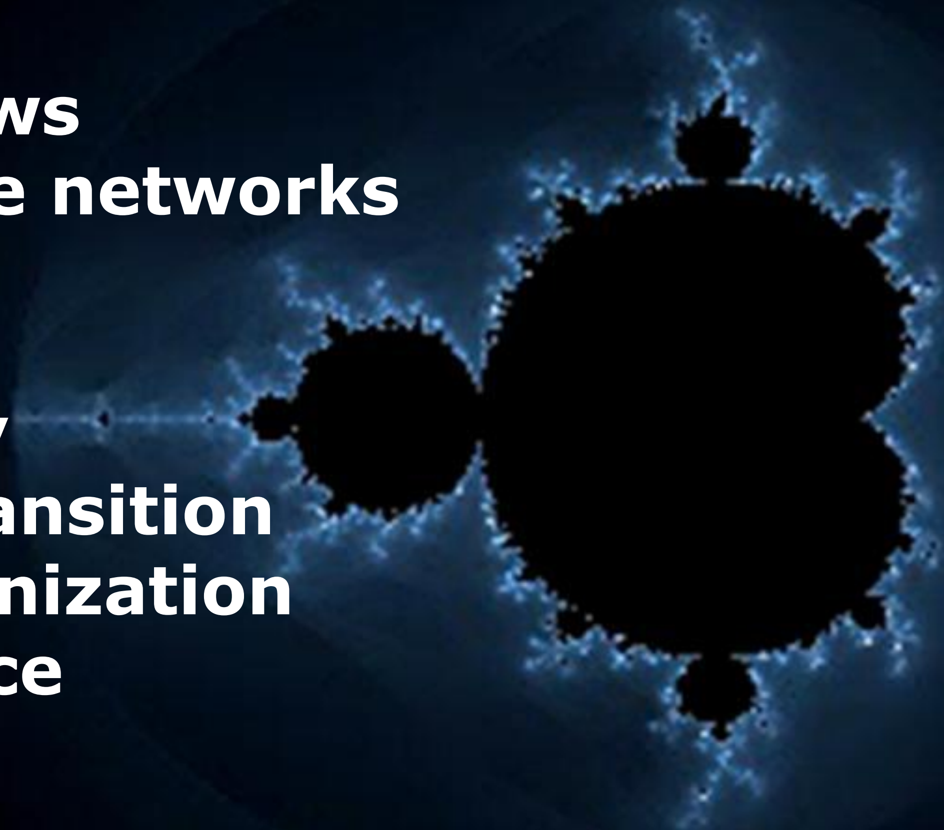


Complex emergent behavior

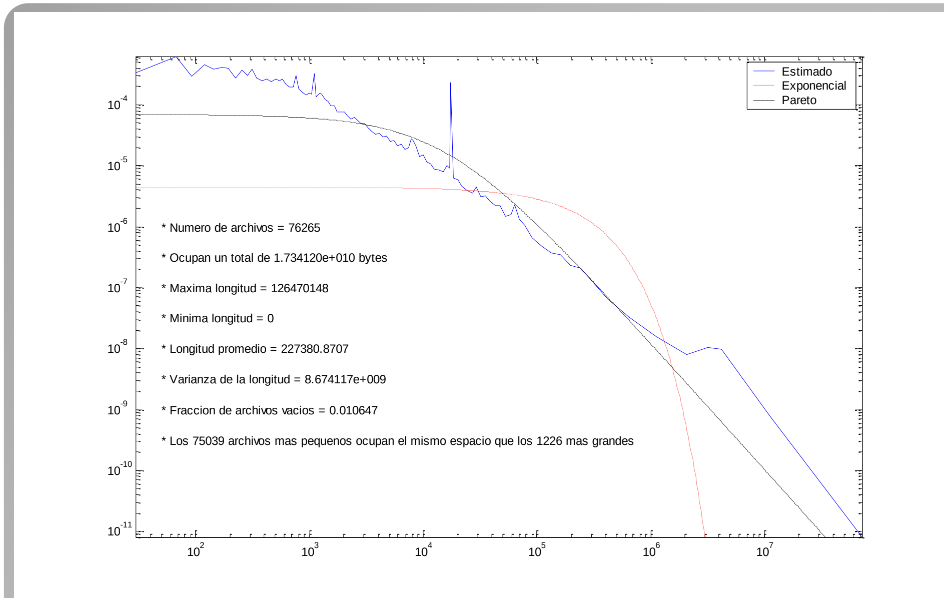
Complex Systems Theory

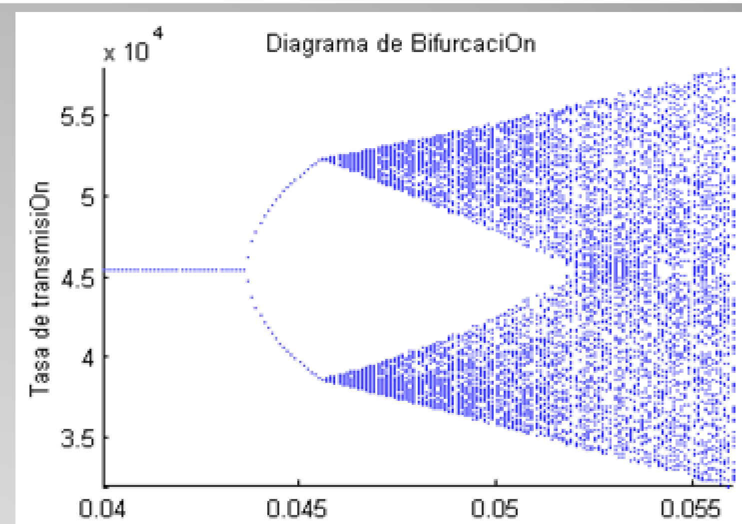


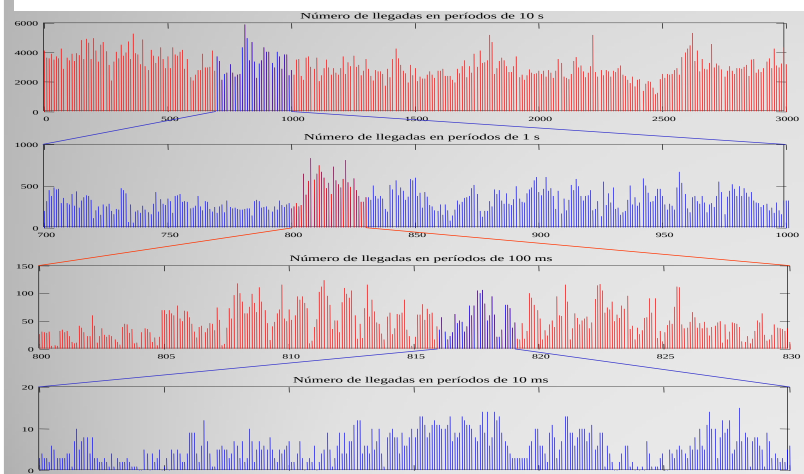
Some complexity signatures

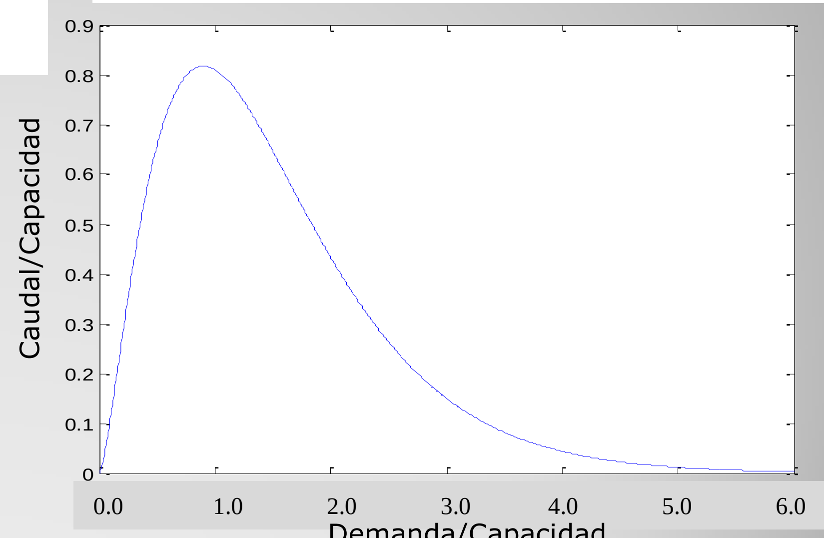


- **Power laws**
- **Free scale networks**
- **Fractals**
- **Chaos**
- **Criticality**
- **Phase Transition**
- **Self-organization**
- **Emergence**
- **Learning**
- **Evolution**
- **Adaptability**
- **…**



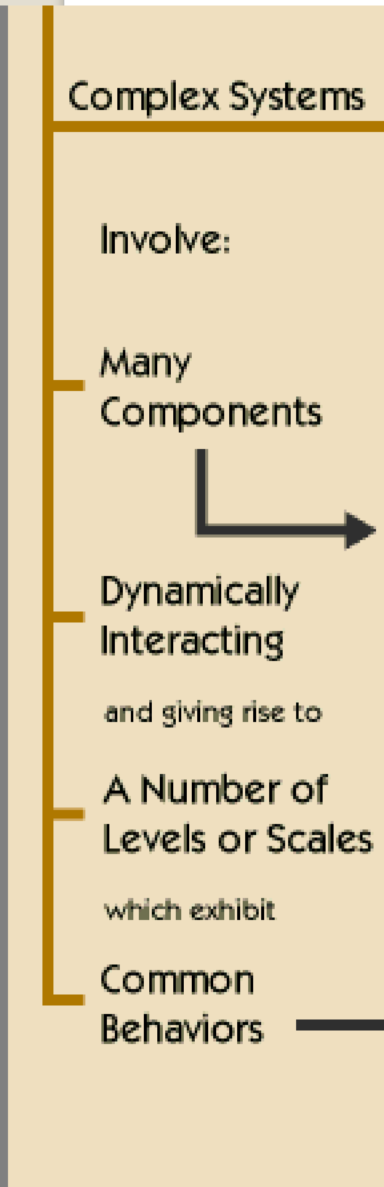


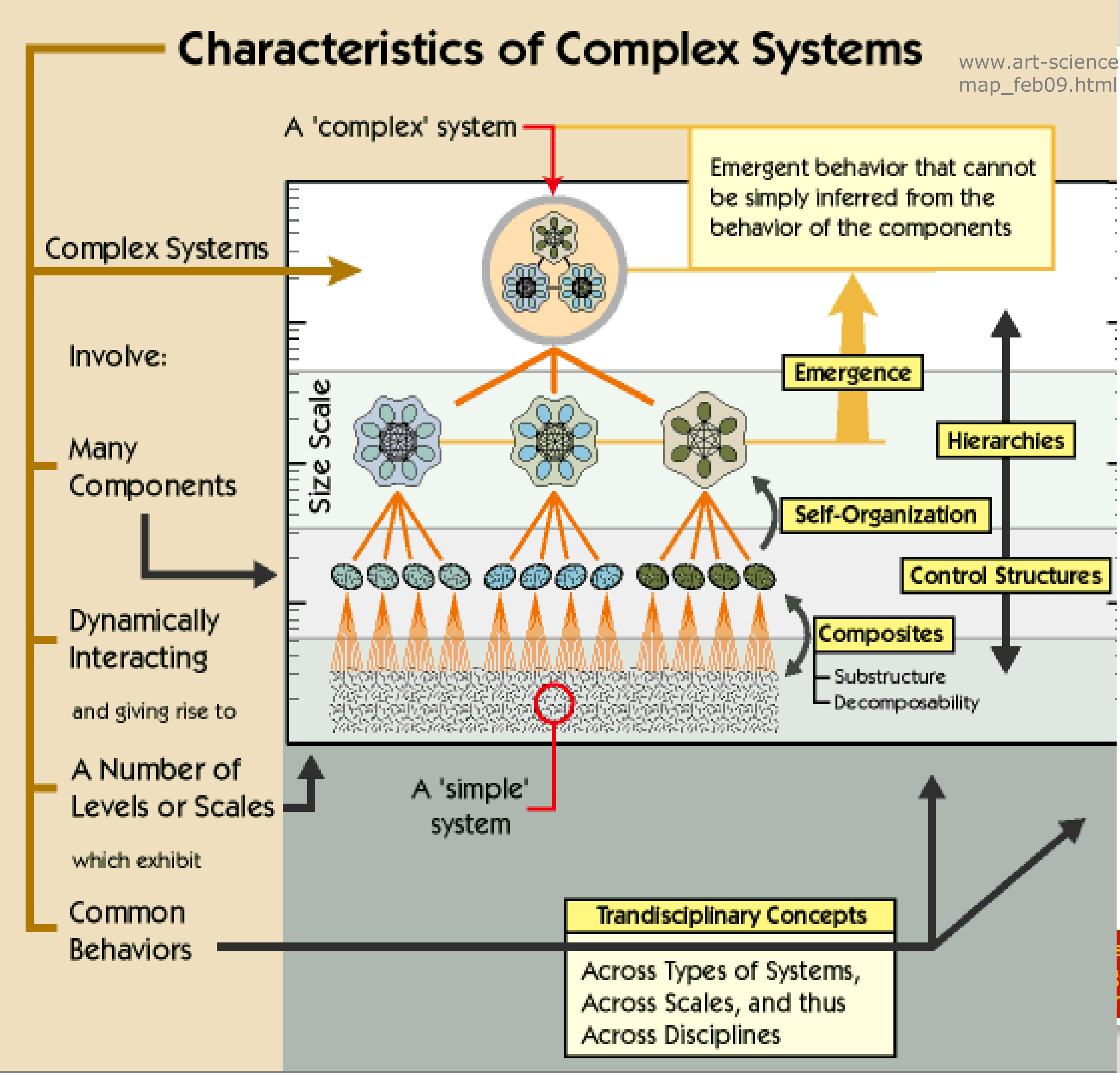




Complexity in communication Networks

www.art-sciencefactory.com/complexity-map_feb09.html



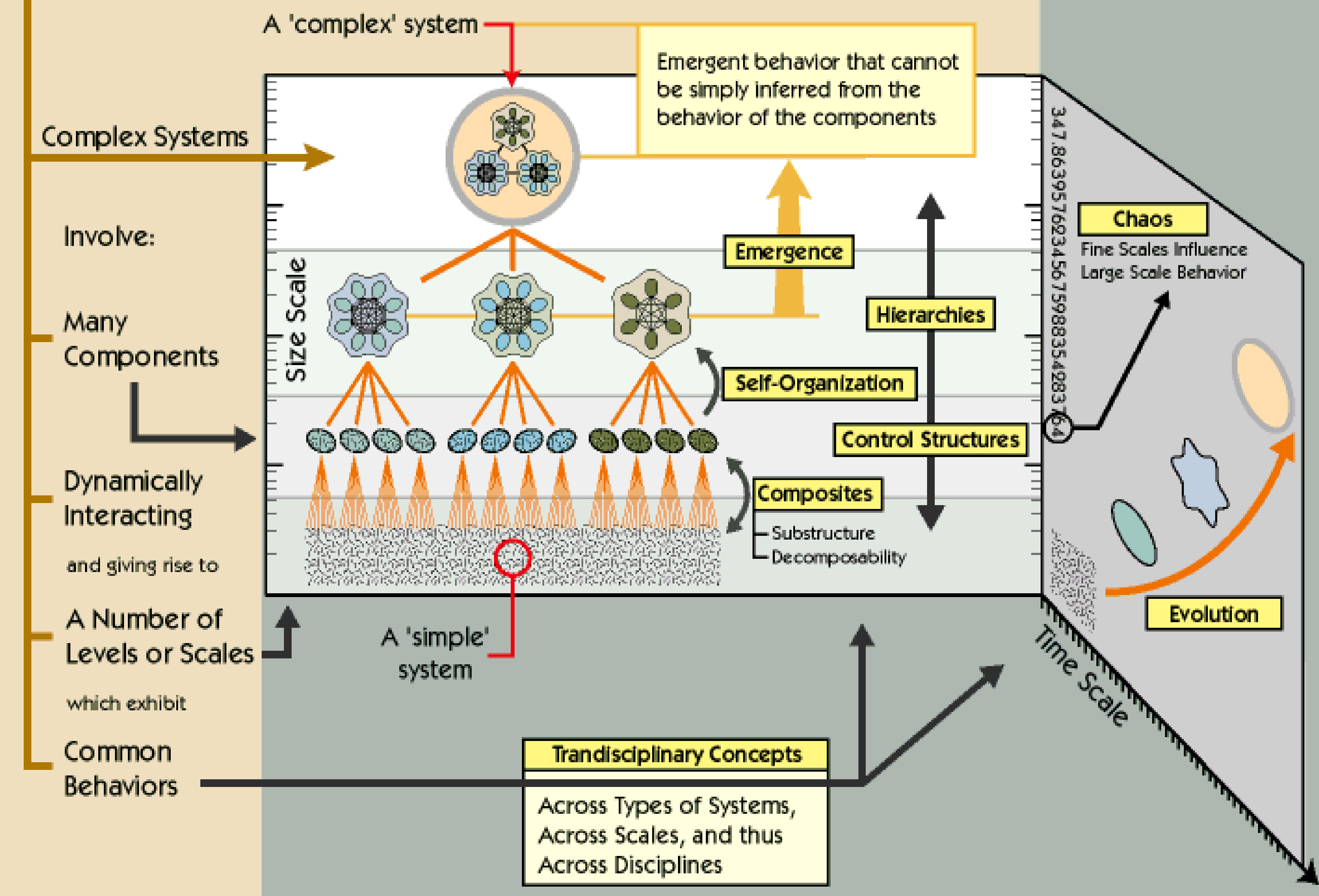


www.art-sciencefactory.com/complexity-map_feb09.html

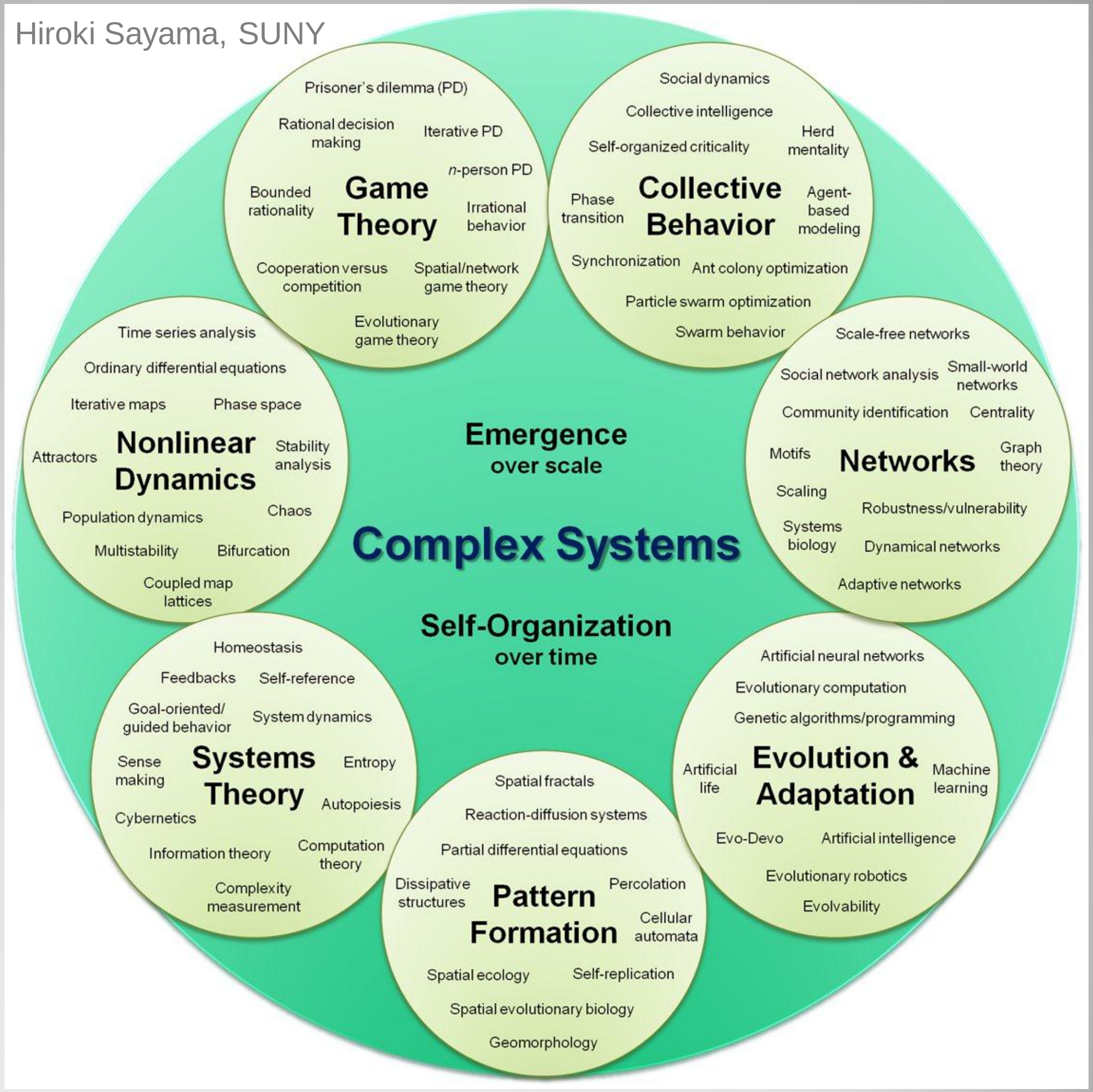
eering Applications
ent Behavior in MANETs
rsidad Distrital

Characteristics of Complex Systems

www.art-sciencefactory.com/complexity-map_feb09.html



Complex Systems Theory



Synchronization in MANETs

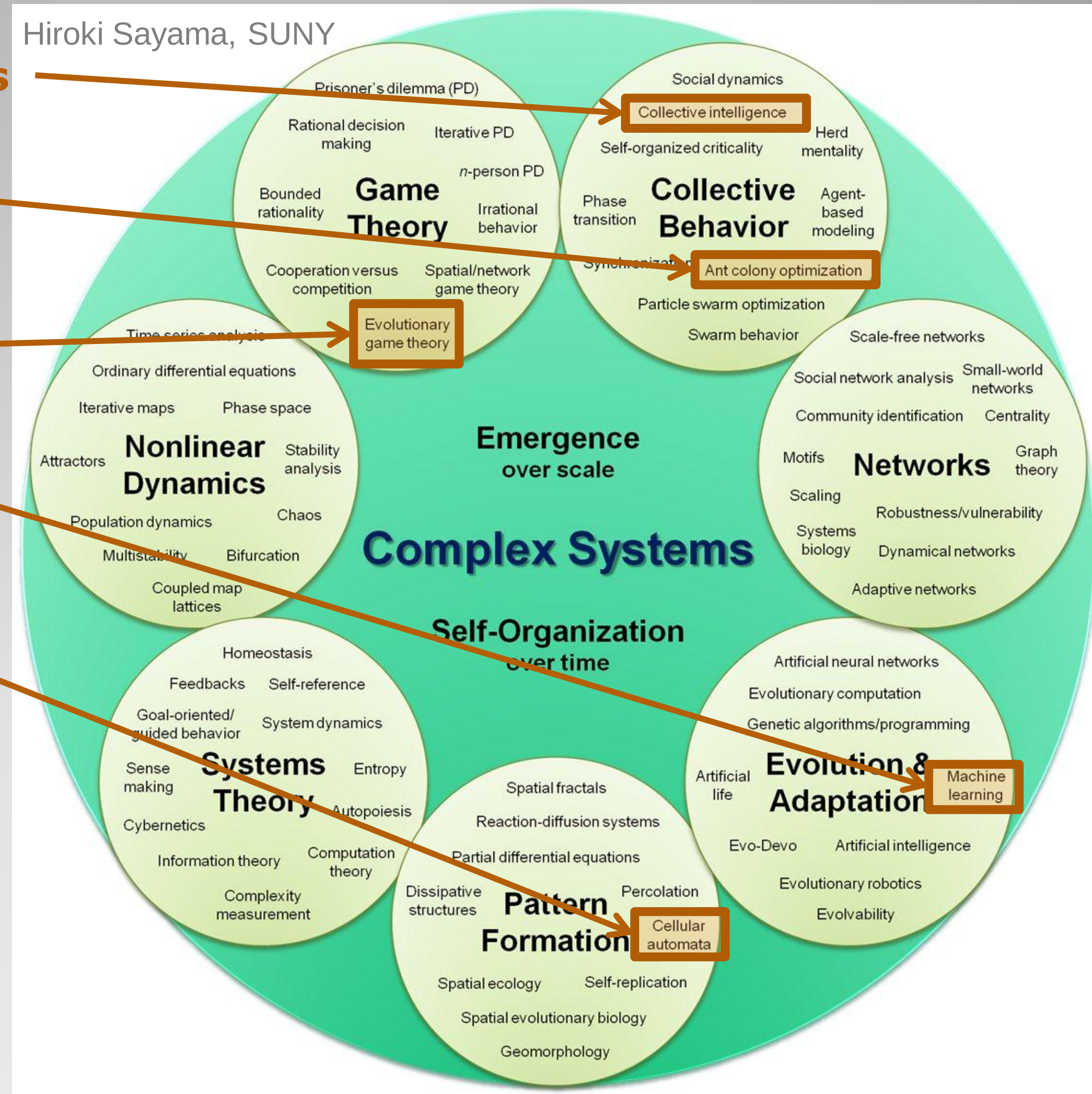
Traffic dispersion over Alternate routes

Cooperation in MANETs

Bandwidth estimation in MANETs

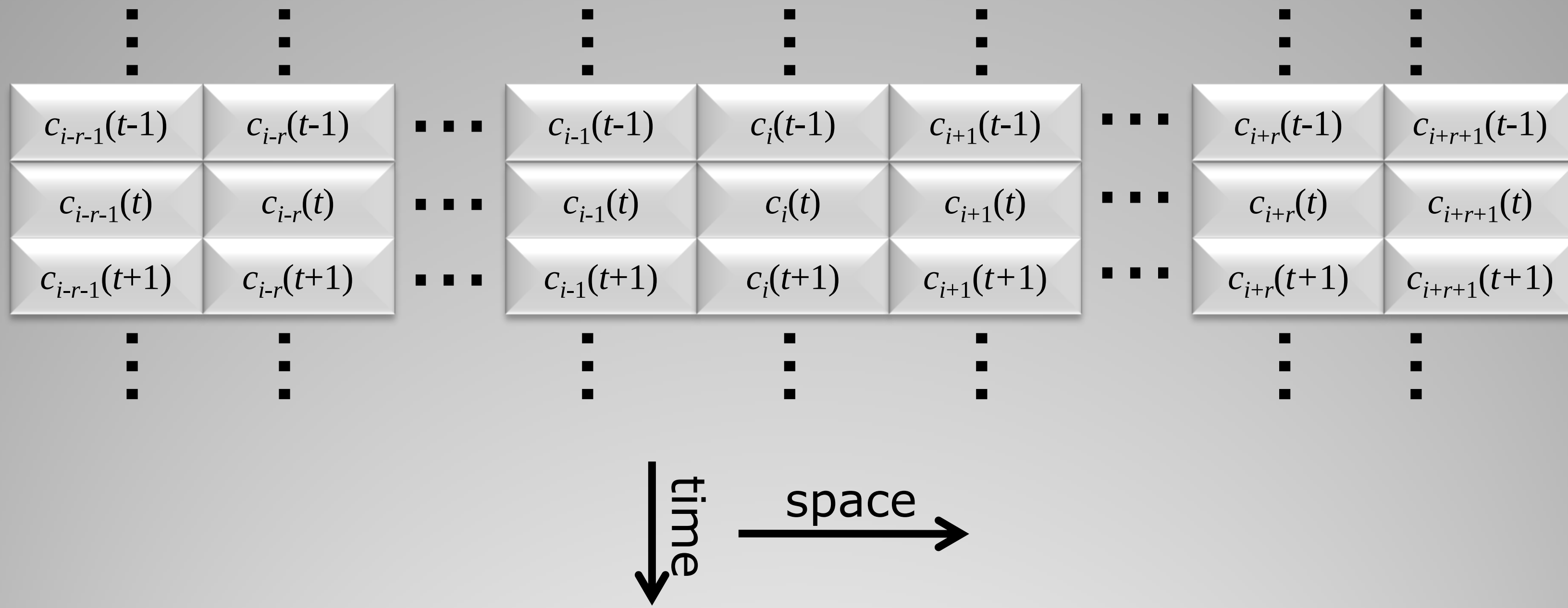
Transmission scheduling in MANETs

Complex Systems Theory



ideas
Investigación, Desarrollo y Aplicaciones en Señales

Example: Cellular automata

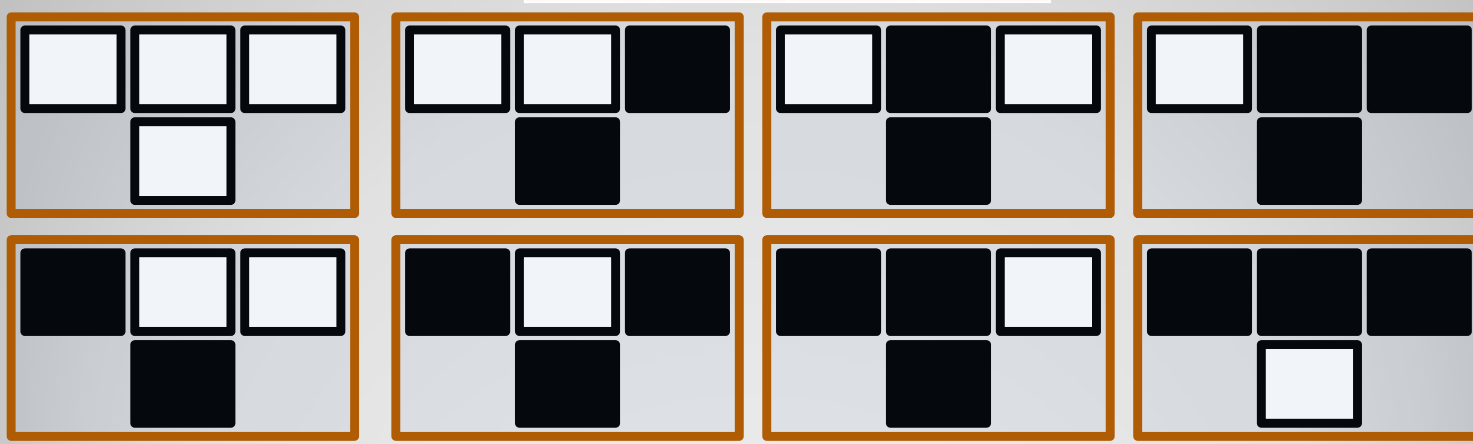


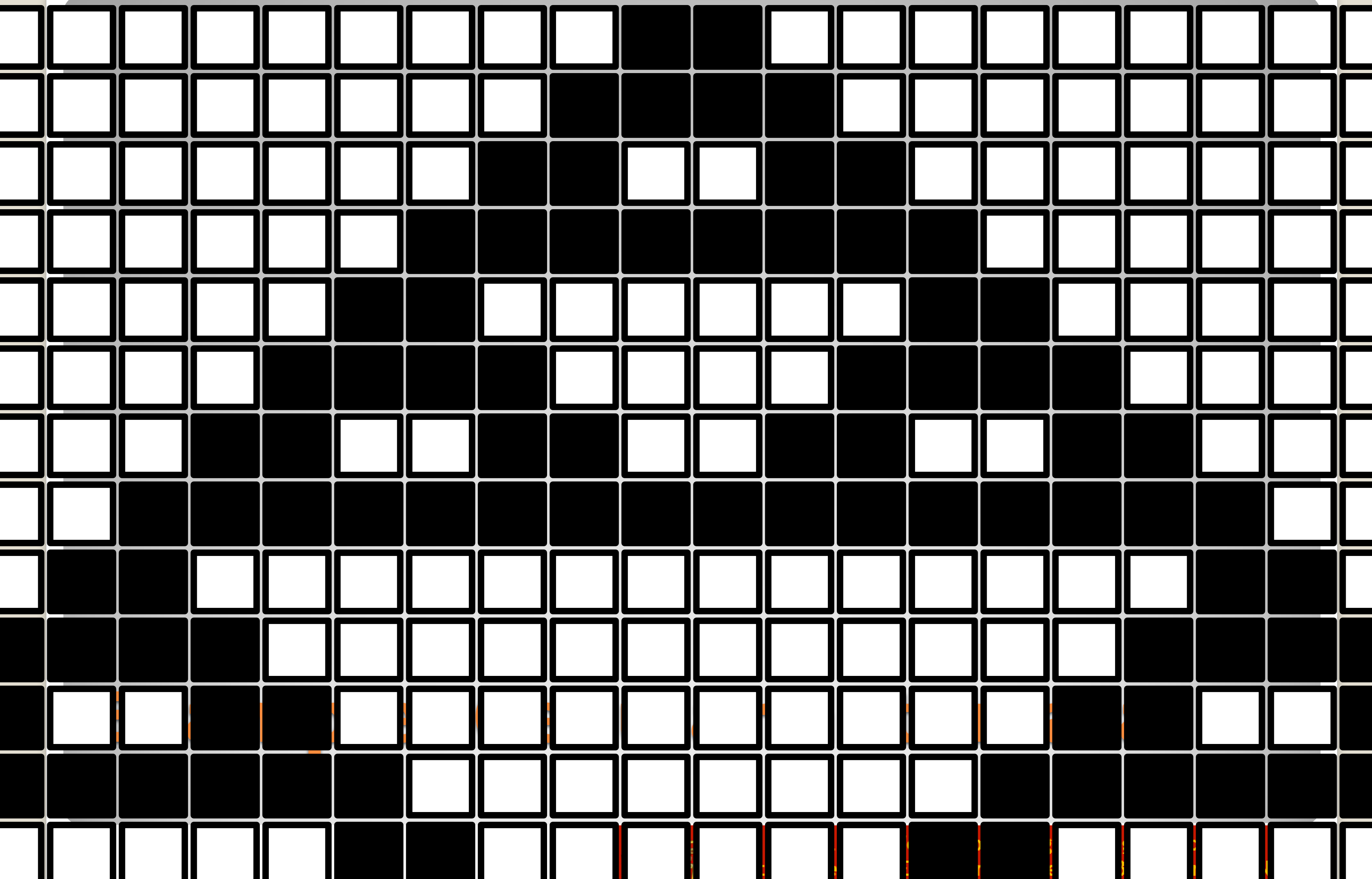
$c_i(t)$ is the state of cell i at instant t

The next state of cell i depends on the current states of the neighbor cells in a radius r

Example: Cellular automata

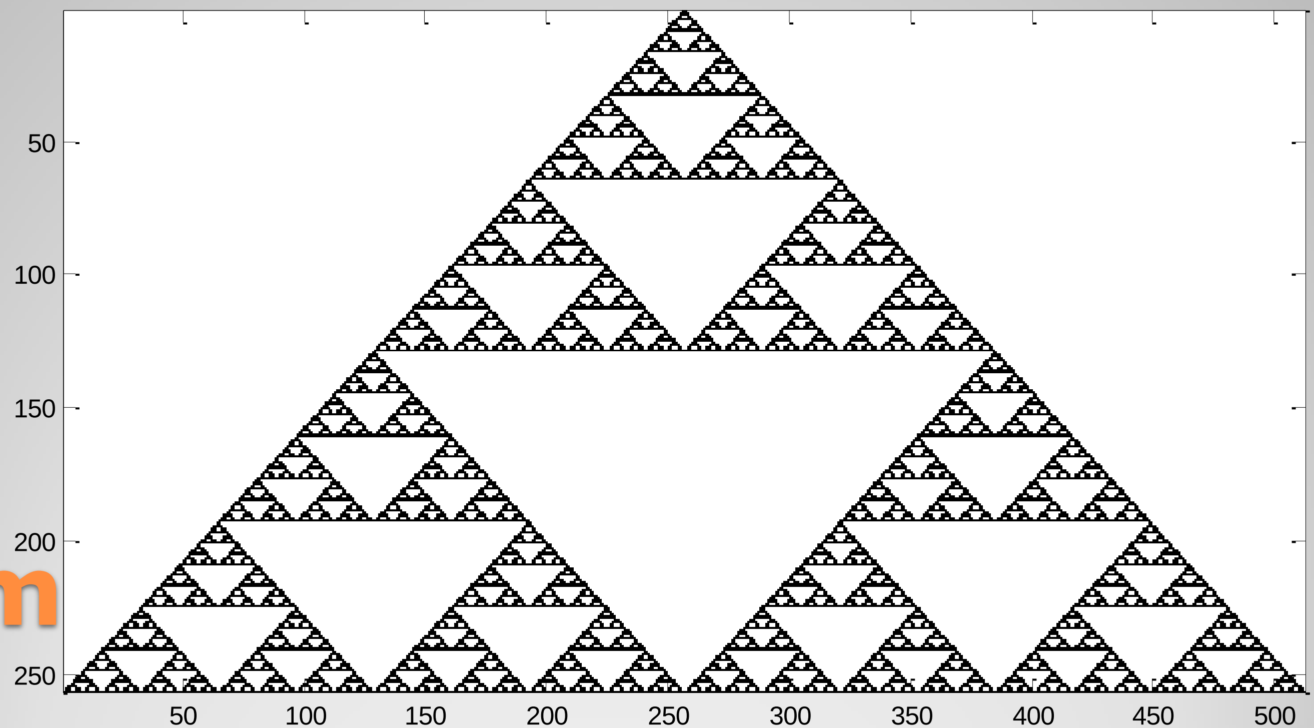
$c_{i-1}(t)$	$c_i(t)$	$c_{i+1}(t)$	$c_i(t+1)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



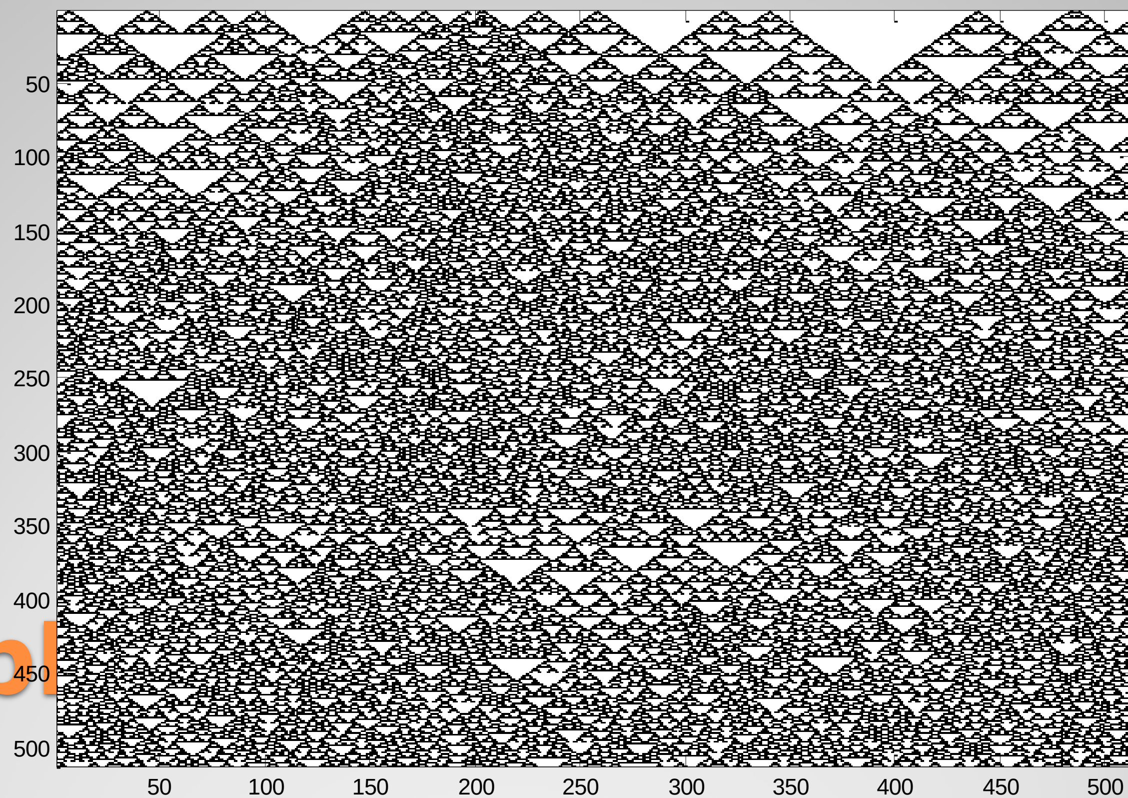


```
colormap([1 1 1; 0 0 0])     % 0 blanco, 1 negro
n = 512; i=2:n-1;            % Tamaño de la línea
c = zeros(n/2,n);            % Historia del autómata
c(1,n/2 + [0 1]) = 1;        % Estado inicial
for t=2:n/2                  % Indice de tiempo
    sum(i) = c(t-1,i-1) + c(t-1,i) + c(t-1,i+1);
    sum(1) = c(t-1,n) + c(t-1,1) + c(t-1,2);
    sum(n) = c(t-1,1) + c(t-1,n) + c(t-1,n-1);
    c(t,:) = (sum<3).*(sum>0); % Evalúa la regla
end
imagesc(c)
```

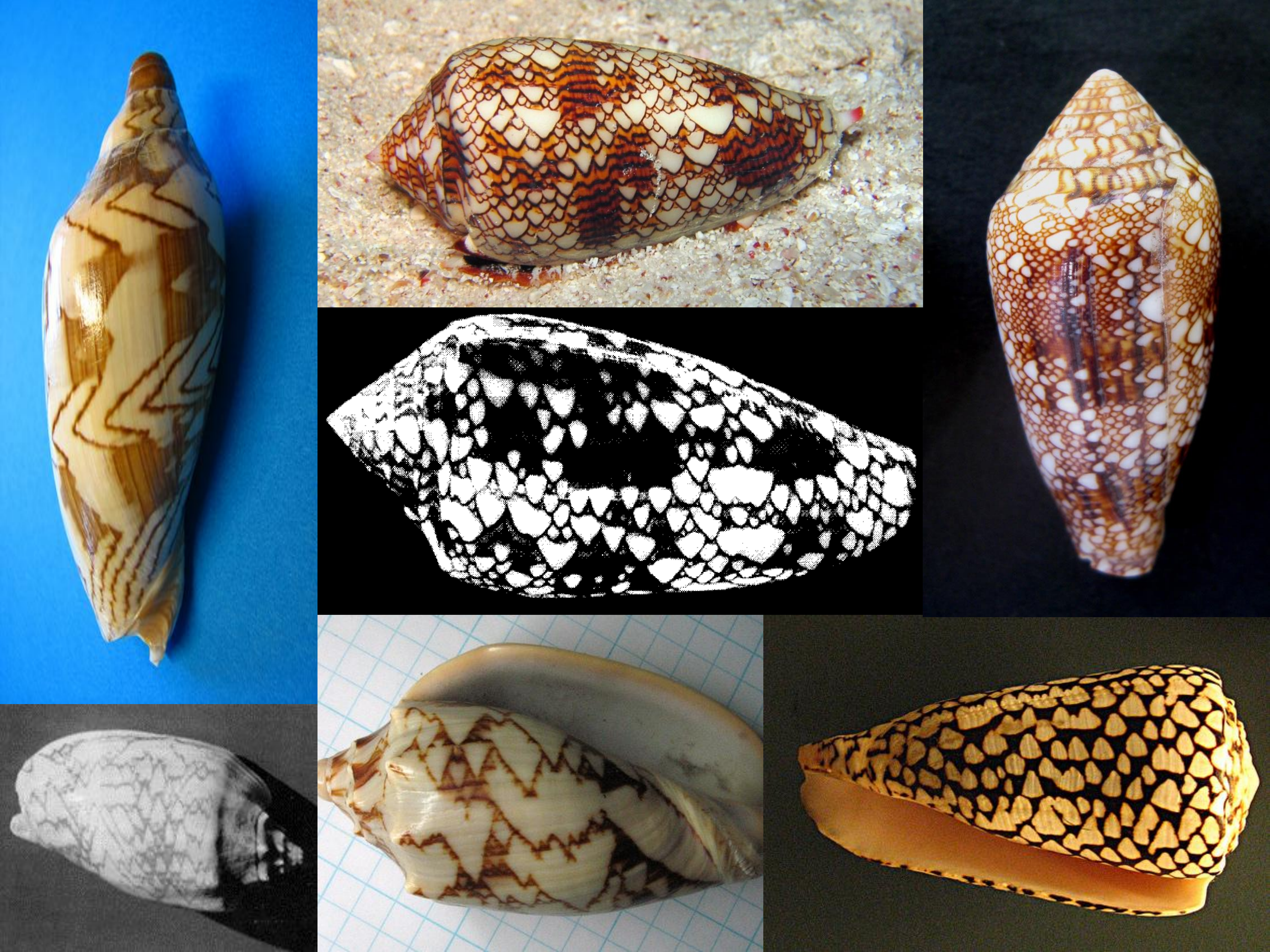
Exam



```
colormap([1 1 1; 0 0 0])    % 0 blanco, 1 negro
n = 512; i=2:n-1;           % Tamaño de la linea
c = zeros(n,n);             % Historia del autómata
c(1,:) = (rand(1,n)>0.95); % Estado inicial
for t=2:n                   % Indice de tiempo
    sum(i) = c(t-1,i-1) + c(t-1,i) + c(t-1,i+1);
    sum(1) = c(t-1,n) + c(t-1,1) + c(t-1,2);
    sum(n) = c(t-1,1) + c(t-1,n) + c(t-1,n-1);
    c(t,:) = (sum<3).*(sum>0); % Evalúa la regla
end
imagesc(c)
```



Exampl a



```
n=64 ;                                 % Tamaño del mundo
mundo = (rand(n,n)<0.5);               % Inicializa el mundo
imagesc(mundo);                        % Visualiza el mundo
colormap([1 1 1; 0 0 0]);              % 0 - blanco, 1 - negro
axis equal
axis tight
while(1)
    mundo = [mundo(n,n) mundo(n,1:n) mundo(n,1); ... % Forma el toroide
             mundo(1:n,n) mundo mundo(1:n,1); ...
             mundo(1,n) mundo(1,1:n) mundo(1,1)];
    % Evalúa la regla de interacción
    suma  =  mundo(1:n,1:n)    + mundo(1:n,2:n+1)    + mundo(1:n,3:n+2) + ...
             mundo(2:n+1,1:n) +                        mundo(2:n+1,3:n+2) + ...
             mundo(3:n+2,1:n) + mundo(3:n+2,2:n+1) + mundo(3:n+2,3:n+2);
    mundo = ((suma==3) + (suma==2).*mundo(2:n+1,2:n+1));
    imagesc(mundo);                   % Actualiza la imagen
    drawnow
end
```

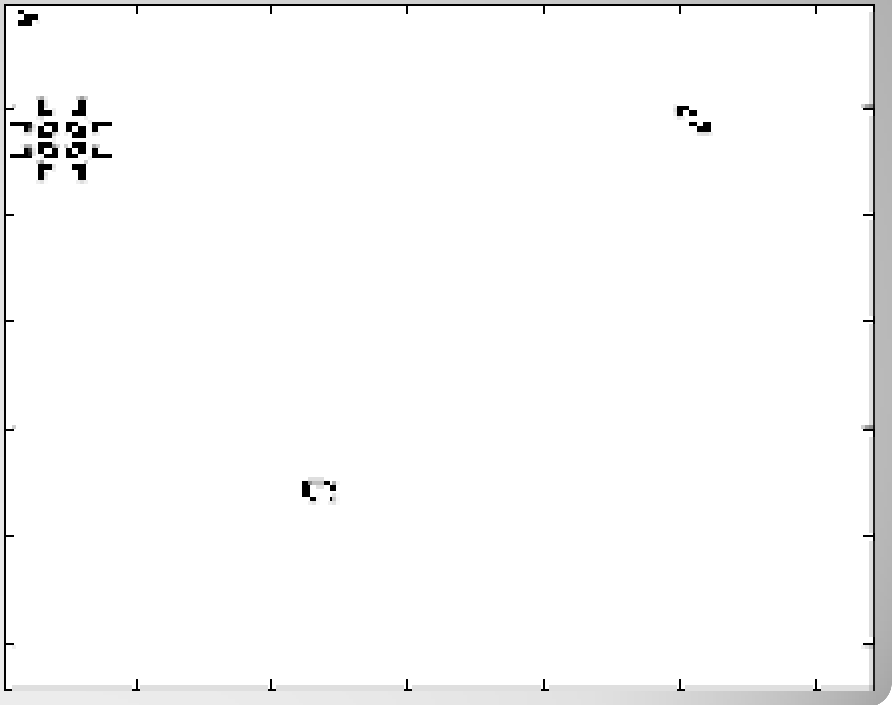
Game of life

```
n = 128;
mundo = zeros(n,n);
mundo(2:4,2:4) = [0 0 1; 1 0 1; 0 1 1];
mundo(20:32,3:15) = [0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     1 0 0 0 0 1 0 1 0 0 0 0 1;...
                     0 0 0 0 0 0 0 0 0 0 0 0 0;...
                     0 0 1 1 1 0 0 0 1 1 1 0 0];

mundo(90:93,45:49) = [0 1 1 0 0;...
                      1 1 0 1 1;...
                      0 1 1 1 1;...
                      0 0 1 1 0];

mundo(20:24,100:104) = [1 1 0 0 0;...
                        1 0 0 0 0;...
                        0 1 0 1 0;...
                        0 0 0 0 1;...
                        0 0 0 1 1];
```

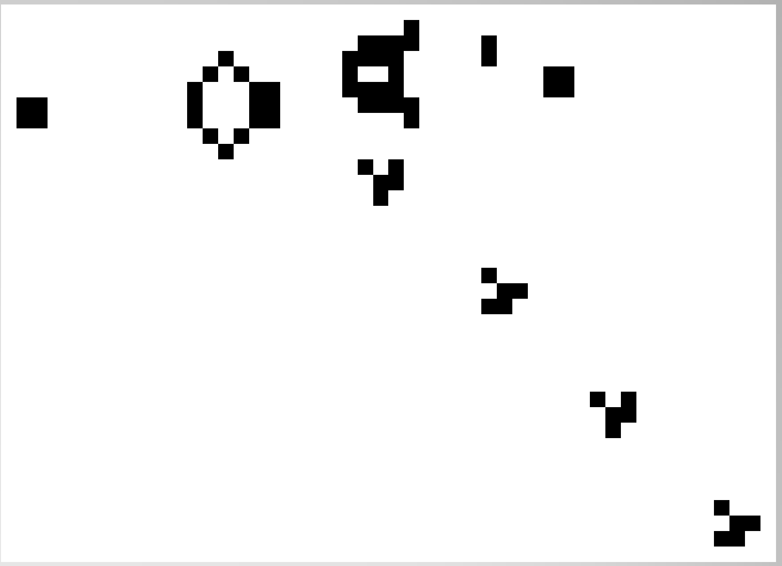
Oscilation and displacement



Game of life

```
mundo = zeros(n,n);                    % Inicializa el mundo
mundo(2:10,2:37) = [0 0 0 0 1 1 0 0 0; ...
                    0 0 0 0 1 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 1 1 1 0 0; ...
                    0 0 0 1 0 0 0 1 0; ...
                    0 0 1 0 0 0 0 0 1; ...
                    0 0 1 0 0 0 0 0 1; ...
                    0 0 0 0 0 1 0 0 0; ...
                    0 0 0 1 0 0 0 1 0; ...
                    0 0 0 0 1 1 1 0 0; ...
                    0 0 0 0 0 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 1 1 1 0 0 0 0; ...
                    0 0 1 1 1 0 0 0 0; ...
                    0 1 0 0 0 1 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    1 1 0 0 0 1 1 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 0 0 0 0 0 0 0; ...
                    0 0 1 1 0 0 0 0 0; ...
                    0 0 1 1 0 0 0 0 0]';
```

Fabrication



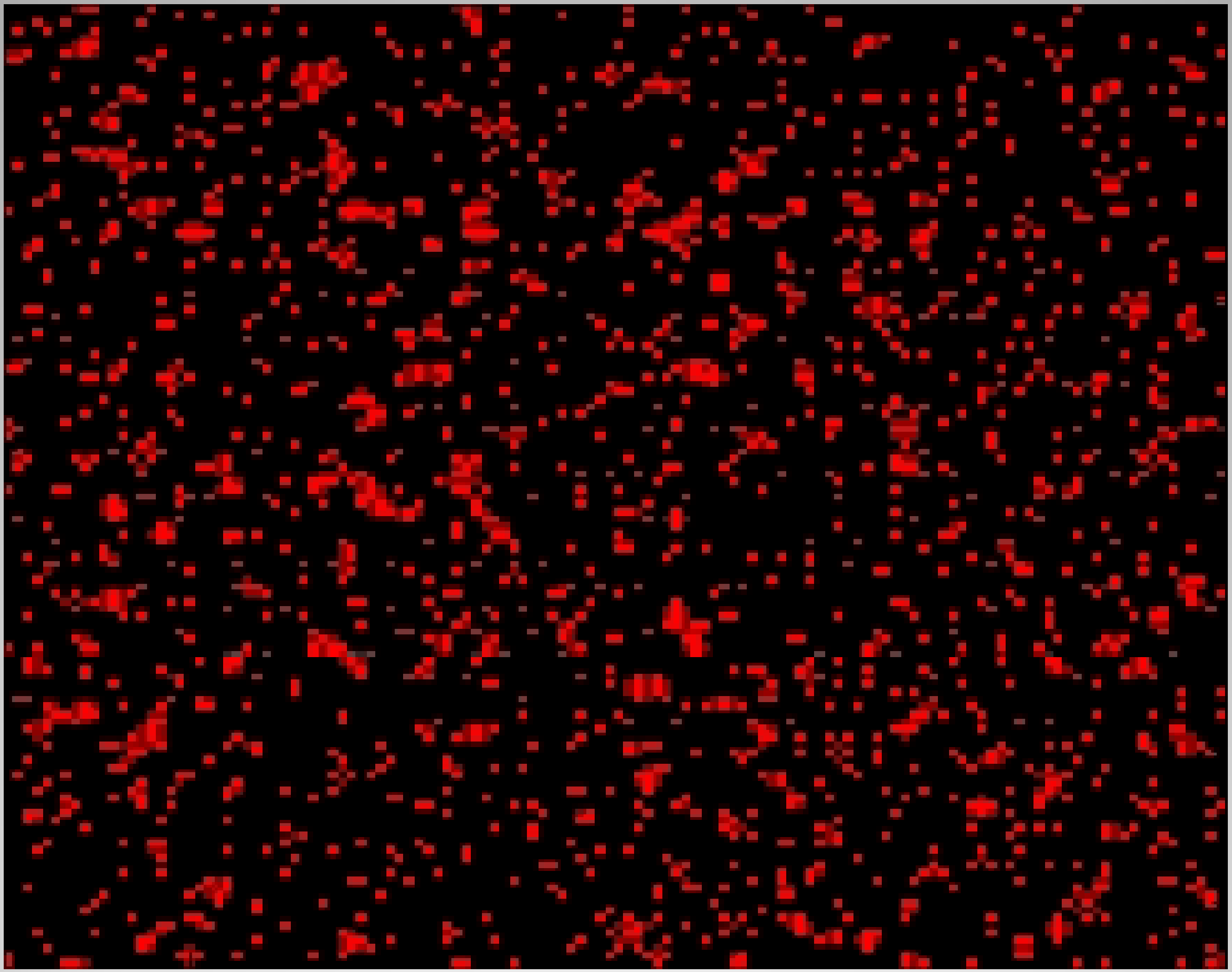
Game of life

Why Game of "Life"?
Self-reproduction!

Game of life

```
n=128 ;                              % Tamaño del mundo
mundo = rand(n,n)>0.9;               % Inicializa el mundo
imagesc(mundo);                      % Visualiza el mundo
colormap([(0:9)'/9 zeros(10,2)]);    % Niveles de rojo
axis equal
axis tight
while(1)
    mundo = [mundo(n,n)    mundo(n,1:n) mundo(n,1);    ... % Forma el toroide
             mundo(1:n,n)  mundo        mundo(1:n,1); ...
             mundo(1,n)    mundo(1,1:n) mundo(1,1)];
    % Evalúa la regla de interacción
    suma   =  ((mundo(1:n,1:n)>0)      .* (mundo(1:n,1:n)<6)     + ...
              (mundo(1:n,2:n+1)>0)    .* (mundo(1:n,2:n+1)<6)    + ...
              (mundo(1:n,3:n+2)>0)    .* (mundo(1:n,3:n+2)<6)    + ...
              (mundo(2:n+1,1:n)>0)    .* (mundo(2:n+1,1:n)<6)    + ...
              (mundo(2:n+1,3:n+2)>0)  .* (mundo(2:n+1,3:n+2)<6)  + ...
              (mundo(3:n+2,1:n)>0)    .* (mundo(3:n+2,1:n)<6)    + ...
              (mundo(3:n+2,2:n+1)>0)  .* (mundo(3:n+2,2:n+1)<6)  + ...
              (mundo(3:n+2,3:n+2)>0)  .* (mundo(3:n+2,3:n+2)<6));
    mundo = mod(mundo(2:n+1,2:n+1) + 1,10);
    mundo = mundo.*(mundo~=1) + (mundo==1).*(suma>2);
    imagesc(mundo);                  % Actualiza la imagen
    drawnow
end
```

Excitable media



Excitable media

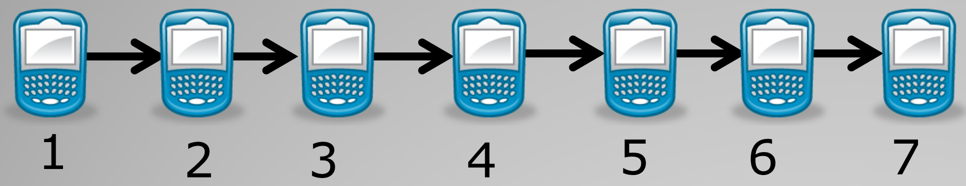
http://www.youtube.com/watch?v=IBa4kgXI4Cg&feature=related

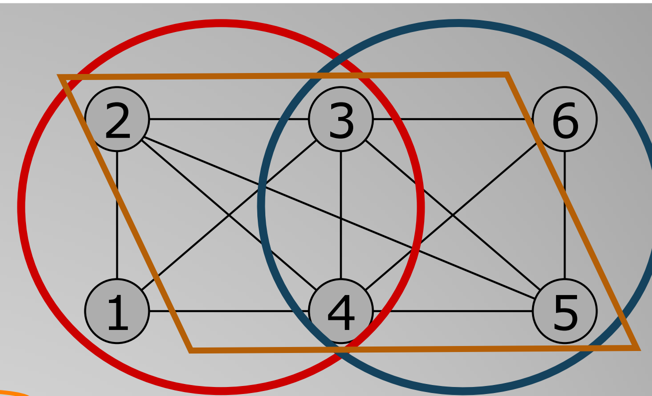
Excitable media

Applied in many different areas

- Statistical Mechanics
- Autocatalytical chemical sets
- Gene regulation
- Multicellular organisms
- Colonies and super-organisms
- Flocks, schools, herds
- Ecosystems
- Economics and Society
- …
- Two toy Network examples: Optimal scheduling and traffic dispersion

Emergent Perfect Scheduling



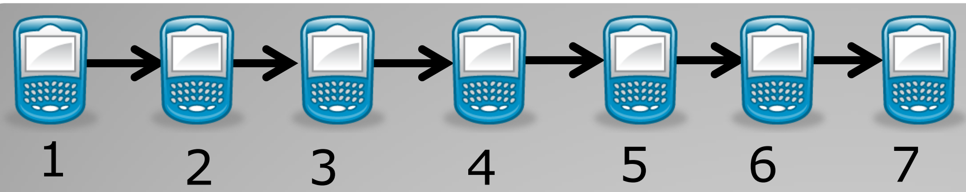


Possible states in a time slot

1. Silence and no activity perception
2. Successful transmission to the next node to the right
3. Successful reception from the next node to the left
4. Detection of a transmission from the next node to the right (to a distant destination)
5. Failed transmission due to a collision
6. Detection of a collision in the neighborhood

Use the state in the previous three time slots to decide whether to transmit in the current slot

ideas
Investigación, Desarrollo y Aplicaciones en Señales



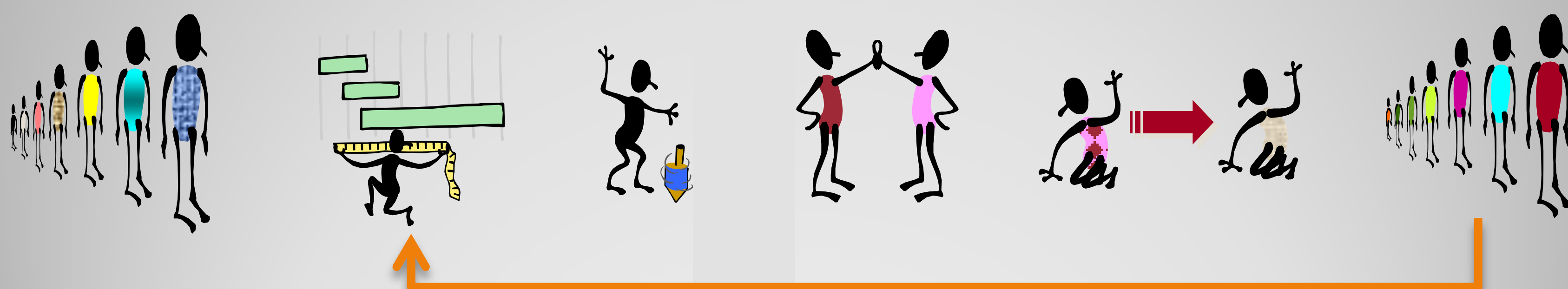
Possible states in a time slot

1. Silence and no activity perception
2. Successful transmission to the next node to the right
3. Successful reception from the next node to the left
4. Detection of a transmission from the next node to the right (to a distant destination)
5. Failed transmission due to a collision
6. Detection of a collision in the neighborhood

216 (=6^3) possible sequences of three consecutive previous states

$2^{216} \approx 10^{65}$ possible rules

Each rule is a sequence of 216 binary genes: Genetic Evolution



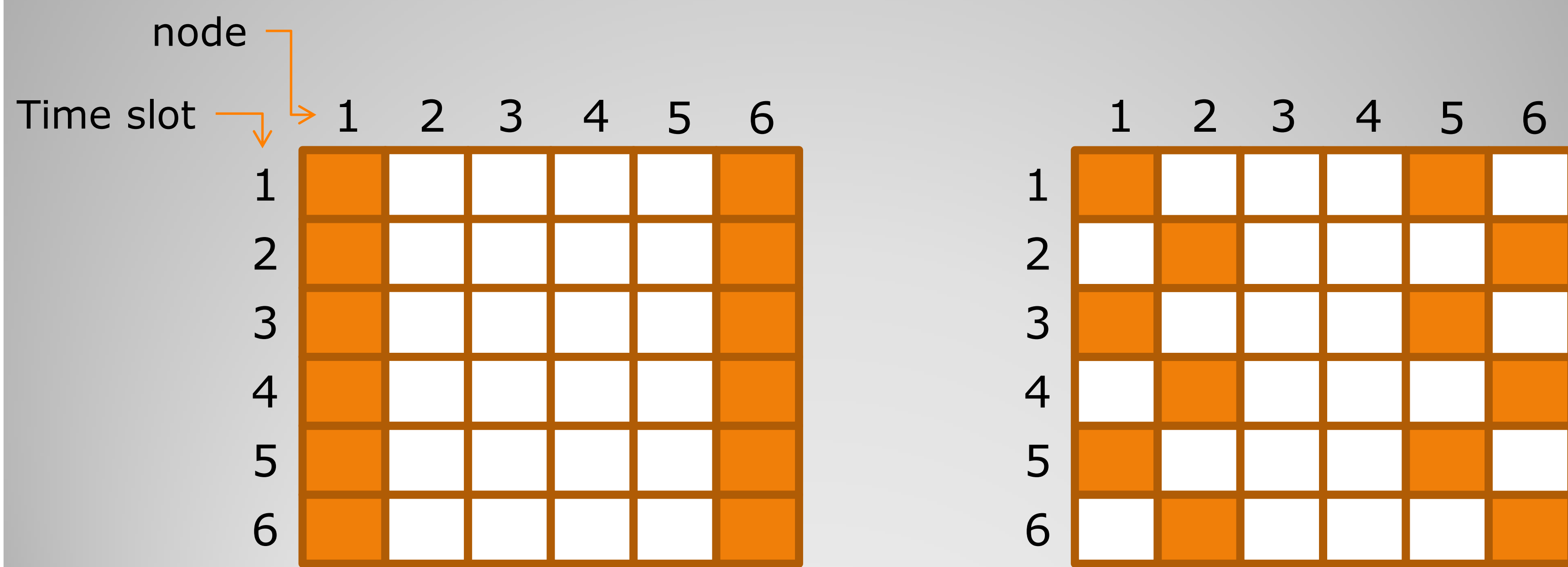
Emergent Perfect Scheduling

2012 IEEE Workshop on Engineering Applications
Tutorial Session: Engineering Emergent Behavior in MANETs
Marco A. Alzate, Universidad Distrital

Maximize the throughput

$$\max\left[\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{6}\sum_{n=1}^{6}e_n(t)\right)\right]$$

where $e_n(t)\in\{0,1\}$ is the number of successful transmissions the n^{th} node achieves in the t^{th} slot

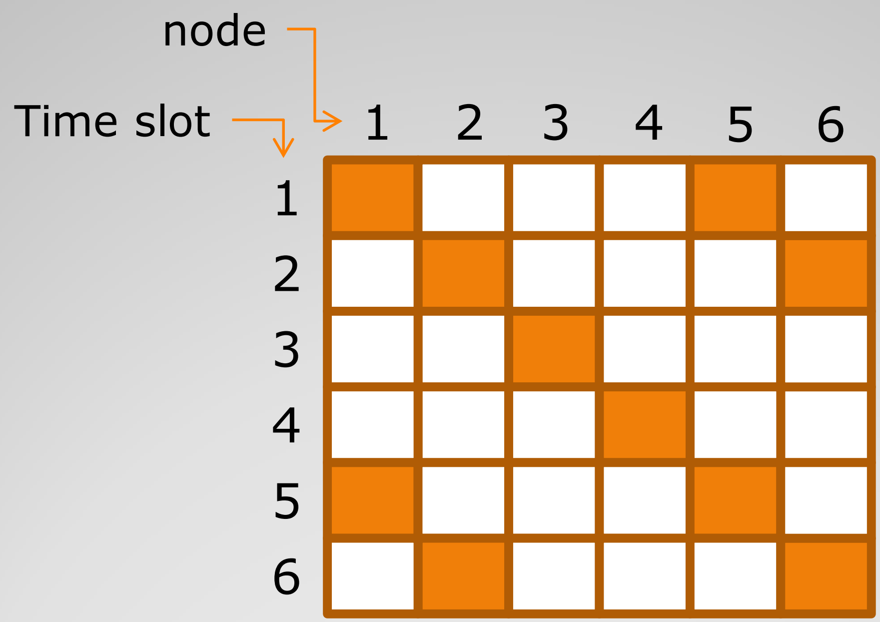


Emergent Perfect Scheduling

Add some restrictions

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} e_n(t) > 0, \quad n = 1,2,3,4,5,6$$

Punish Collisions



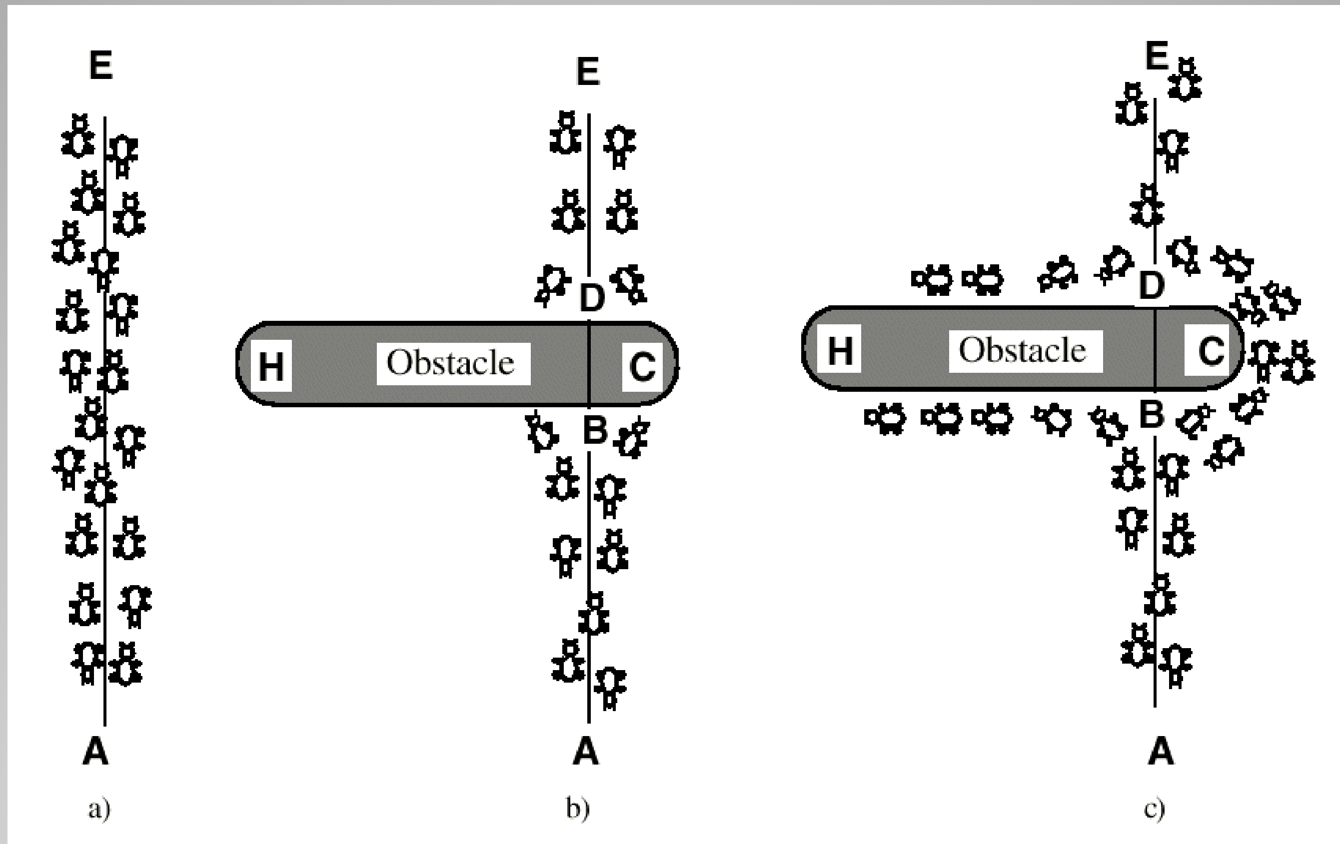
Emergent Perfect Scheduling

A toy model… but a suggestive one

Convergence toward the optimal pattern for different initial states
14240A046046492865F8830CD3940283836221044282 78A430188C

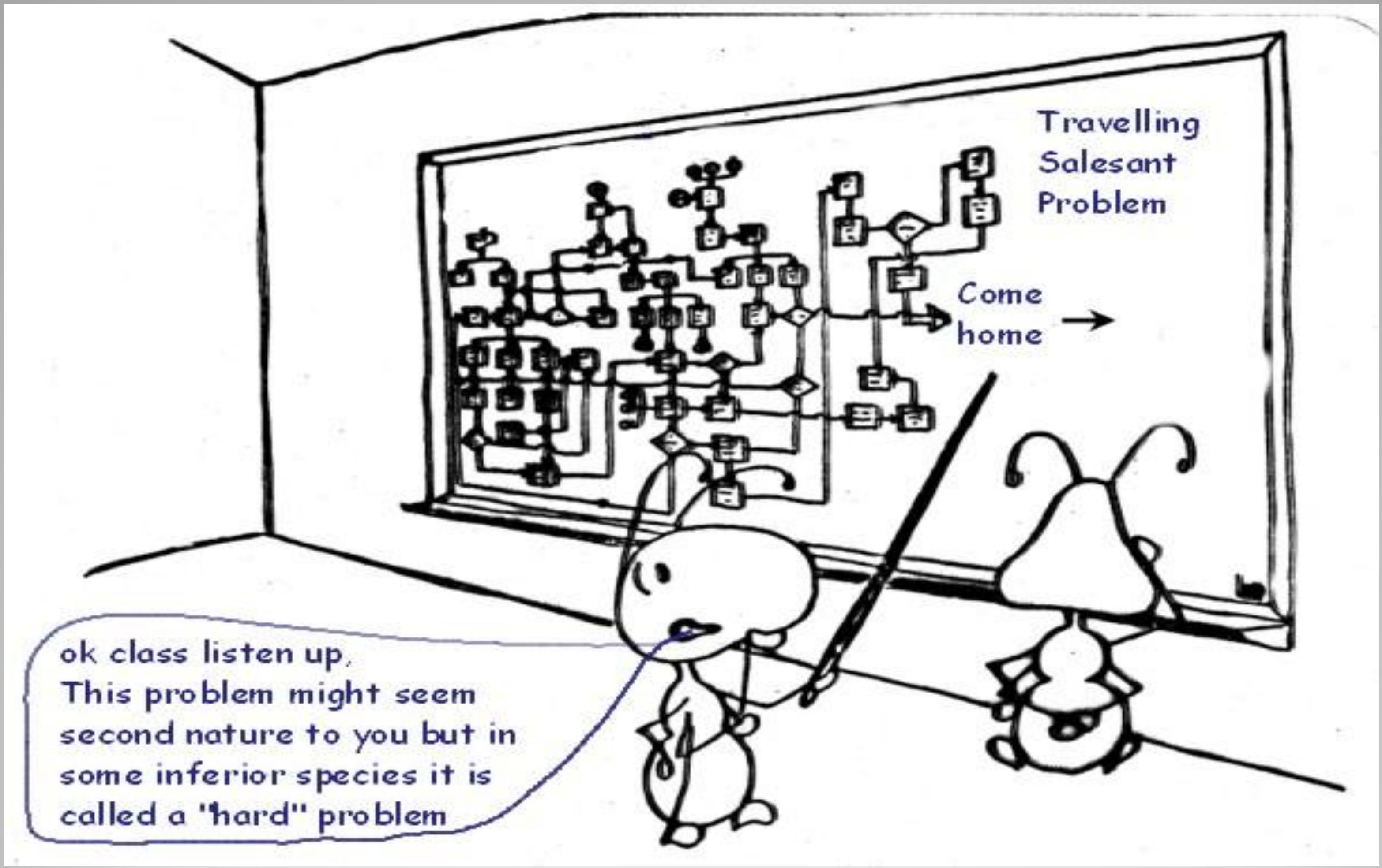
Emergent Perfect Scheduling

http://ugpro143.blogspot.com/2009/11/ant-colony-optimization.html

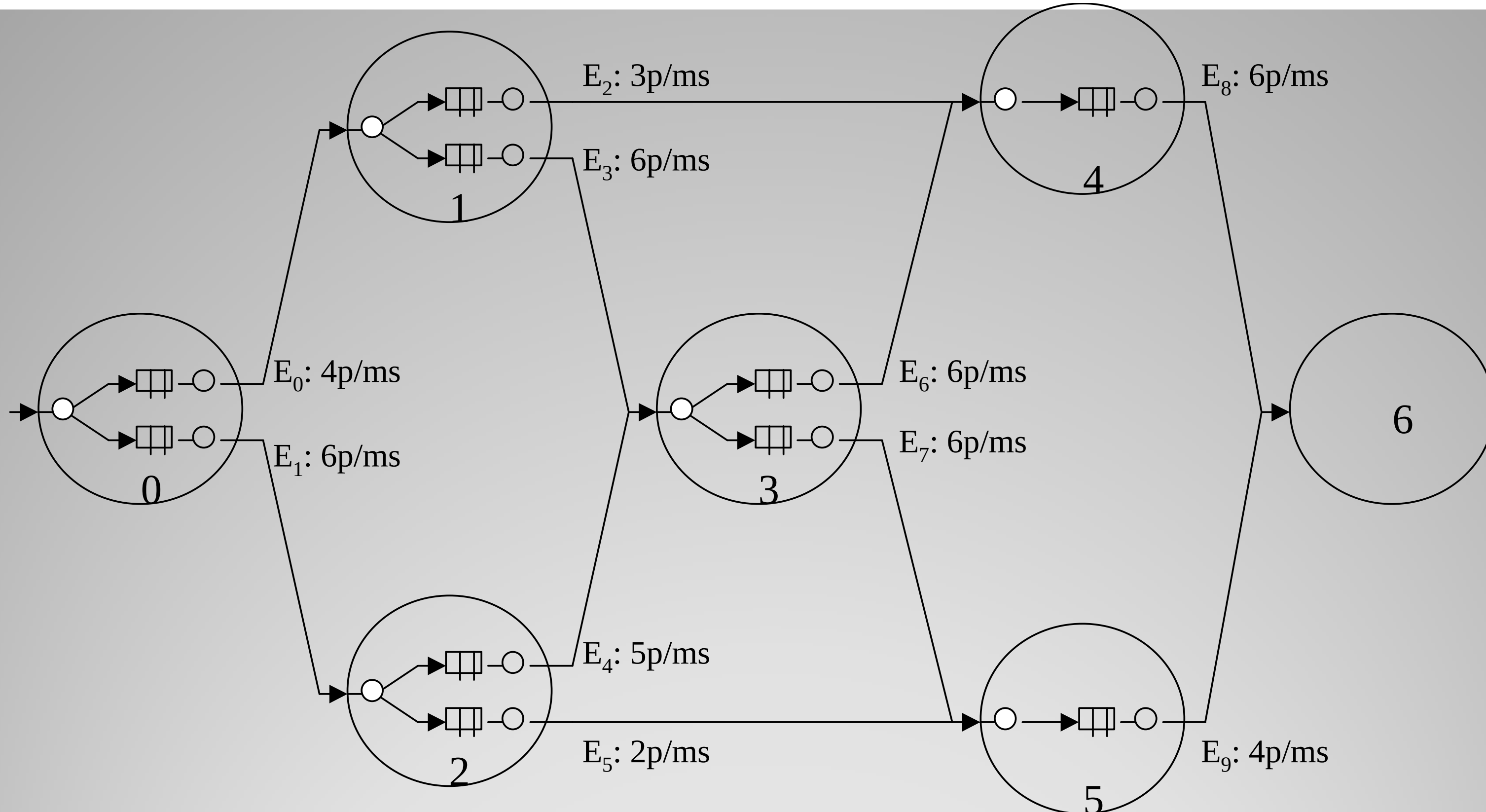


Estigmergy, Evaporation, Randomness

Optimal packet dispersion

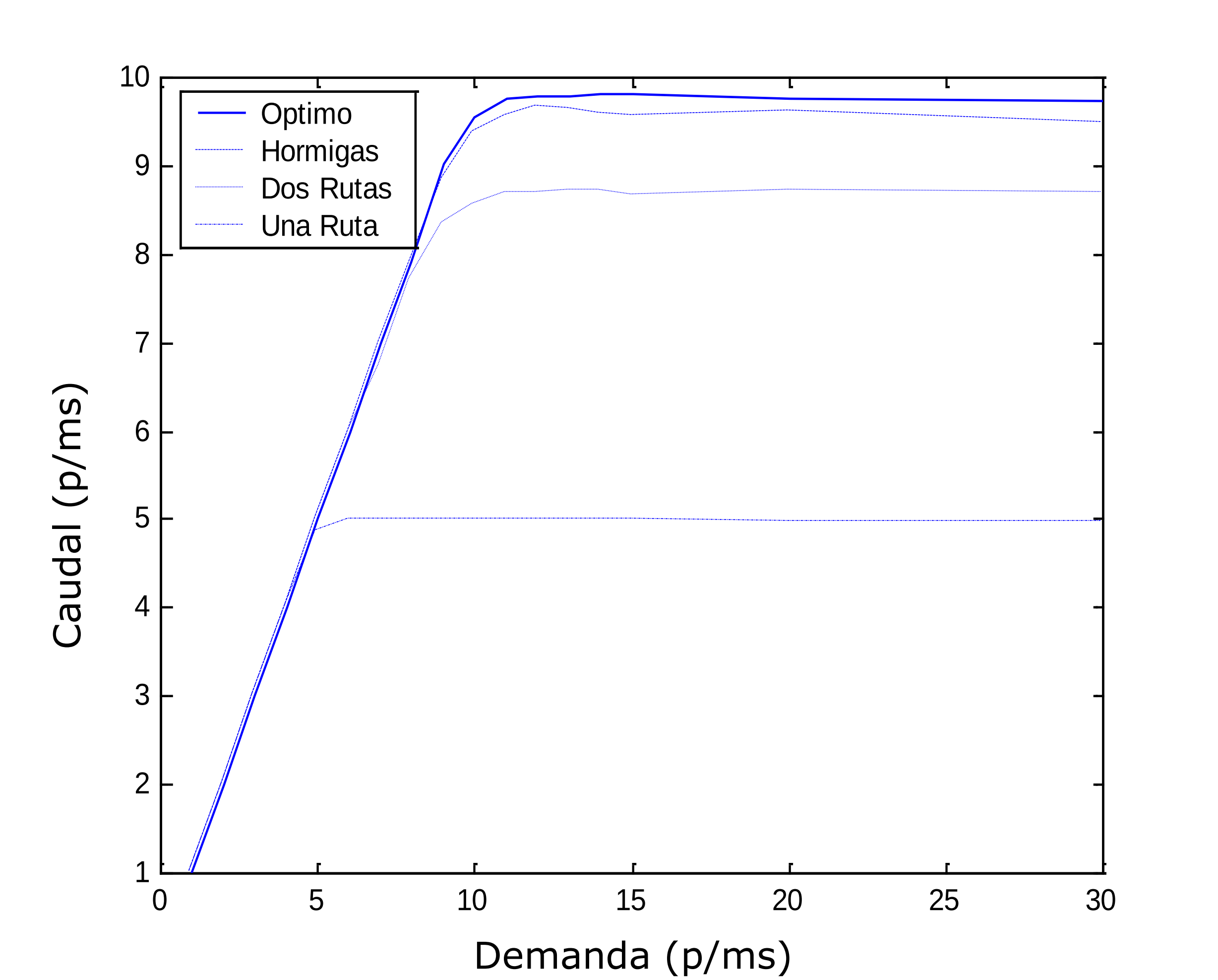


Optimal packet dispersion

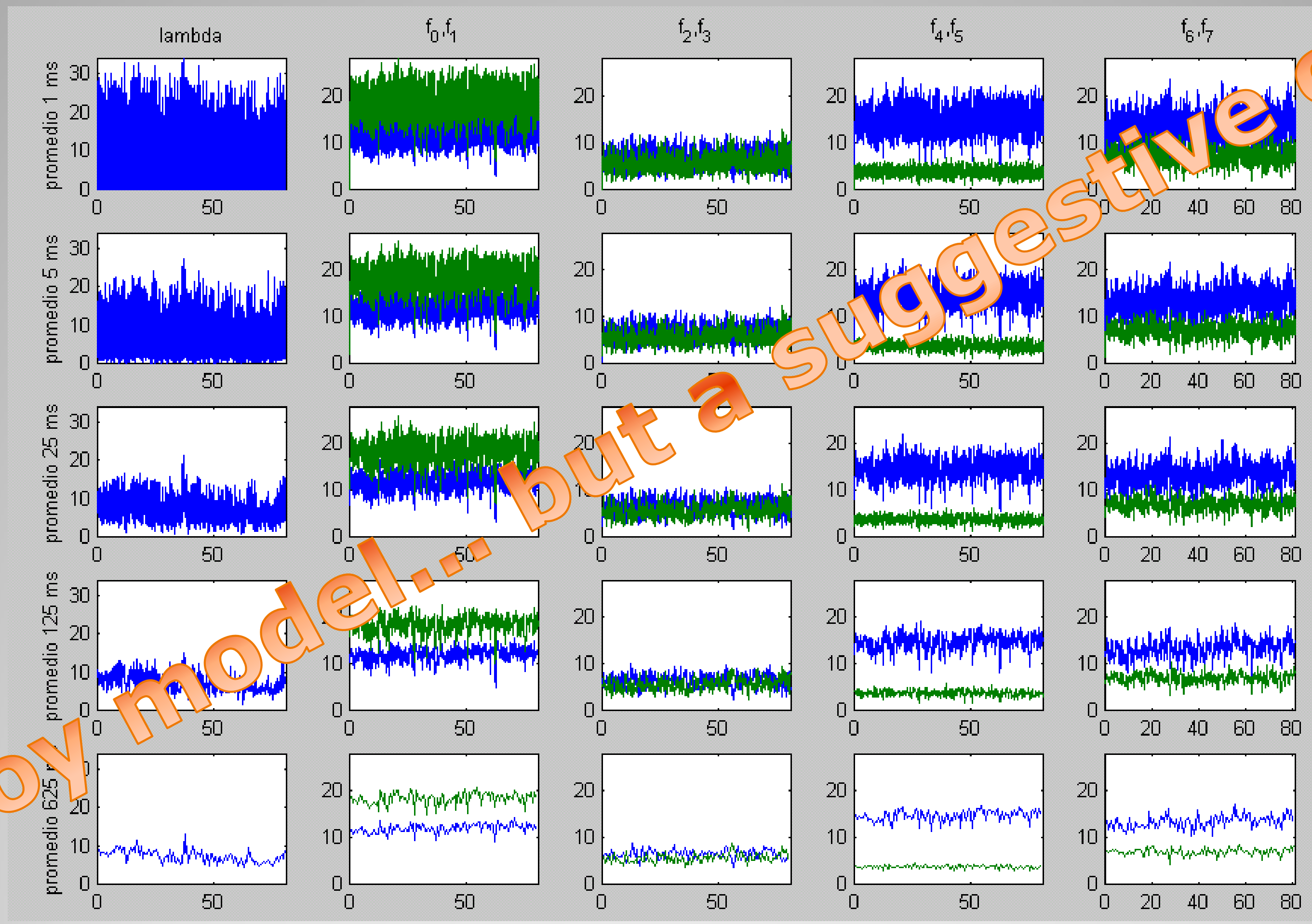


$$P(t,n_i,n_j)=\frac{\tau(t,n_i,n_j)}{\sum_{n_k}\tau(t,n_i,n_k)} \quad \forall n_j \in \{n_k\}$$

Optimal packet dispersion



Optimal packet dispersion



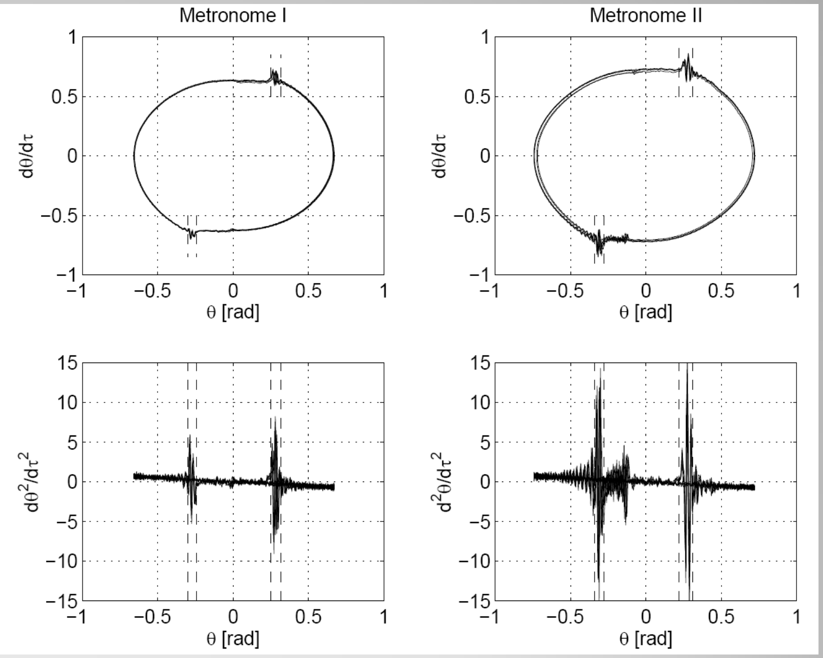
A toy model... but a suggestive one

Optimal packet dispersion

- Introduction to Mobile Ad Hoc Networks
- Introduction to Complex Adaptive Systems
- **Emergent Synchronization in MANETs**
- Emergent Cooperation in MANETs
- Formal methods for engineering emergent behavior in MANETs

Schedule

- Spontaneous synchronization occurs ubiquitously in nature (from particles, to atoms, to molecules, to cells, to insects, to humans, to planets, to galaxies, …)



W.T. Oud, H. Nijmeijer and A.Yu. Pogromsky "A Study of Huijgens' Synchronization: Experimental Results", Lecture Notes in Control and Information Sciences, Springer, 2006

Spontaneous Synchronization

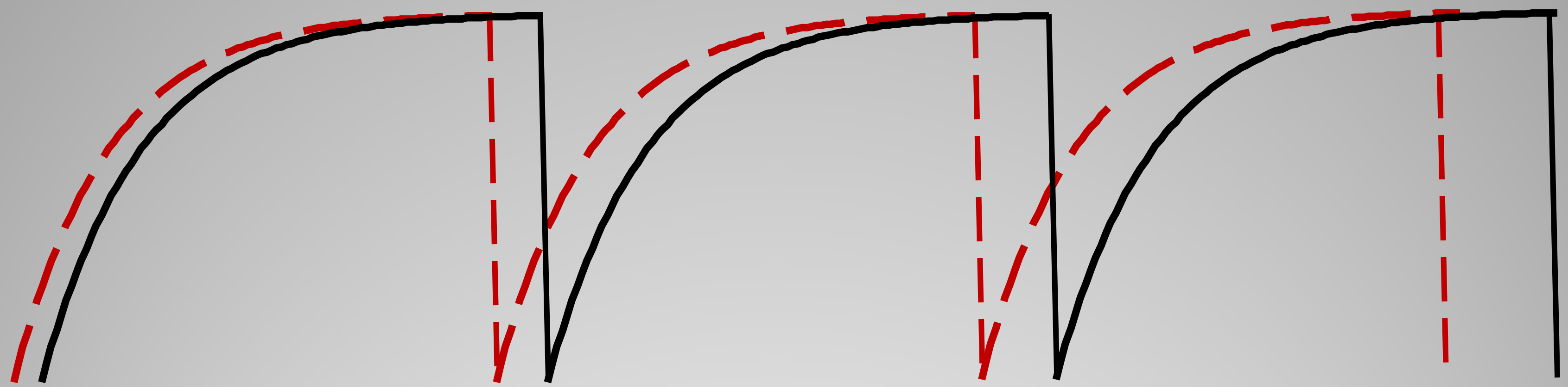
Spontaneous Synchronization

ideas
Investigación, Desarrollo y Aplicaciones en Señales

WEA
2012 IEEE Workshop on Engineering Applications
Tutorial Session: Engineering Emergent Behavior in MANETs
Marco A. Alzate, Universidad Distrital

http://www.youtube.com/watch?v=sROKYelaWbo

Each oscillator follows a simple law of "Integrate and fire"



Then they interact through a simple rule: When an oscillator fires, neighbors state advances a small quantity ε

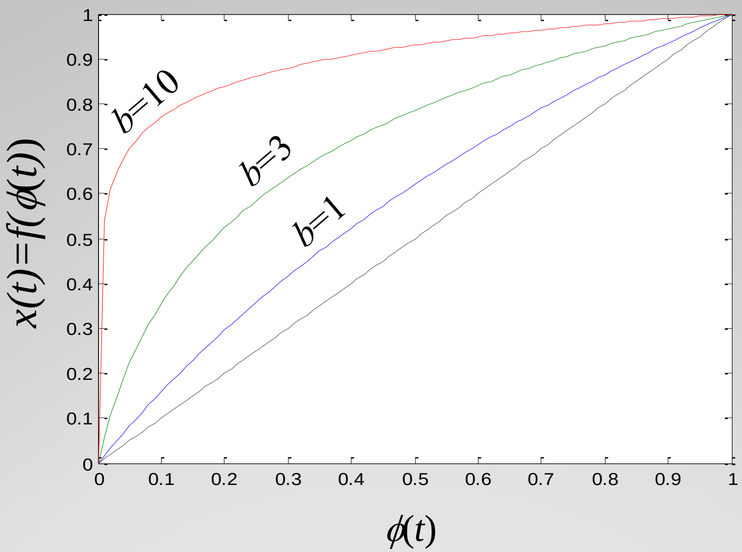
$$x_i(t) = 1 \Rightarrow \begin{cases} x_i(t^+) = 0 \\ x_j(t^+) = \min(1, x_j(t^-) + \varepsilon) \quad j \text{ neighbor of } i, j \neq i \end{cases}$$

R. Mirollo and S. Strogatz, "Synchronization of pulse-coupled biological oscillators", SIAM Journal of Applied Mathematics, December 1990

Firefly Spontaneous Synchronization

At instant t, an oscilador is in a fraction ϕ of its phase,

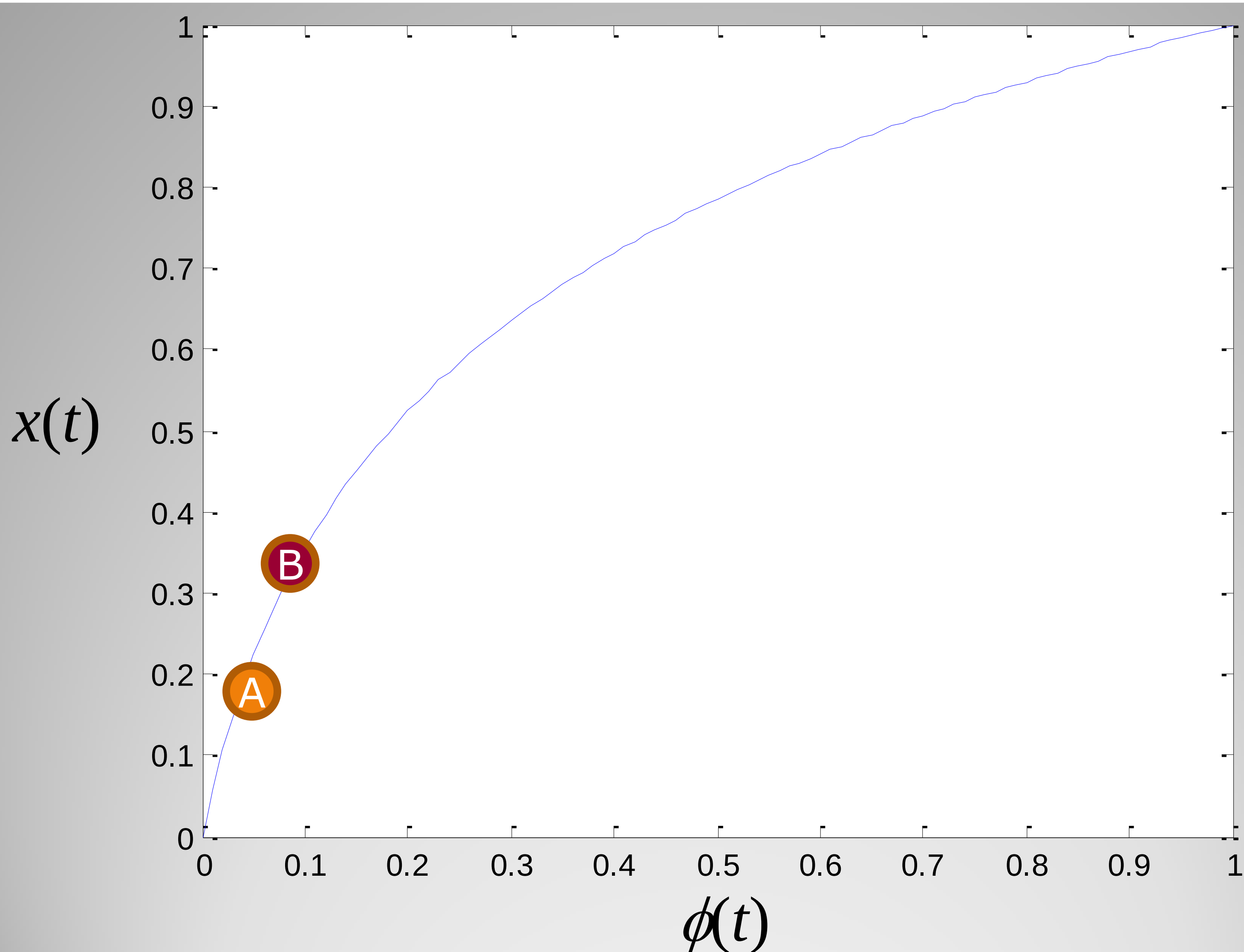
$$x(t) = f(\phi(t)) = \frac{1}{b}\log\left(1+\left(e^{b}-1\right)\phi\right)$$



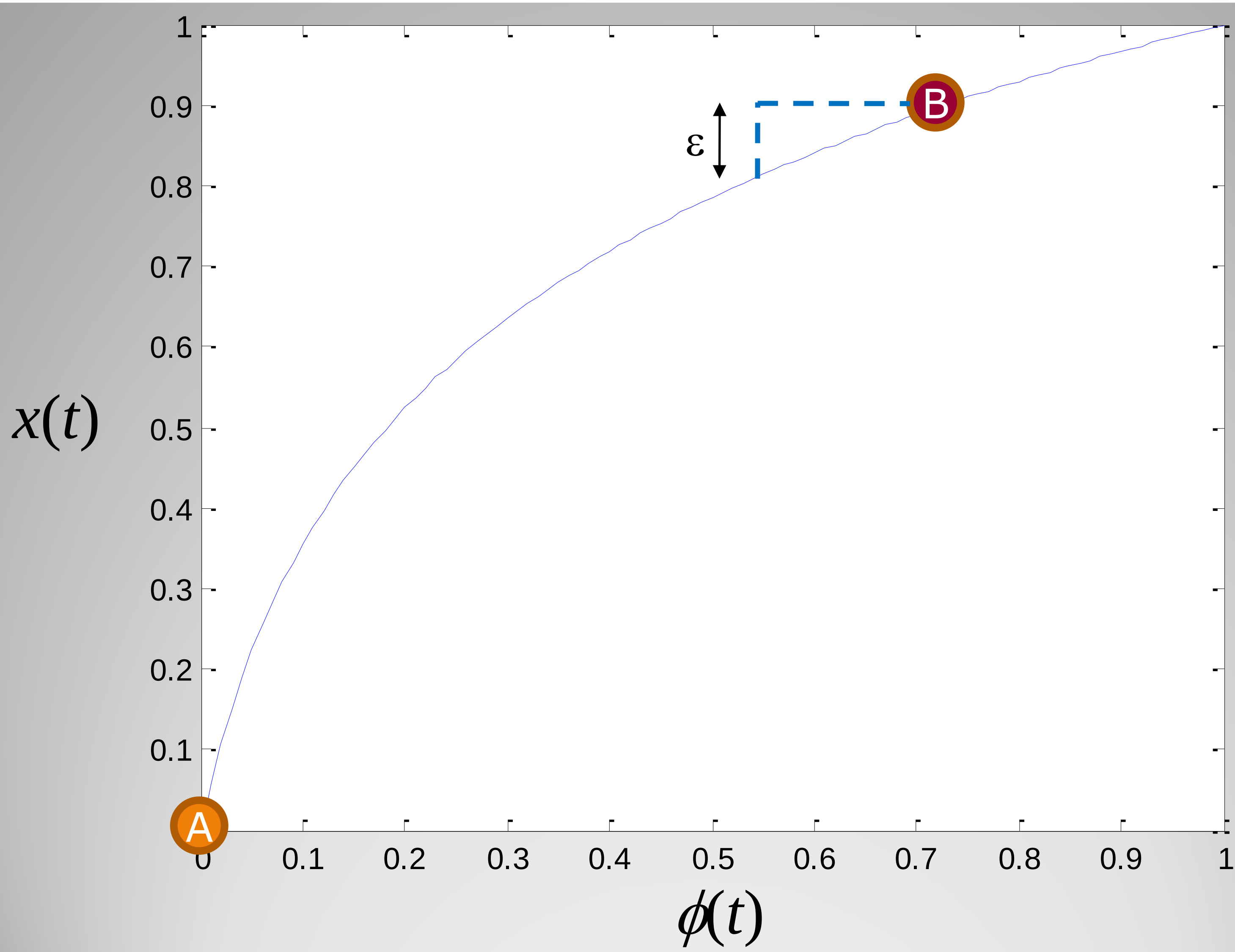
When the oscillator is in state $x(t)$, it is in a fraction ϕ of its phase,

$$\phi(t) = g(x(t)) = g\left(f(\phi(t))\right) = \frac{e^{bx}-1}{e^{b}-1}$$

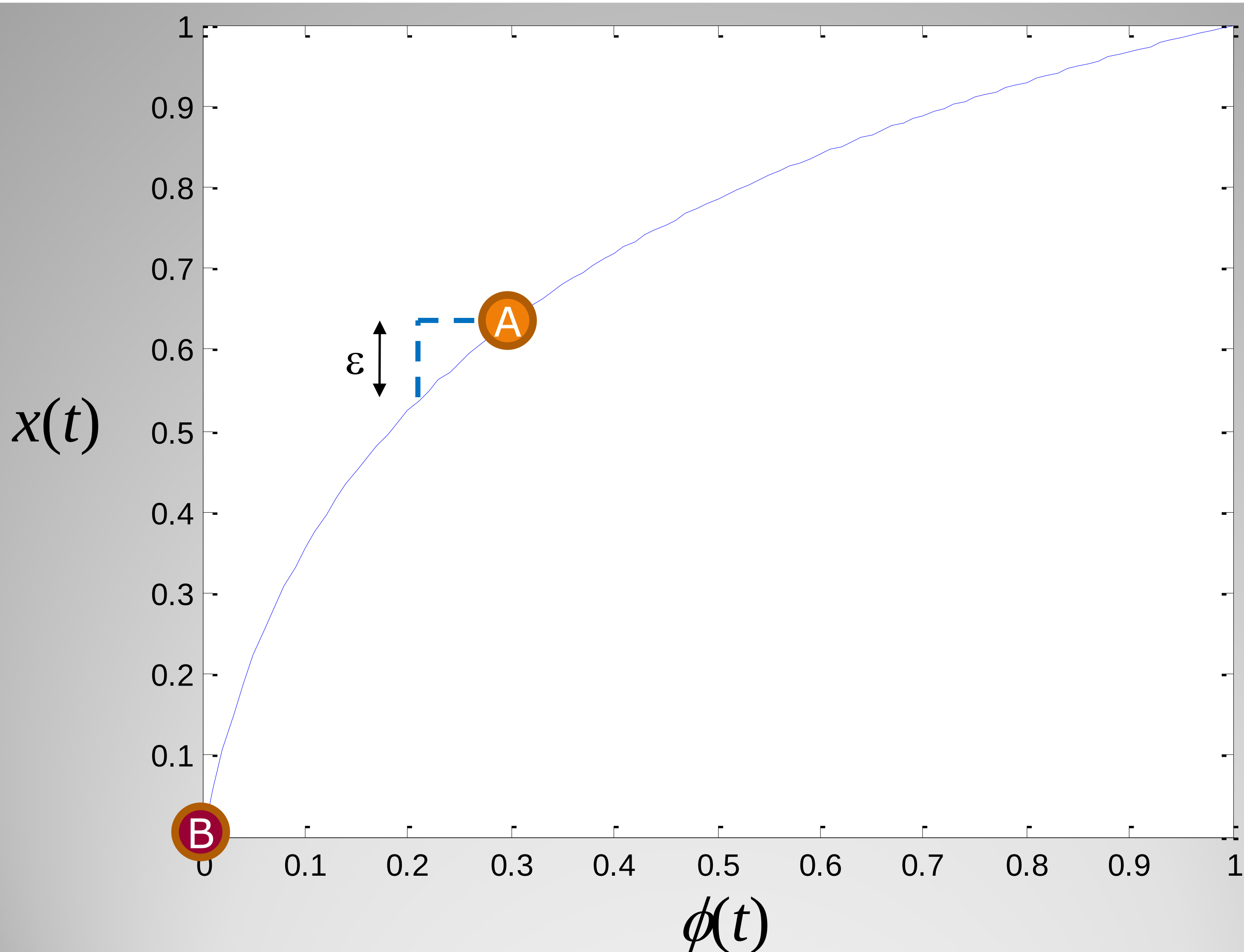
Firefly Spontaneous Synchronization



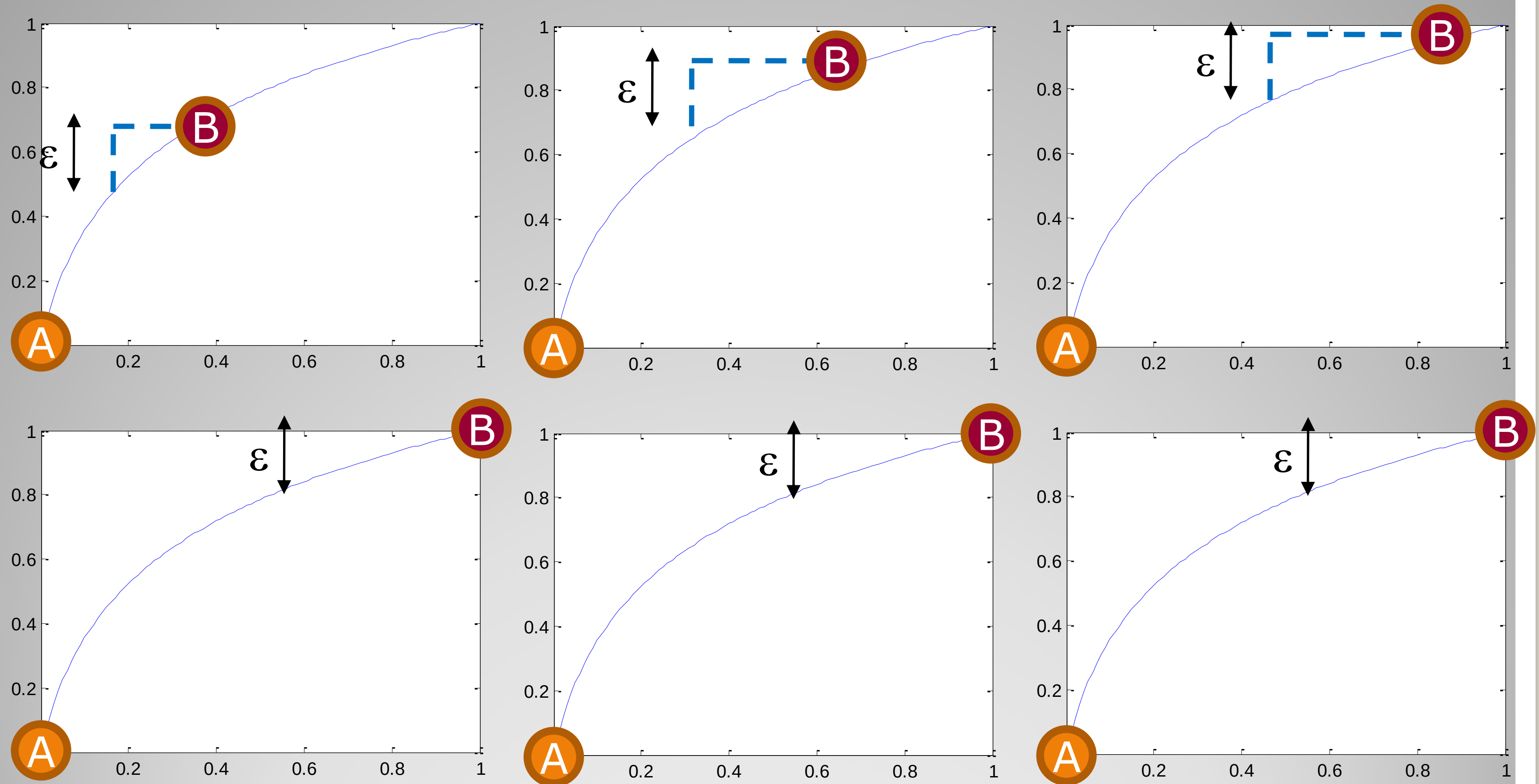
x(t)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ϕ(t)
A
B



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
$x(t)$
$\phi(t)$
ε
A
B


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x(t)
ϕ(t)
ε
A
B

Let us put an stroboscope immediately after A fires



Firefly Spontaneous Synchronization

Firefly Spontaneous Synchronization

- After A fires, A's phase is zero and B's phase is ϕ
- B will fire when its phase advances 1-ϕ. By then, A will be in $x_A = f(1-\phi)$
- Then, B returns to zero and A jumps to

$$x_A = \min(1, \varepsilon + f(1-\phi))$$

- If $x_A = 1$, we achieved synchronism.
- otherwise, $x_A = \varepsilon + f(1-\phi) < 1$ and A's phase is

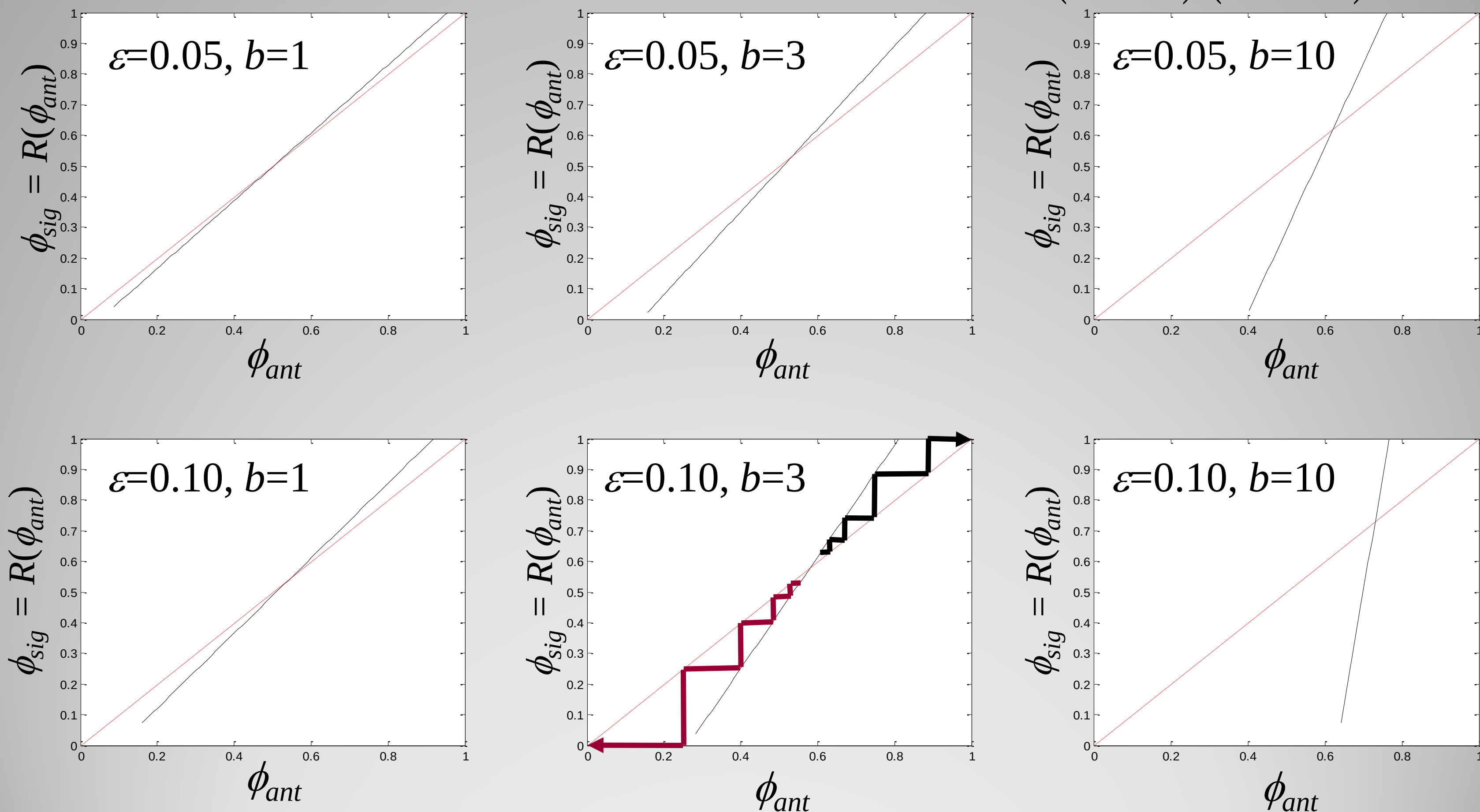
$$h(\phi) = g(\varepsilon + f(1-\phi))$$

- In the next round, if they have not achieves synch, their phases will be $(0, h(h(\phi)))$
- **Return map**: phase of B the next time A fires, if it was at phase ϕ at the previous firing of A

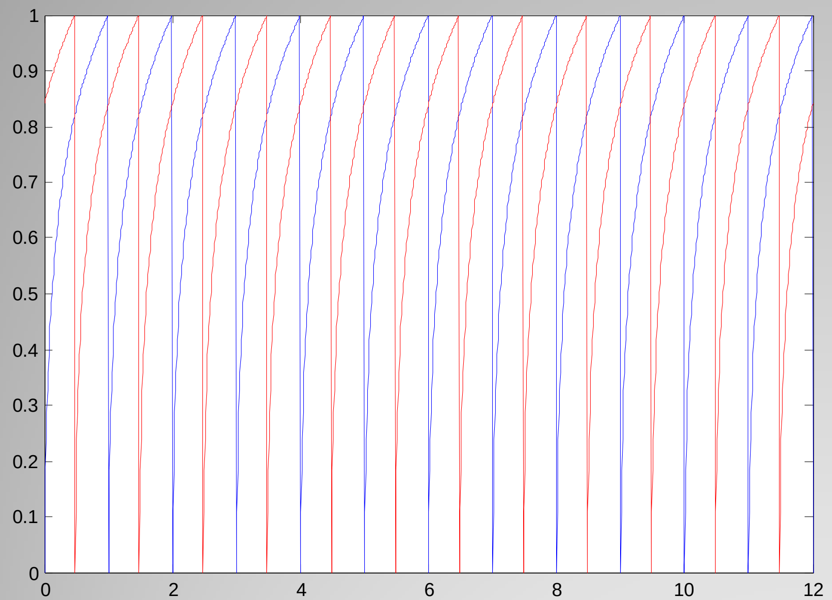
$$R(\phi) = h\big(h(\phi)\big) = g\Big(\varepsilon + f\Big(1 - g\big(\varepsilon + f(1-\phi)\big)\Big)\Big)$$

The return map has a unique fixed point, which is a repeller

$$\phi^* = \frac{e^{b(1+\varepsilon)} - 1}{\left(e^{b} - 1\right)\left(e^{b\varepsilon} - 1\right)}$$

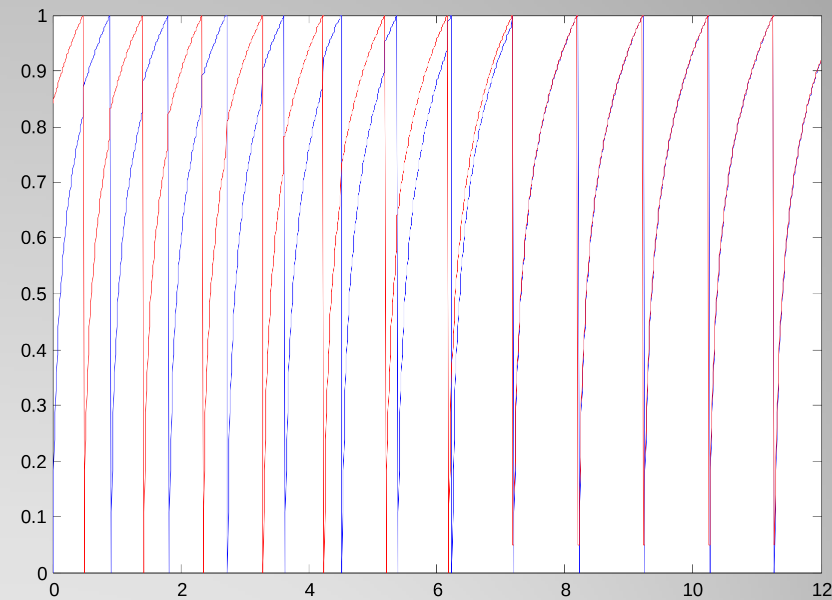


Firefly Spontaneous Synchronization



Independent Oscillators

$b = 3, \varepsilon = 0.0$



Weakly coupled oscillators

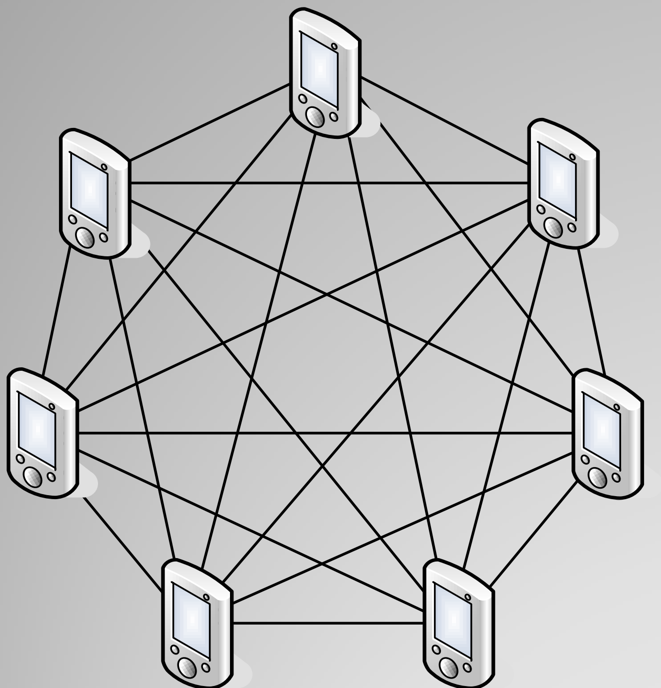
$b = 3, \varepsilon = 0.05$

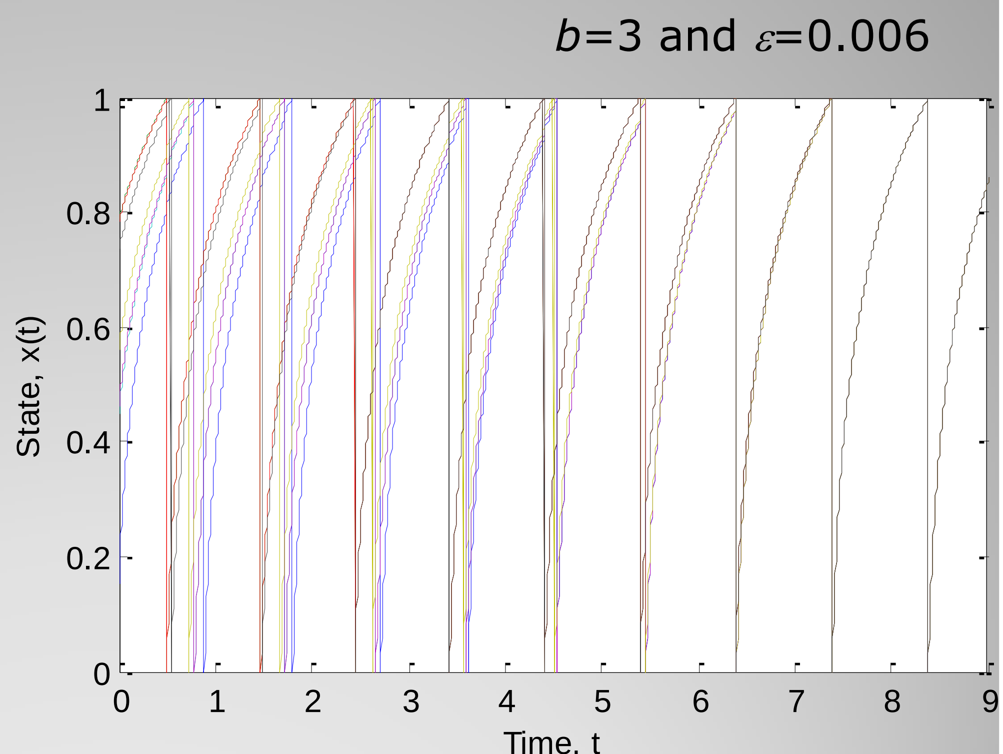
Firefly Spontaneous Synchronization

ideas
Investigación, Desarrollo y Aplicaciones en Señales

- **At physical layer**
  - Frequency Hopping
- **At multiple access layer**
  - Packet scheduling
- **At network layer**
  - Temporary assignment of transmission resources
- **At Transport Layer**
  - Synchronization of transmission patterns for maximum utilization of resources
- **At application layer**
  - Consistent ordering of events in sensor networks

Why synchronization is important in mobile ad hoc networks

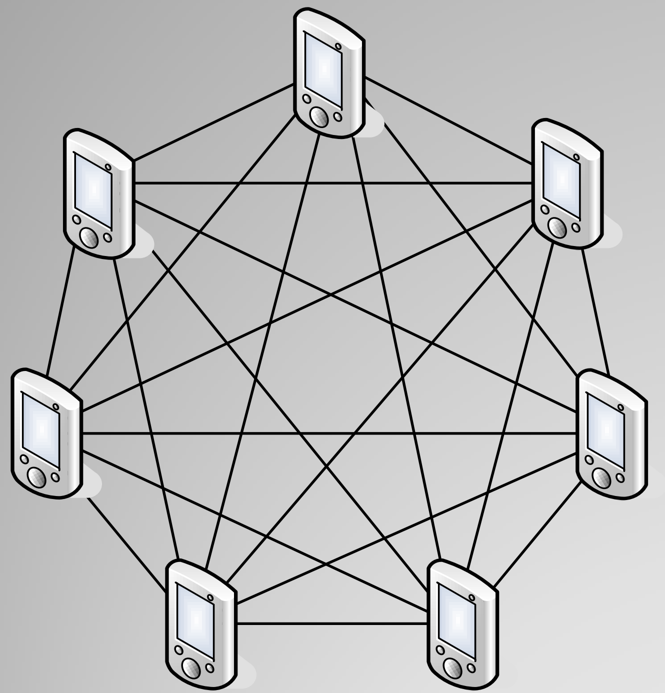


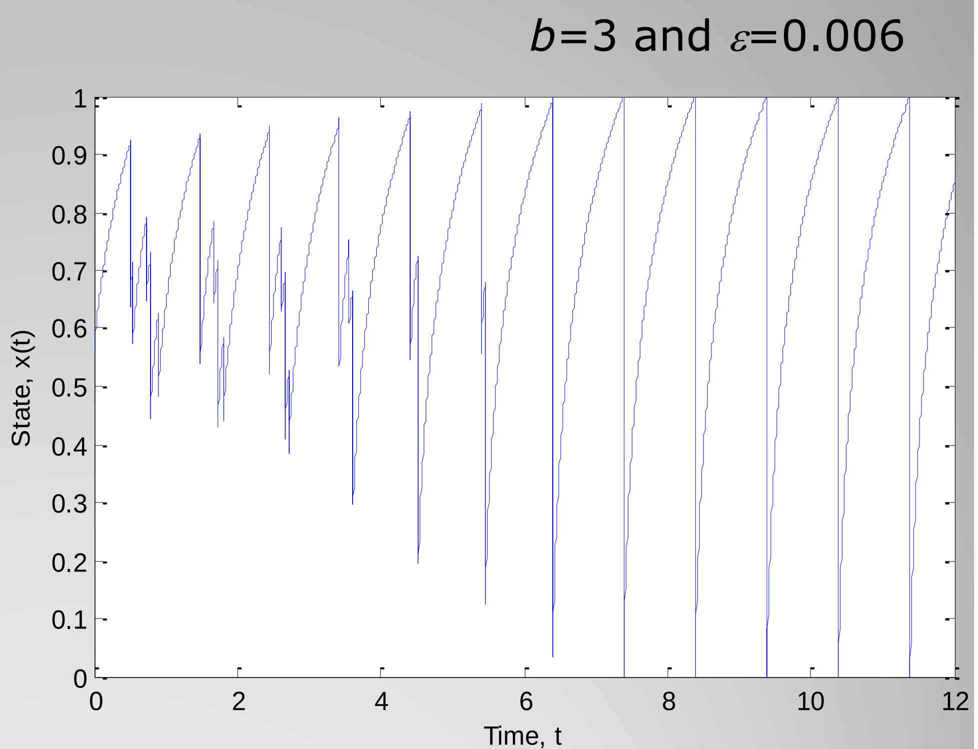


MANET Bio-inspired Synchronization

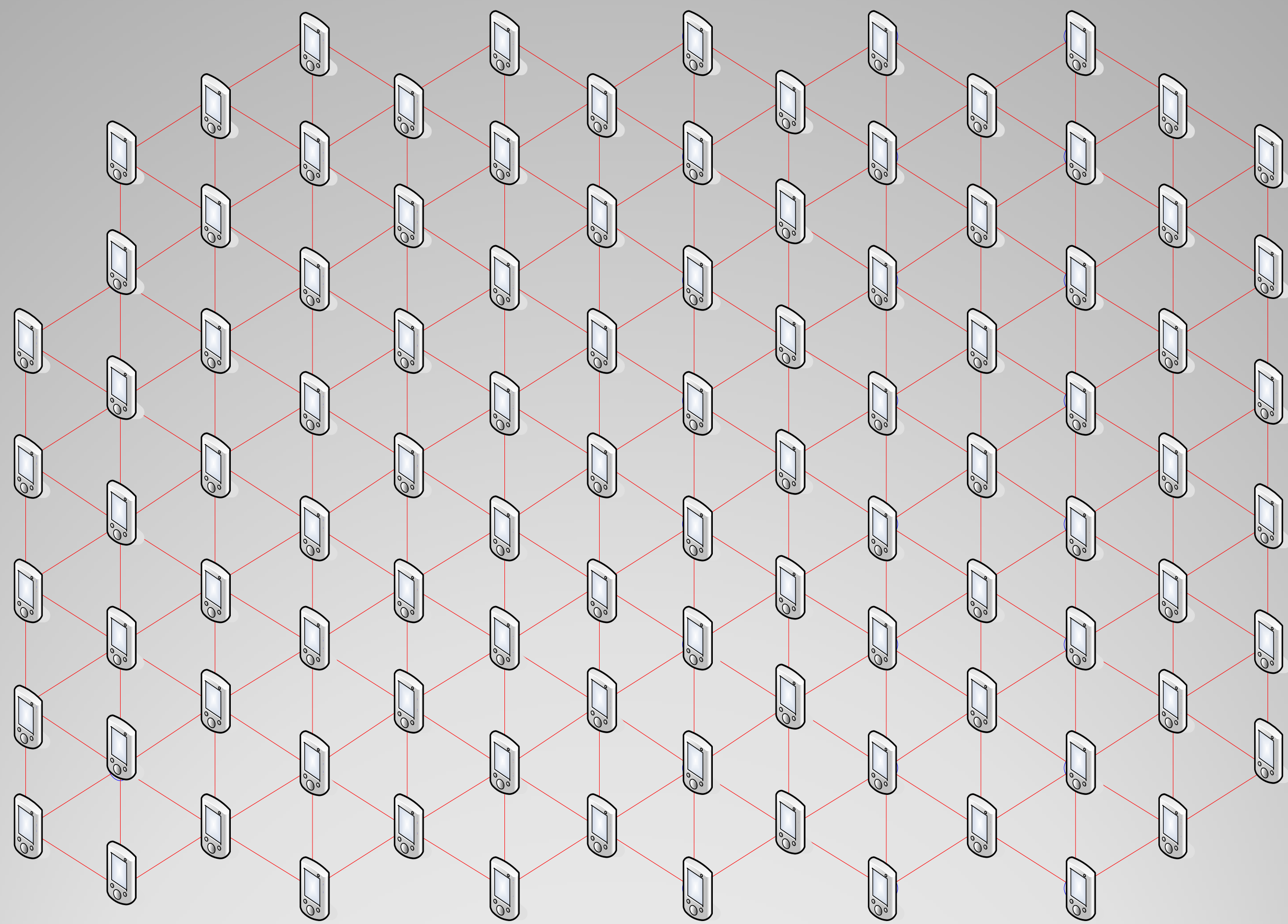
ideas
Investigación, Desarrollo y Aplicaciones en Señales

WEA

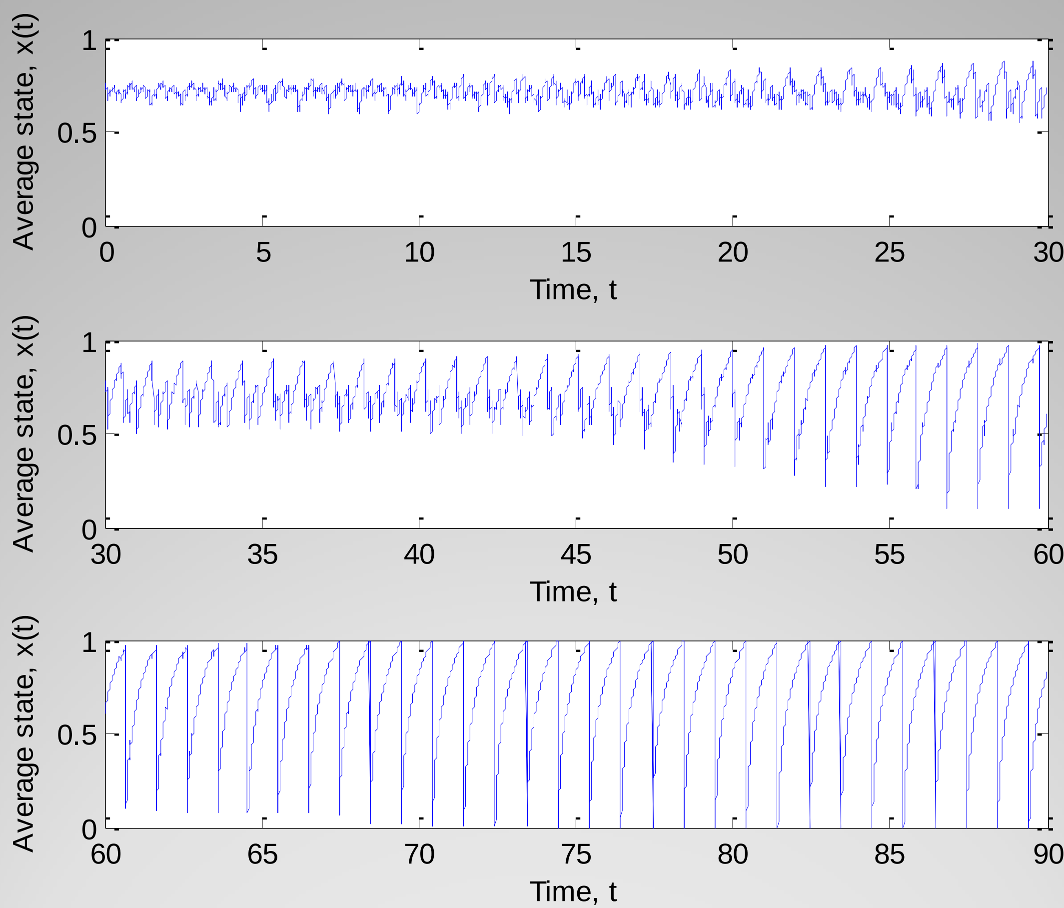




MANET Bio-inspired Synchronization



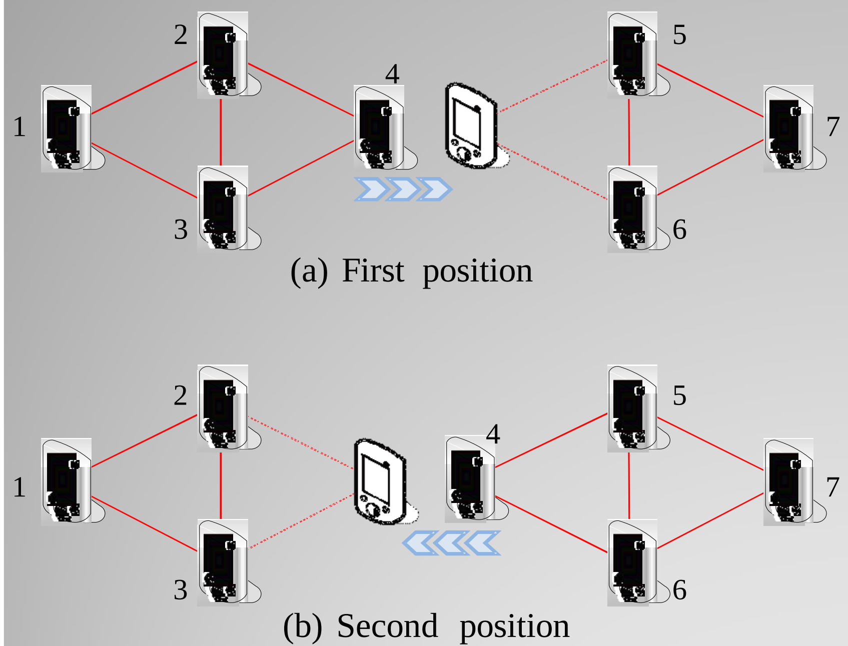
Effects of topology



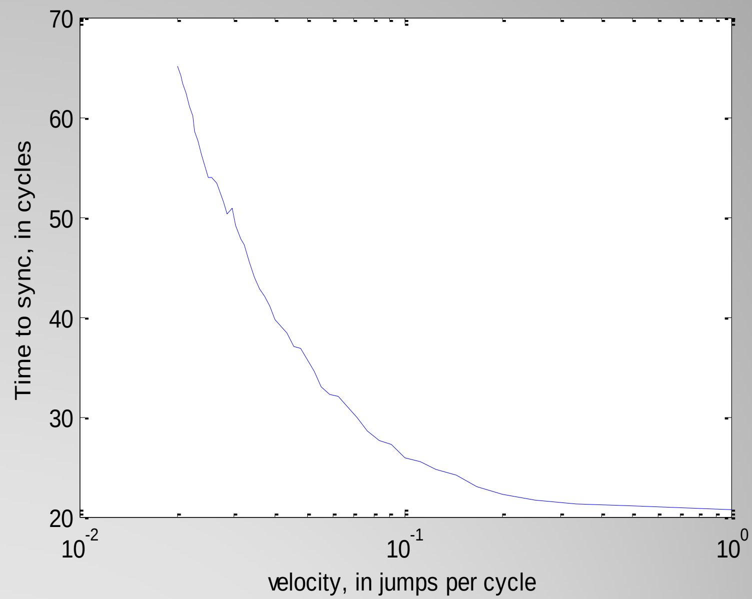
Average state among the 100 nodes

Effects of topology

Effects of Mobility



A Dynamic partitioned network

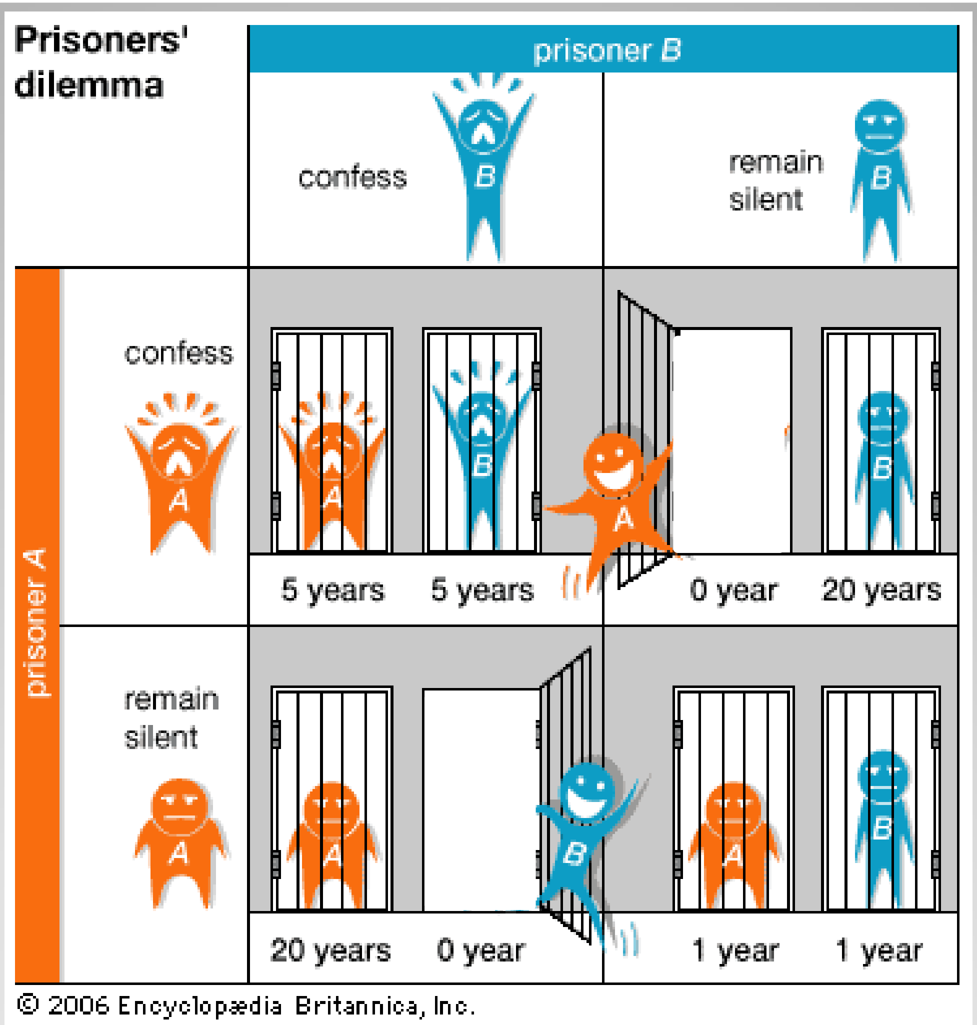


Through mobility, the clusters of a partitioned network get synchronized

- Introduction to Mobile Ad Hoc Networks
- Introduction to Complex Adaptive Systems
- Emergent Synchronization in MANETs
- **Emergent Cooperation in MANETs**
- Formal methods for engineering emergent behavior in MANETs

Schedule

How cooperation evolves?



Prisoner's dilemma

	Confess	Silence
Confess	(C,C)	(T,M)
Silence	(M,T)	(P,P)

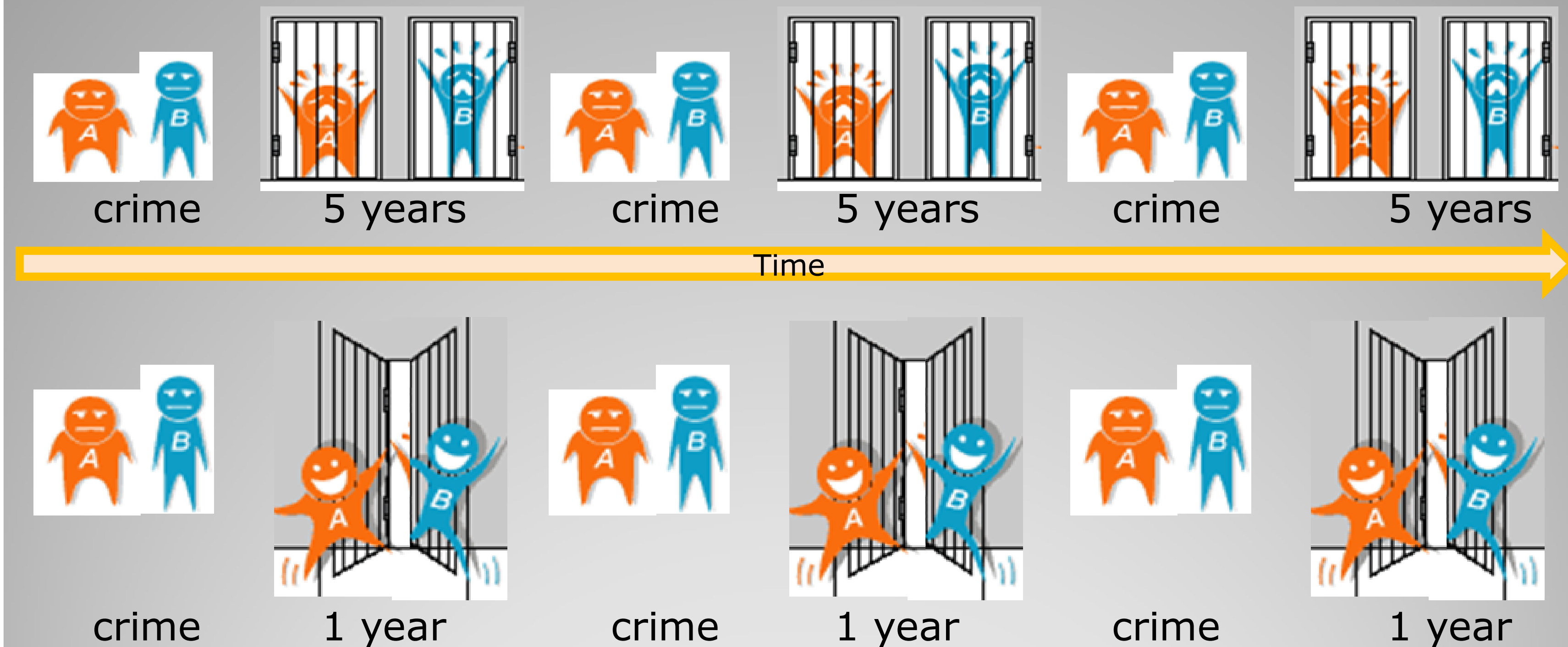
Castigo
Tentación
Marrano
Premio

$$T > P > C > M$$
$$2P > T + M$$

In prisoner's dilemma, to confess (cooperate) dominates over to keep silence (defect), so it is necessary to add an explicit mechanism for cooperation to evolve.

Prisoner's dilemma

Iterated Prisoner's dilemma



Robert Axelrod, "The evolution of cooperation" (1984)

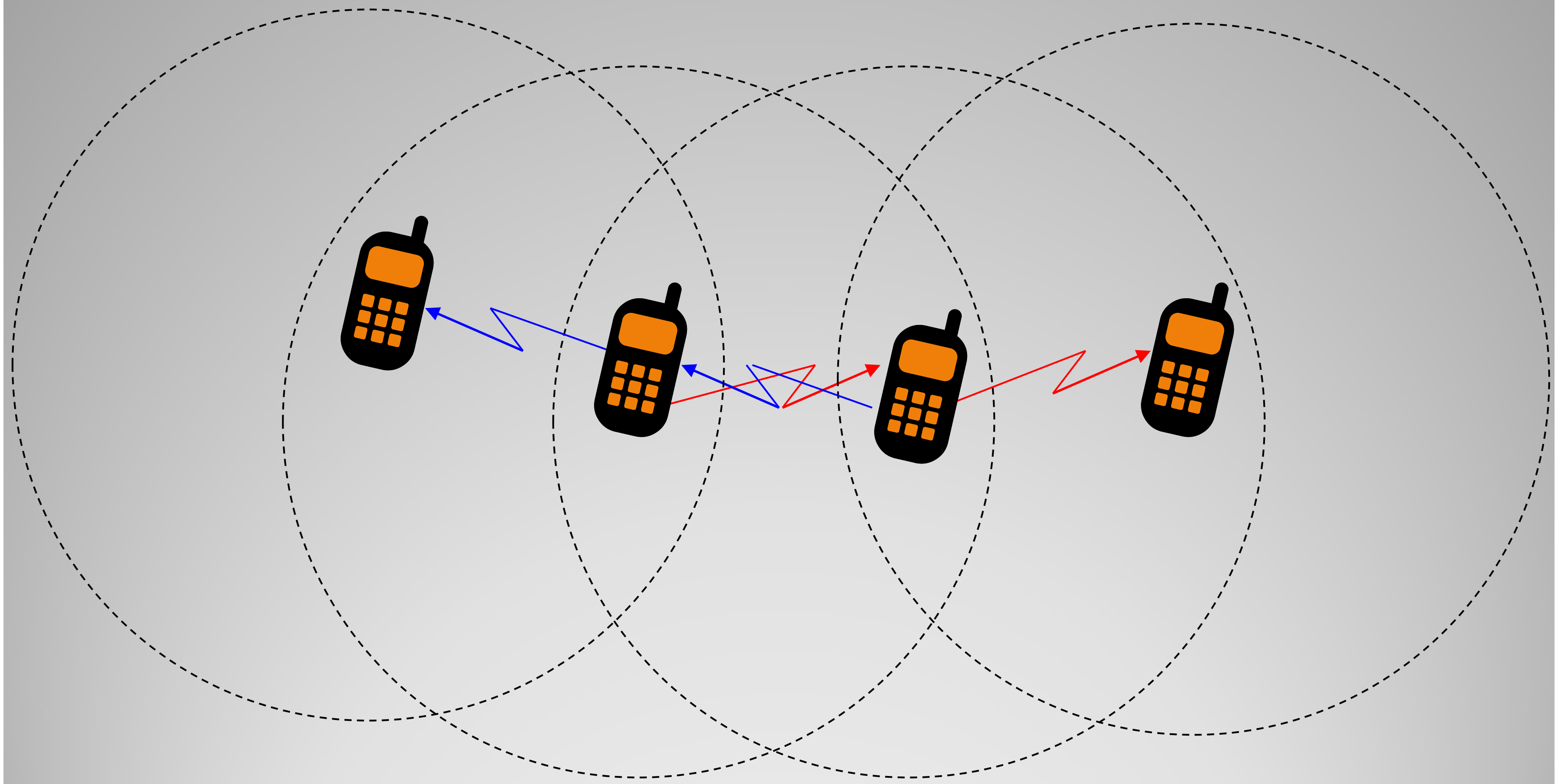
- Reciprocity based cooperation could evolve among selfish egents if a long-term interaction is to be expected.

- My son solution to PD:
  - *"If i commited the crime, I should confess because one must always tell the truth"*

Reputation

Reputation

- Establishes a relationship between behavior in the past and expected behavior in the future
- Involves a behavior that cannot be expected in an isolated interaction.



Mobile Ad Hoc Networks

Each node observes the behavior of its neighbors

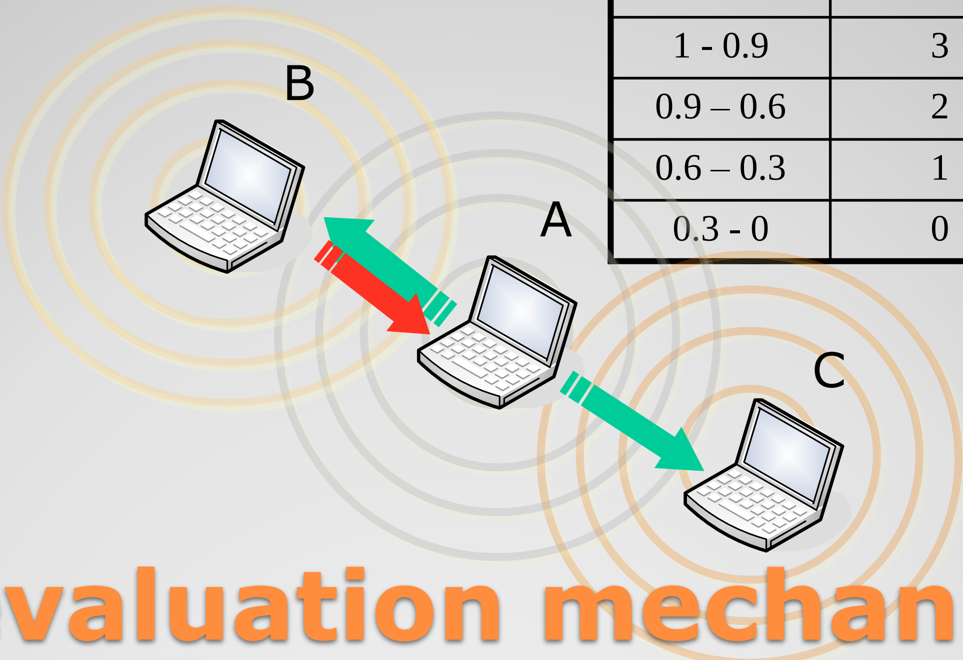
Node *B* has observed that node *A* has received *n* packets to forward and has forwarded n_A of them. Cooperation rate of node A as seen by node B:

$$f_r(B,A) = \frac{n_A}{n}$$

Four possible trust levels, *T*{*B*,*A*}:

Trust table

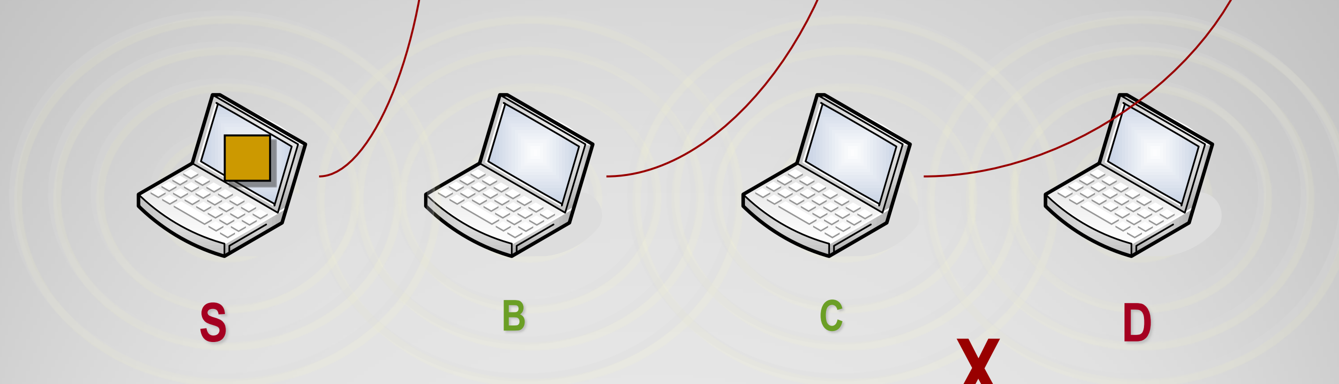
$f_r(B,A)$	$T\{B:A,\cdot\}$
1 - 0.9	3
0.9 – 0.6	2
0.6 – 0.3	1
0.3 - 0	0



Trust evaluation mechanism

Source Node	
State	Pago
Exit	5
Fail	0

Intermediate Node				
	T=3	T=2	T=1	T=0
forward	3	2	1	0.5
Discard	0.5	1	2	3



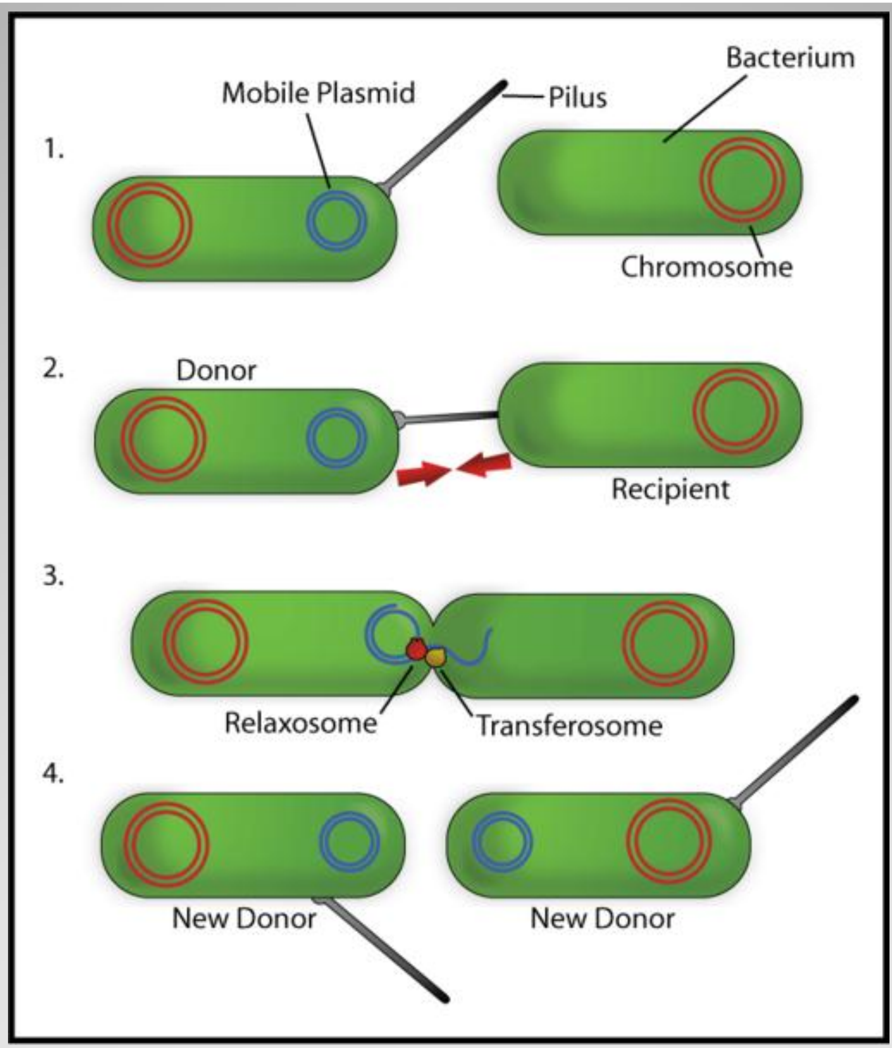
$T\{B,S\}=3$ $T\{C,S\}=0$

Payment structure

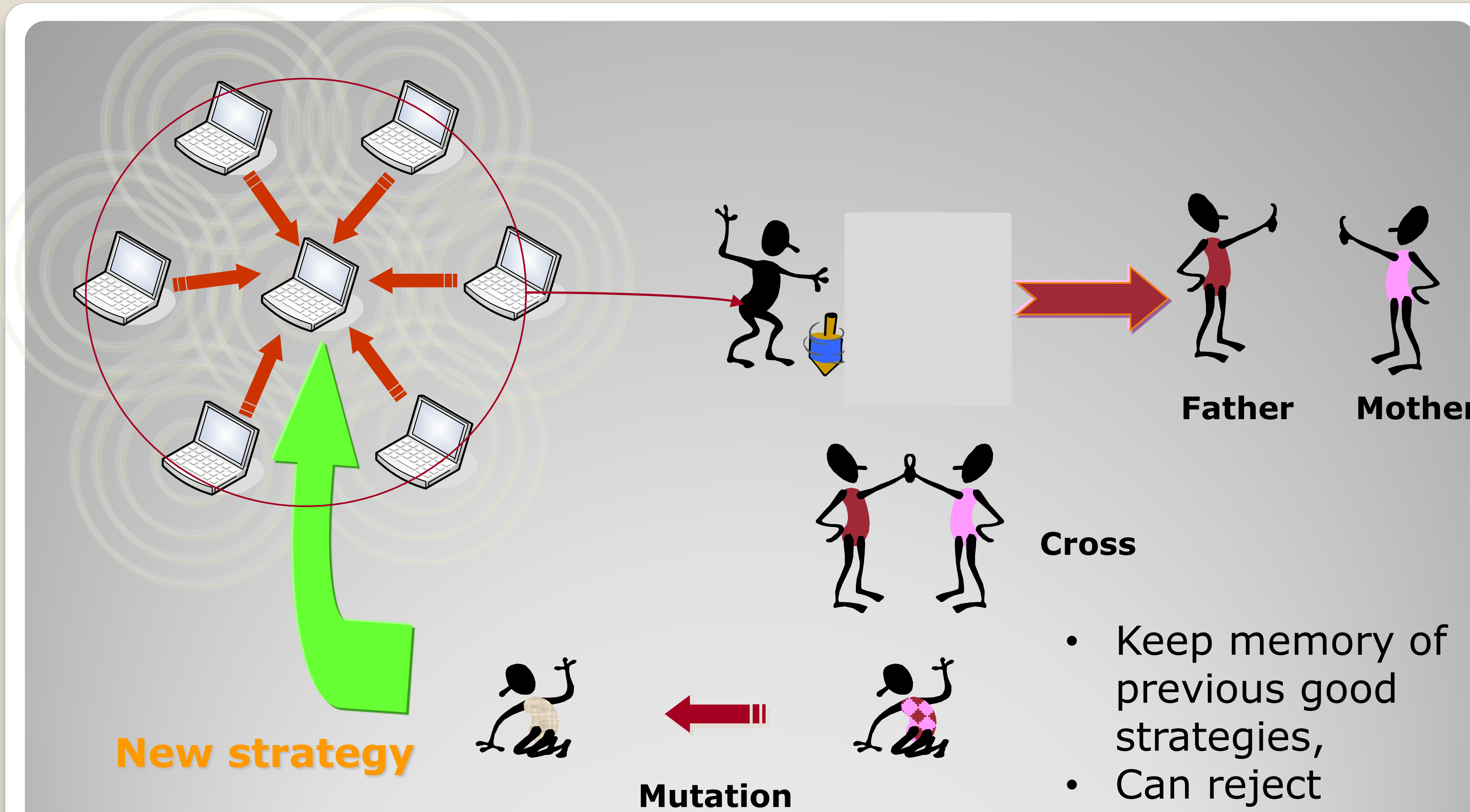
Trust on source node	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
Transmission of previous packet	F	F	E	E	F	F	E	E	F	F	E	E	F	F	E	E
Transmission of second to last packet	F	E	F	E	F	E	F	E	F	E	F	E	F	E	F	E
Decision	D	D	D	C	D	D	C	C	D	C	C	C	D	C	C	C

C - *cooperate,* D - *discard* E - success F - fail

Strategy coding

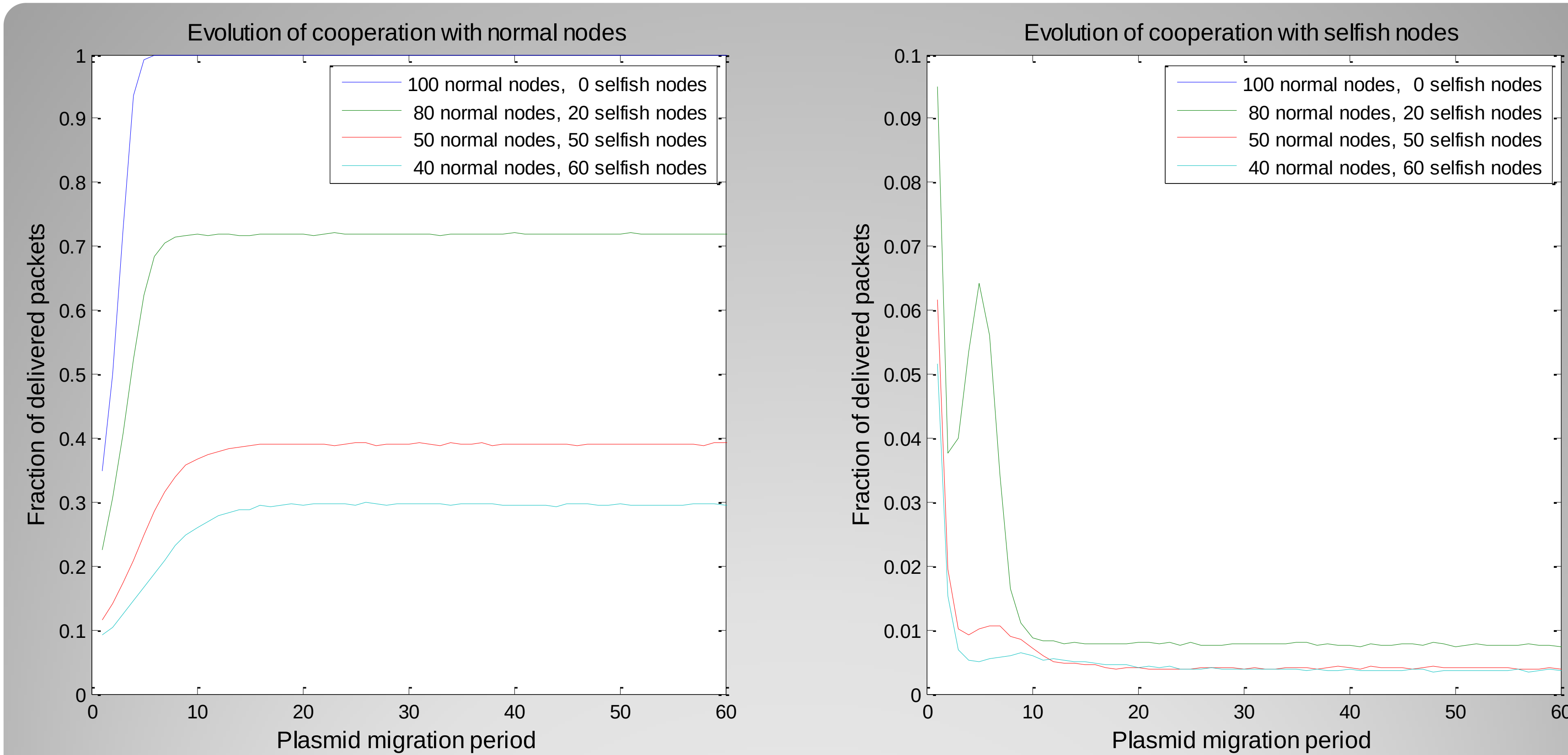


Plasmid migration

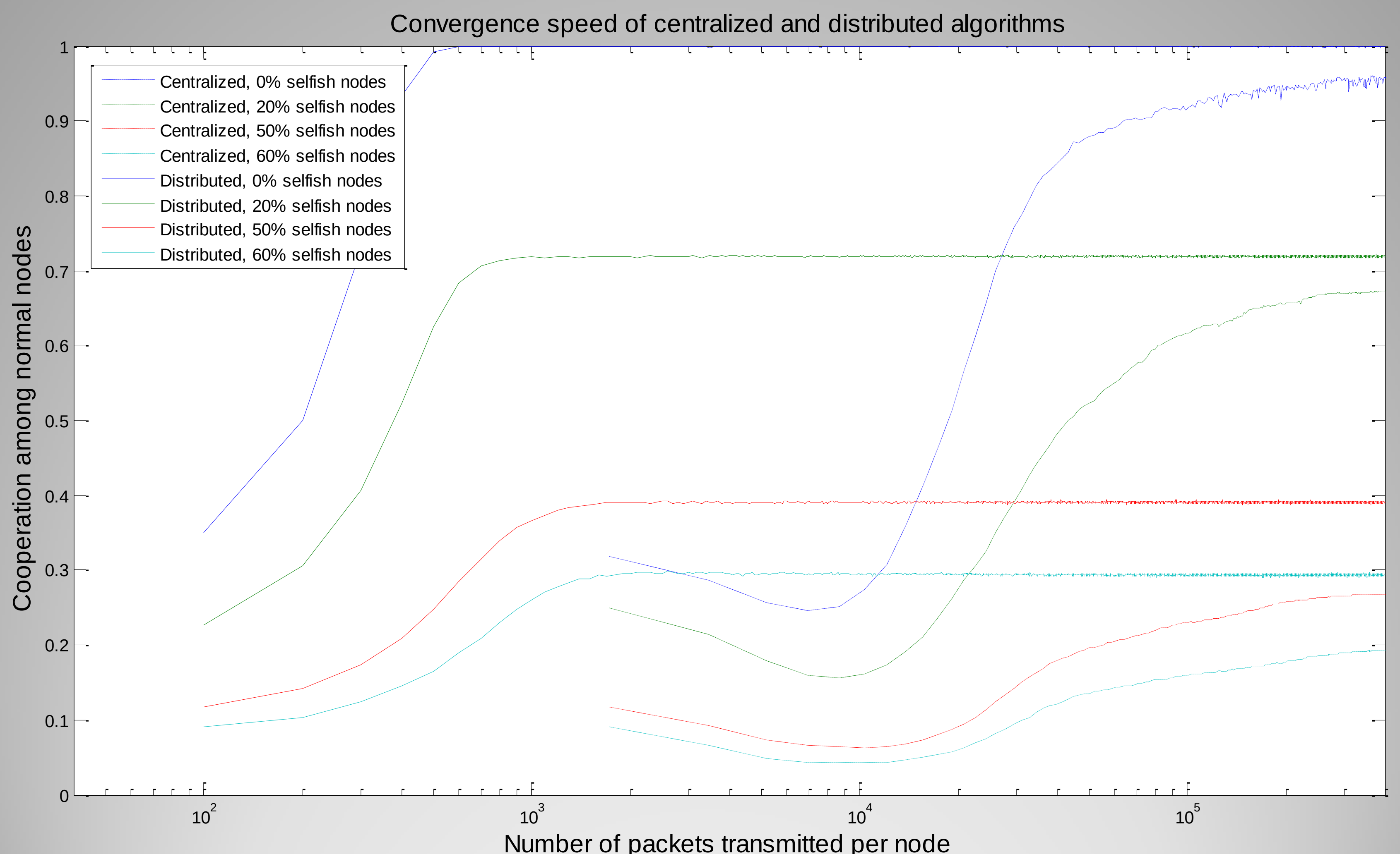


- Keep memory of previous good strategies,
- Can reject parents genes,
- Forget remote past

Plasmid migration



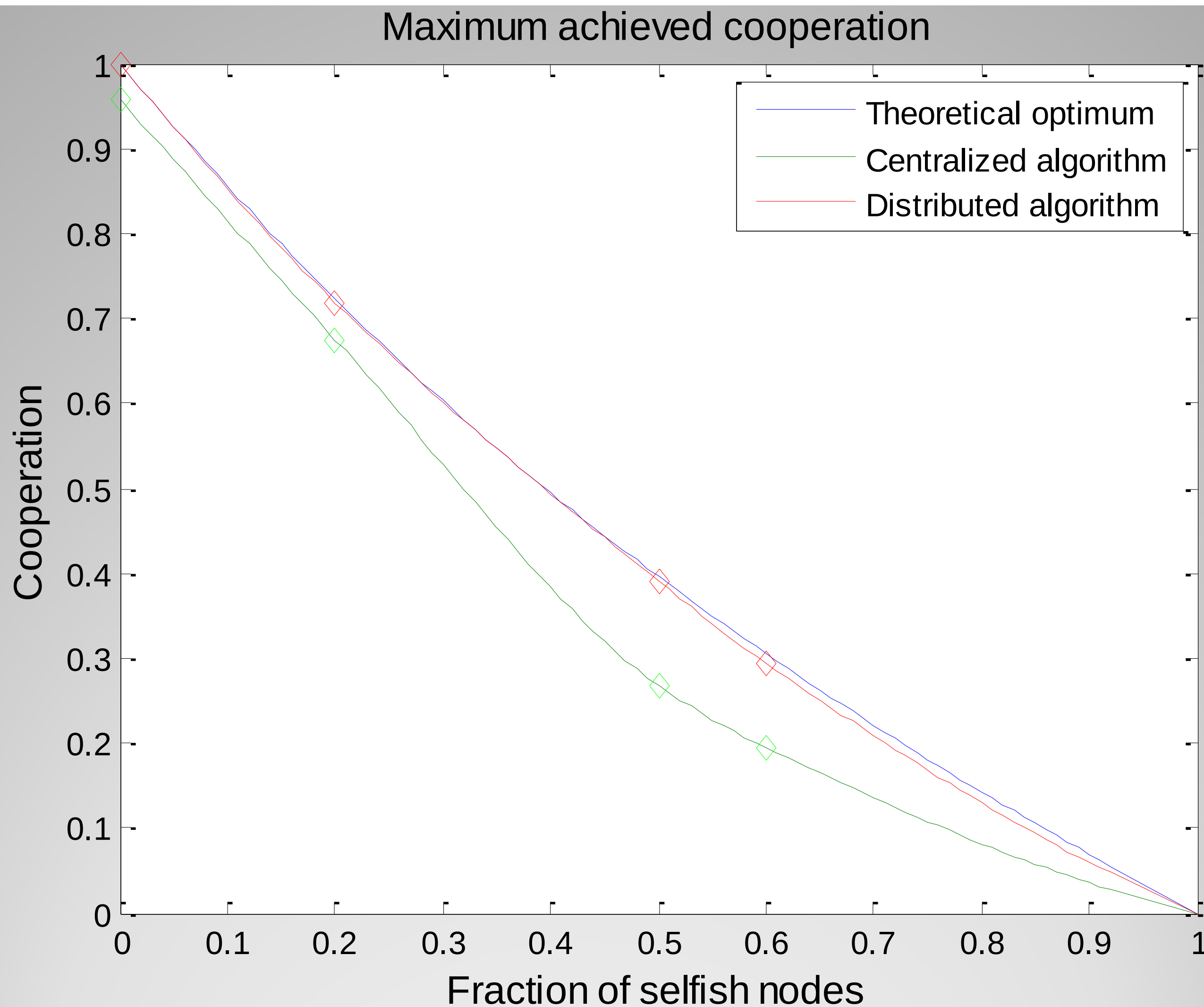
Evolution in four different environments

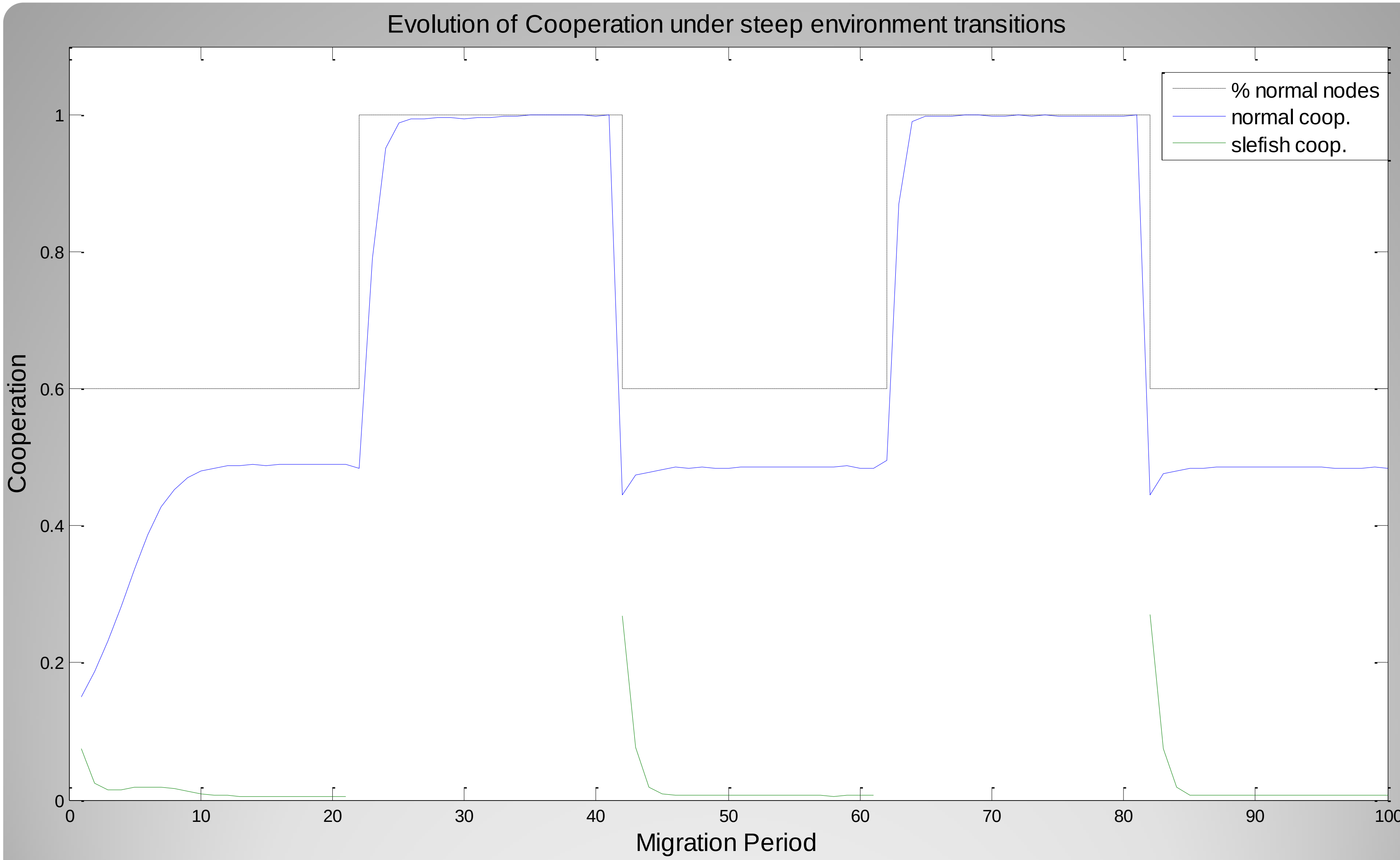


Convergence

ideas
Investigación, Desarrollo y Aplicaciones en Señales

Optimality of solutions





Adaptability

ideas
Investigación, Desarrollo y Aplicaciones en Señales

Schedule

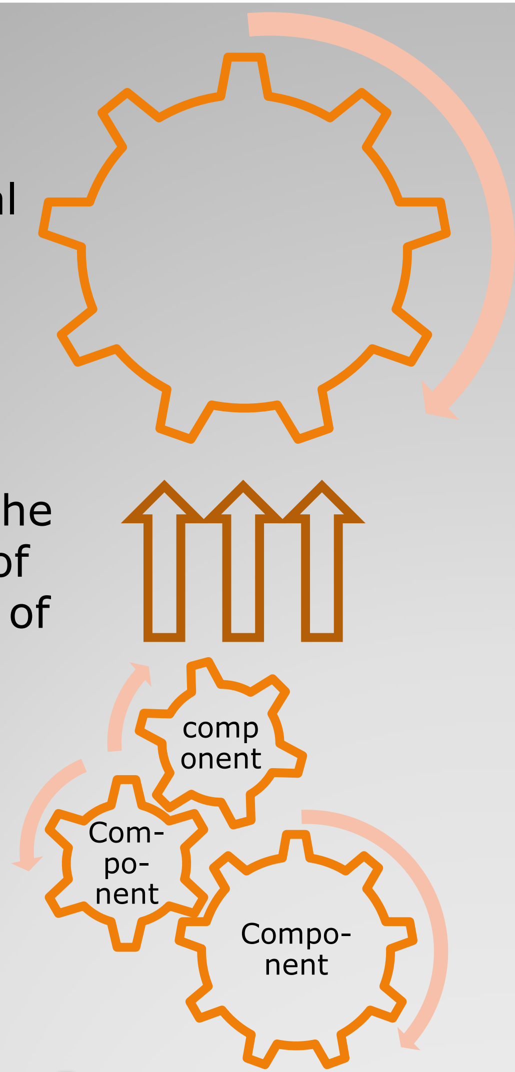
- Introduction to Mobile Ad Hoc Networks
- Introduction to Complex Adaptive Systems
- Emergent Synchronization in MANETs
- Emergent Cooperation in MANETs
- **Formal methods for engineering emergent behavior in MANETs**

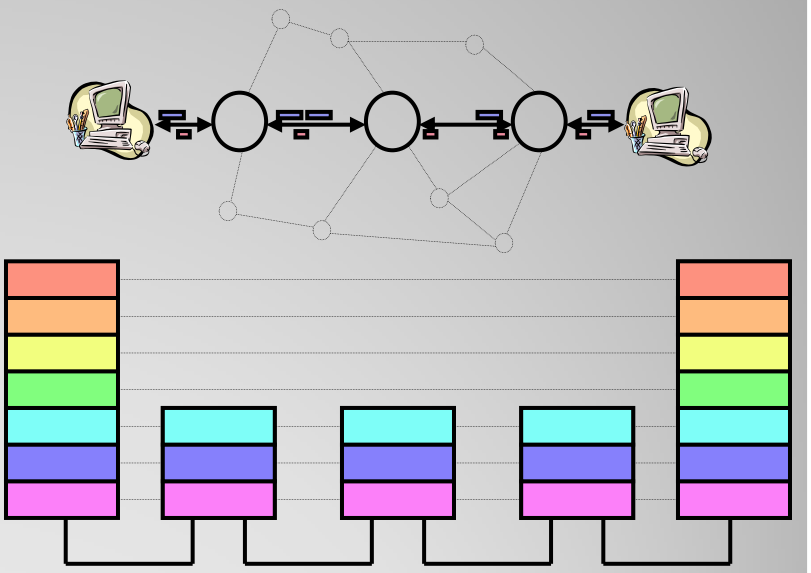
Classical design

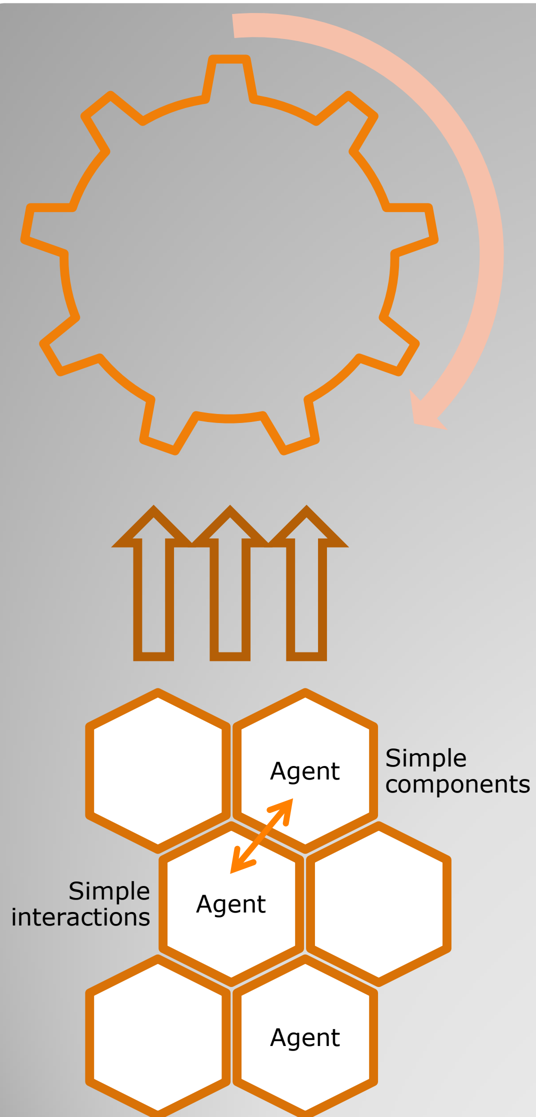
Desired global behavior

The whole functionality comes from the composition of the functions of each component

The structure of the system reflects its functionality





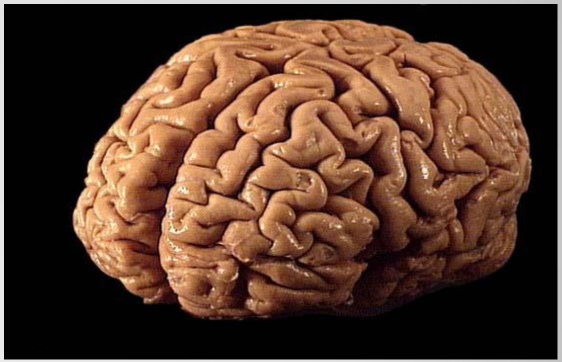


Macroscopic
global behavior

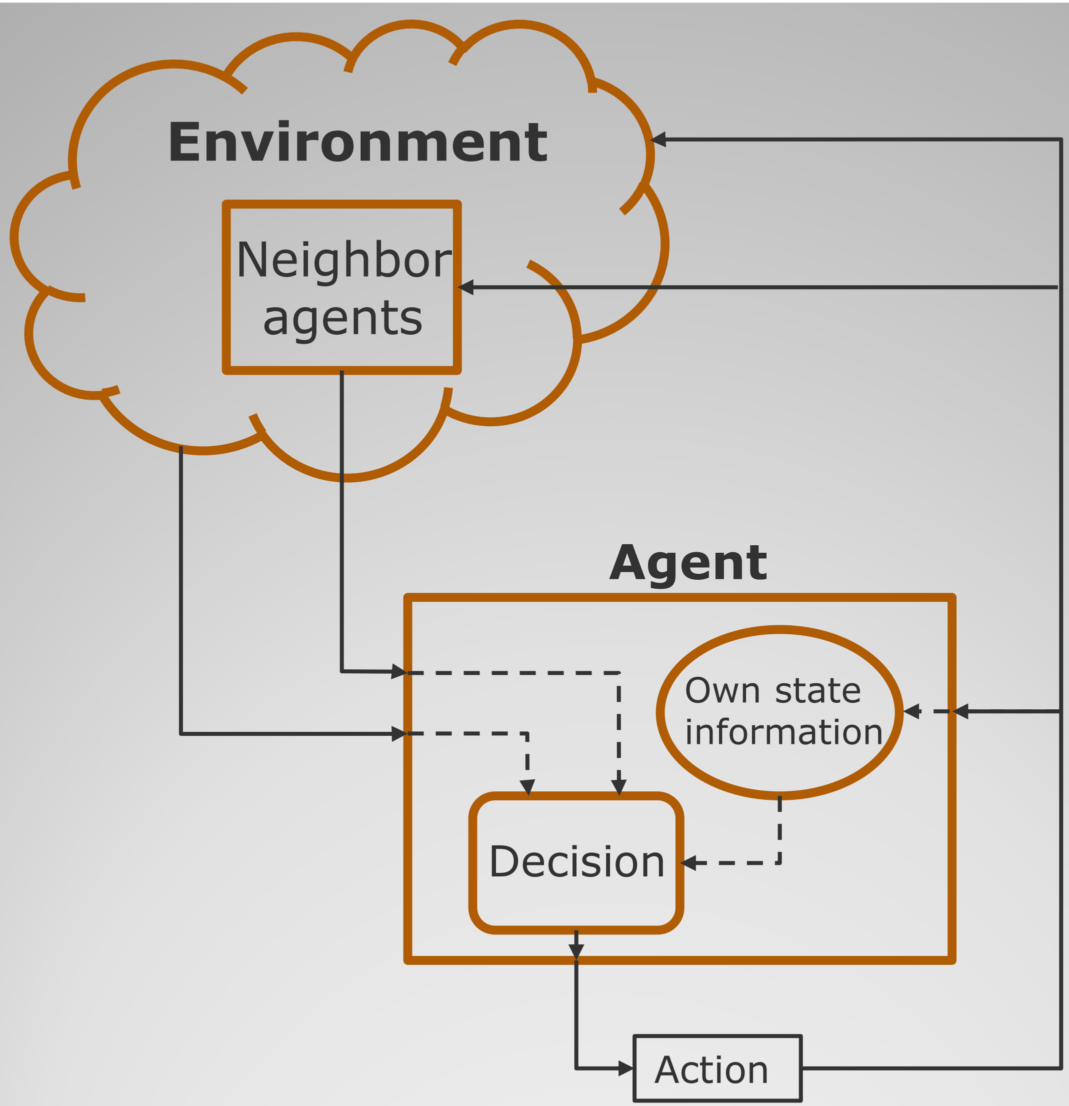
Emergent
self-organization

Microscopic
local behavior

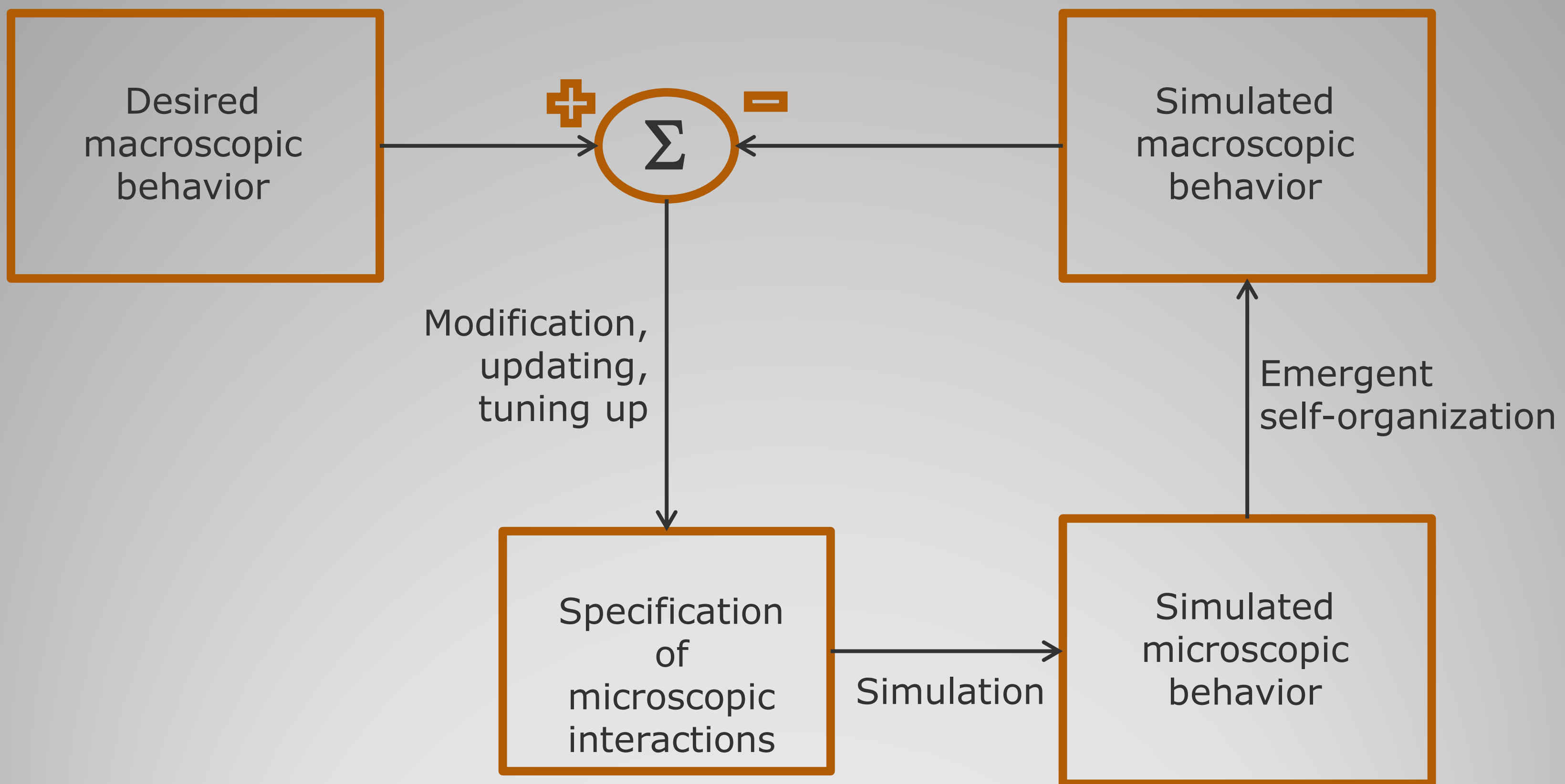
http://www.tech-faq.com/swarm-intelligence.html



CAS design


Environment
Neighbor agents
Agent
Own state information
Decision
Action

CAS design

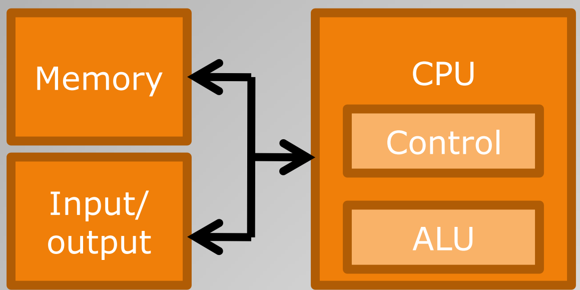


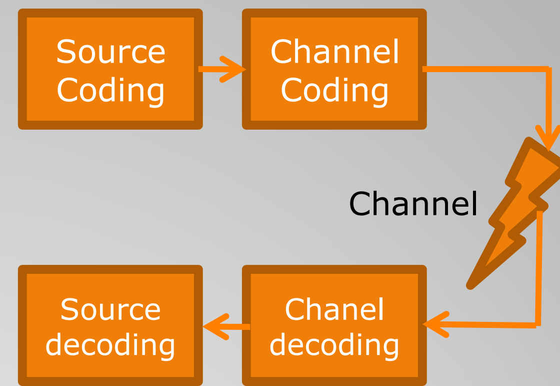
Desired macroscopic behavior
+
Σ
−
Simulated macroscopic behavior
Modification, updating, tuning up
Emergent self-organization
Specification of microscopic interactions
Simulation
Simulated microscopic behavior

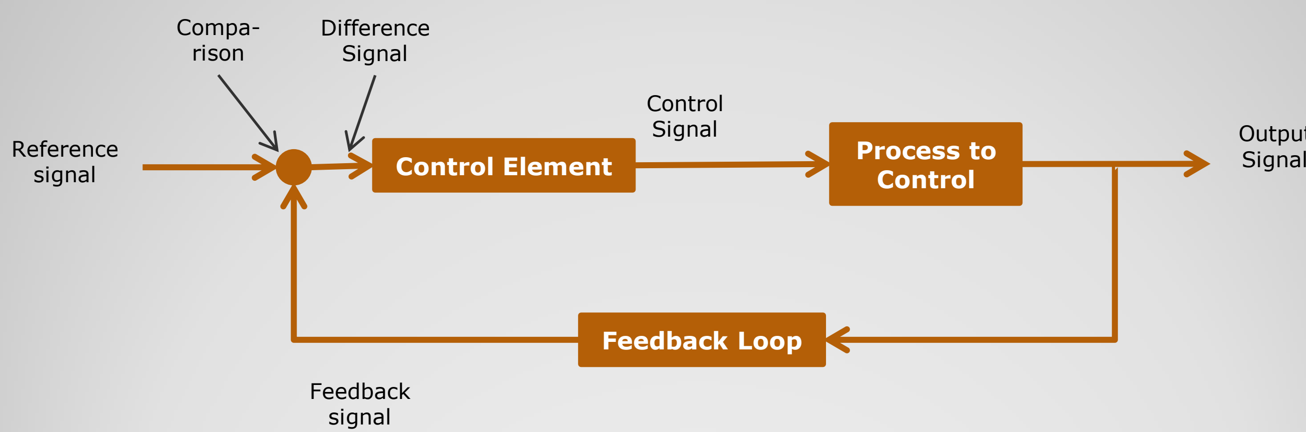
CAS design

Conclusions

- Long history of success of classical engineering







- But new problems out of scope of classical engineering
  - Non-linearity, Fractals, chaos, turbulence, power laws, Self-organization, Emergence, Learning, Evolution, Adaptability
- MANET – CAS
  - Could it be possible to forget about functional structure design and move towards emergent self-organization?
- Many research to do, looking for formalization
  - information theory, non-linear dynamical systems, optimal control, computational intelligence, etc.

Conclusions

Questions?

This presentation is in http://www.udistrital.edu.co/wpmu/malzate/