Distributed policy scheduling in sensor networks

INFOCOM 2007

Yu Chen and Eric Fleury
ARES/INRIA- INSA de Lyon, France

Yet another Applicati

- MOSAR is an Integrated Project supported for 5 years by the European Commission under the Life Science Health Priority of the Sixth Framework Program.
- Coordinated by INSERM (the French National Institute of Health and Medical research);
- MOSAR aims to significantly advance our knowledge regarding the control of antimicrobial resistance of bacteria responsible for major and emerging nosocomial infections.
- http://perso.ens-lyon.fr/eric.fleury/Upload/Mosar/MosarEng080120.wmv

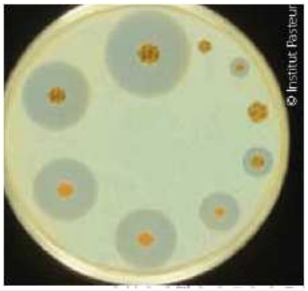
MOSAR project outline

- Better understand the dynamic of AMRB transmission
  - the real-time analysis of the relative contribution of exposure to antibiotics;
  - the intrinsic characteristics of epidemic clones that contribute to inter-individual transmission;
  - the identification of factors contributing to the transmission of strains between individuals in the hospital population and community;

- Model and Prediction of AMRB
  - **Dynamic**: track variations in the number of infected or colonized patients;
  - **Mechanistic**: encapsulate our understanding of mechanisms and processes involved in the bacteria transmission.

AMBR transmission & dynamic

- Investigate the relative contribution in the promotion of:
  - AMR pathogens in hospital settings;
  - of antibiotic selective pressure
  - epidemicity of strains and potential cross-transmission due to inter-individual contacts.
- DATA collected:
  - Antibiotic selective pressure
  - Clonal-specific epidemicity
    - the effectiveness of control measures
    - the duration of colonization
    - the number of susceptible persons


© Institut Pasteur

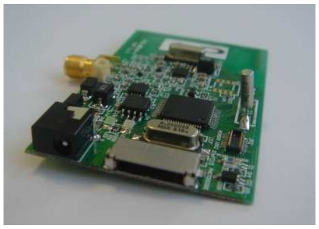
A major conjoint challenge for TIC & LSH

- A data collection strategy will combine for a period of 6 months on 400 actors:
  - an individual antibiotic use;
  - a contact monitoring;
  - a characterization of the isolates to determine their epidemicity;

© Michael Grunspck - FOCUS

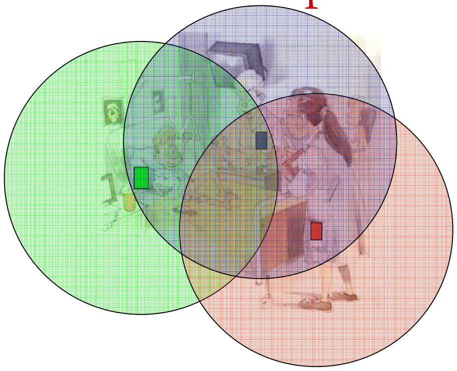
© Institut Pasteur

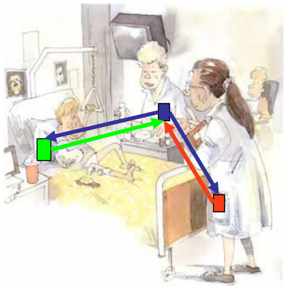
Deployment of a large-scale ambient dynamic networks

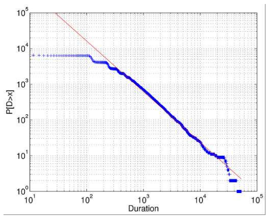


- Document interactions between
  - medical and nursing staff
  - patients to patients
  - patient to medical staff
- Document contact frequencies
  - monitor the dynamic (inter & intra contact)
  - characterize the interaction network
- One sensor ⇔ One person

MOSAR’s experiment

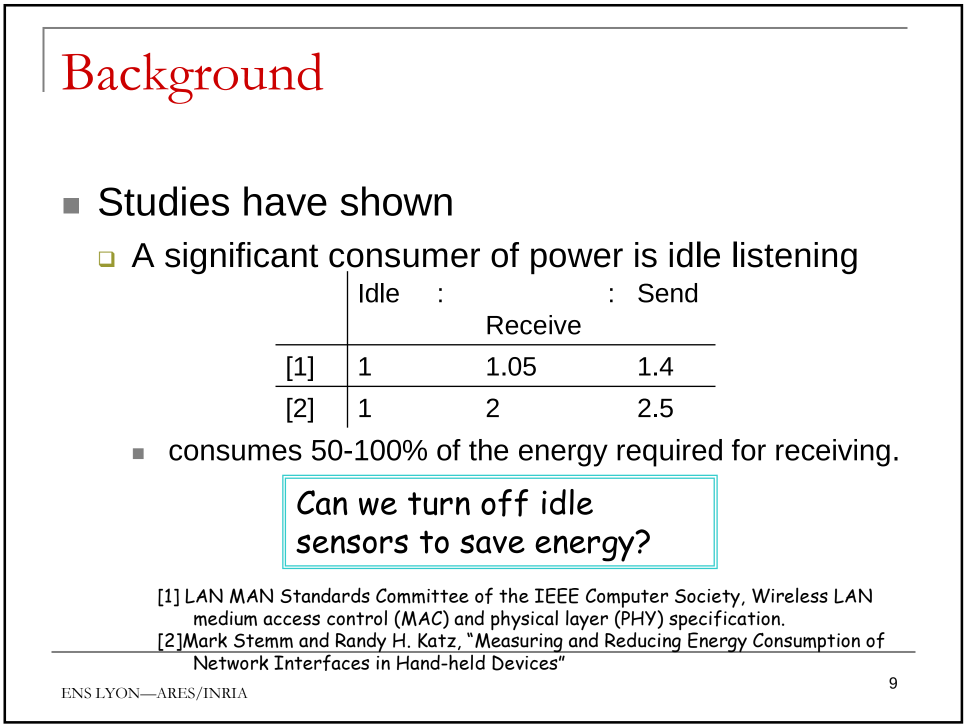






Background

- Wireless Sensor Networks
  - Battery-operated sensor nodes
    - Energy Efficiency
  - How to save energy in WSN?



Background

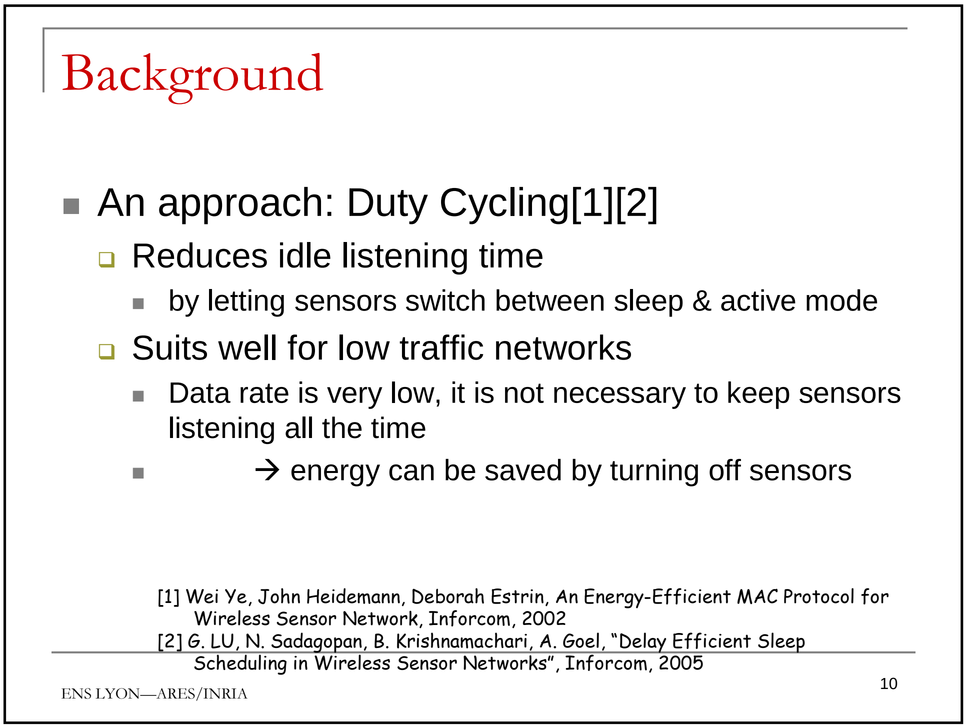
- Studies have shown
  - A significant consumer of power is idle listening

	Idle :	Receive :	Send
[1]	1	1.05	1.4
[2]	1	2	2.5

- consumes 50-100% of the energy required for receiving.

Can we turn off idle sensors to save energy?

[1] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification.
[2]Mark Stemm and Randy H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices"



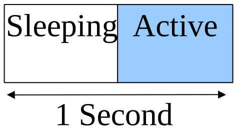
Background

- An approach: Duty Cycling[1][2]
  - Reduces idle listening time
    - by letting sensors switch between sleep & active mode
  - Suits well for low traffic networks
    - Data rate is very low, it is not necessary to keep sensors listening all the time
    - → energy can be saved by turning off sensors

[1] Wei Ye, John Heidemann, Deborah Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Network, Inforcom, 2002
[2] G. LU, N. Sadagopan, B. Krishnamachari, A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks", Inforcom, 2005

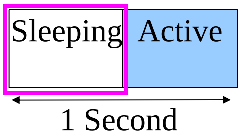
Background

- A Solution: Duty Cycling
  - For example,
    - Duty cycling: in each second, each sensor is scheduled to sleep for half a second and to listen for the other half.
    - Its duty cycle is reduced to 50%.



Background

- A Solution: Duty Cycling
  - For example,
    - If sensors are idle listening most of the time
      - energy consumed by idle listening in the 50% of the time can be saved.

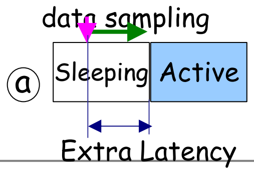


Communication and Power

- Sleep is good…

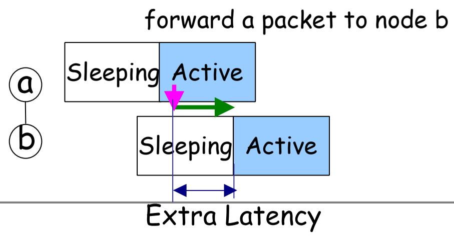
Background

- Problems caused by duty circling
  - Extra message latency
    - Case 1: data sampled by a node during its sleep period have to be queued until the active period



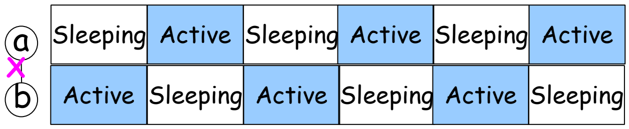
Background

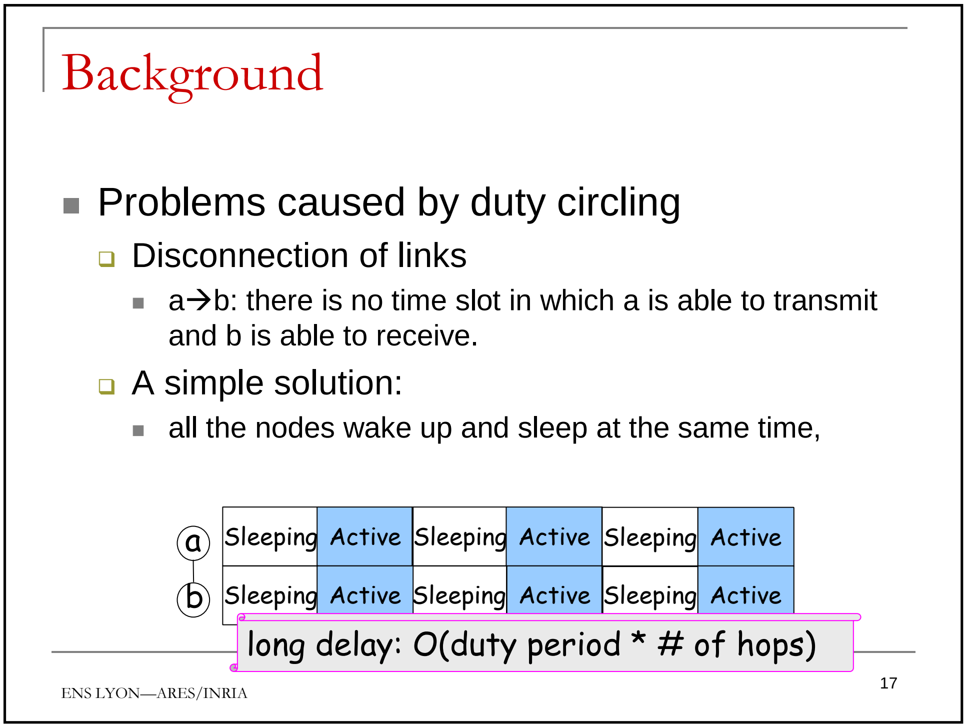
- Problems caused by duty circling
  - Extra message latency
    - Case 2: when a node receives a packet, it has to wait until the next hop wakes up to forward the packet.



Background

- Problems caused by duty circling
  - Disconnection of links
    - a→b: there is no time slot in which a is able to transmit and b is able to receive.
  - Disconnection of links might cause network partitions.



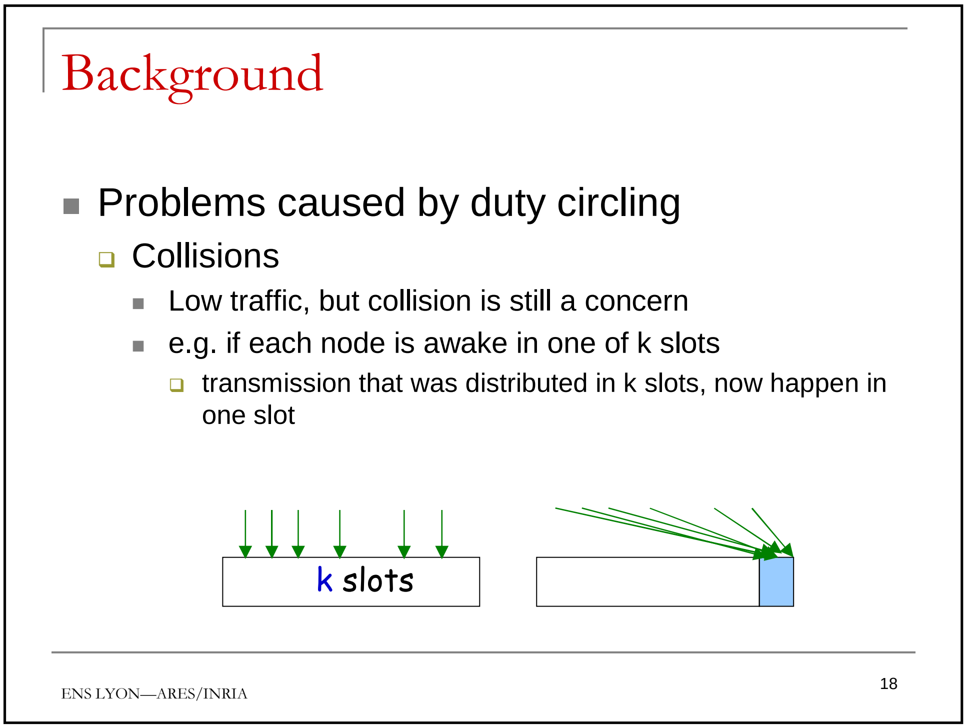


Background

- Problems caused by duty circling
  - Disconnection of links
    - a→b: there is no time slot in which a is able to transmit and b is able to receive.
  - A simple solution:
    - all the nodes wake up and sleep at the same time,

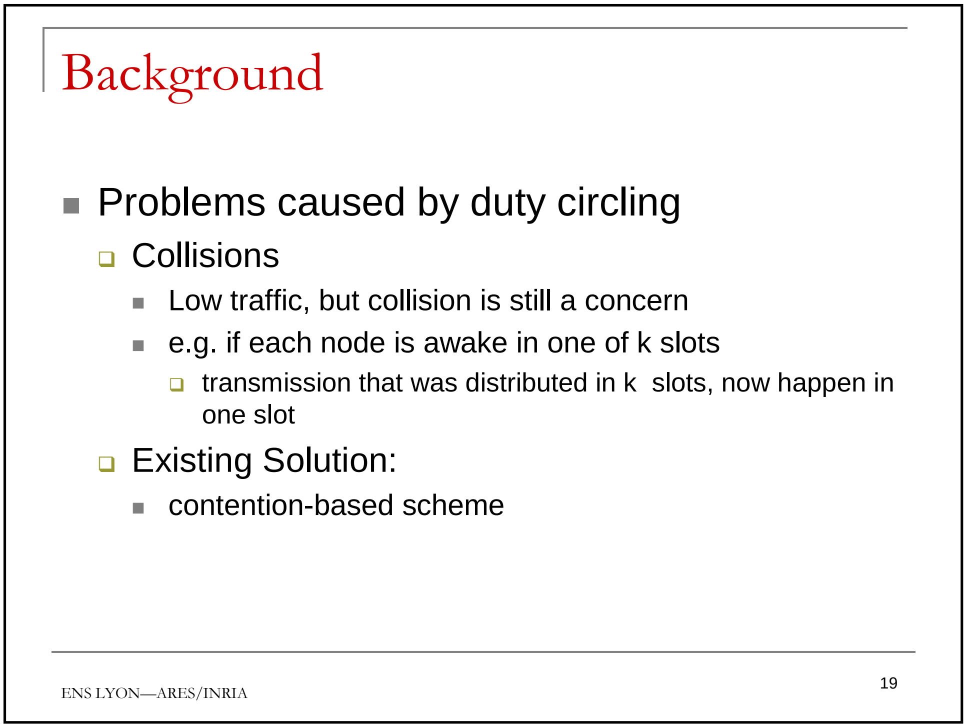
a	Sleeping	Active	Sleeping	Active	Sleeping	Active
b	Sleeping	Active	Sleeping	Active	Sleeping	Active

long delay: O(duty period * # of hops)



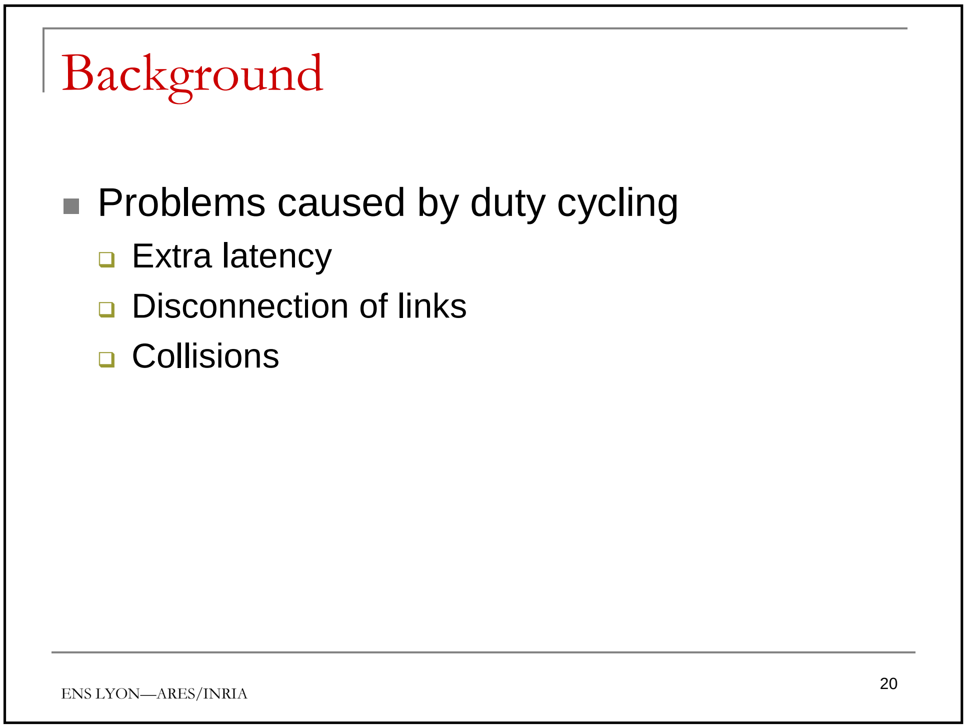
Background

- Problems caused by duty circling
  - Collisions
    - Low traffic, but collision is still a concern
    - e.g. if each node is awake in one of k slots
      - transmission that was distributed in k slots, now happen in one slot



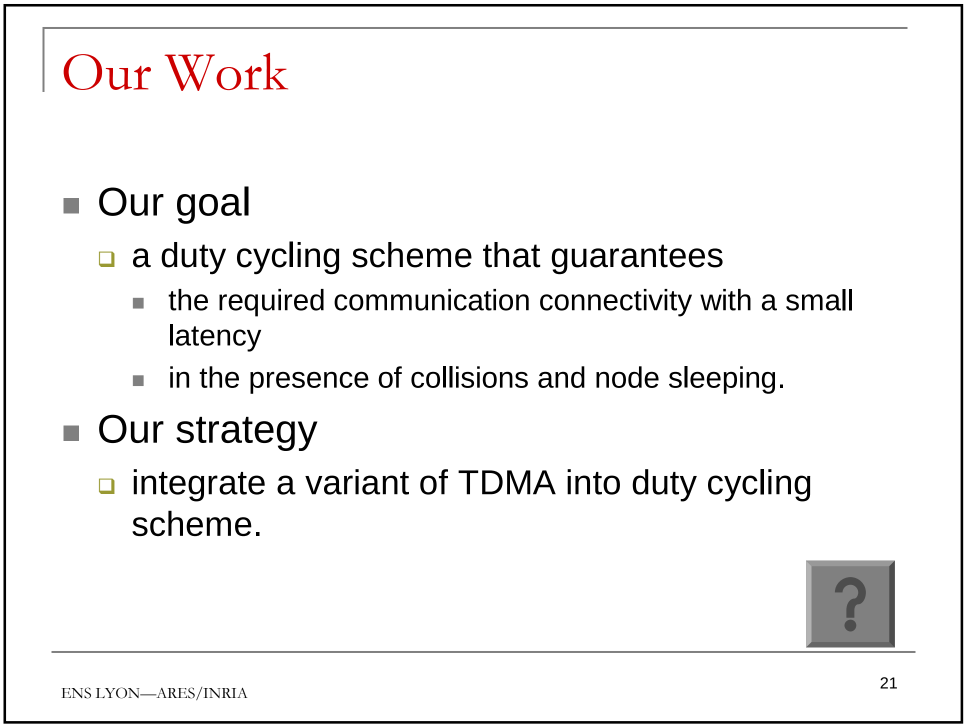
Background

- Problems caused by duty circling
  - Collisions
    - Low traffic, but collision is still a concern
    - e.g. if each node is awake in one of k slots
      - transmission that was distributed in k slots, now happen in one slot
  - Existing Solution:
    - contention-based scheme



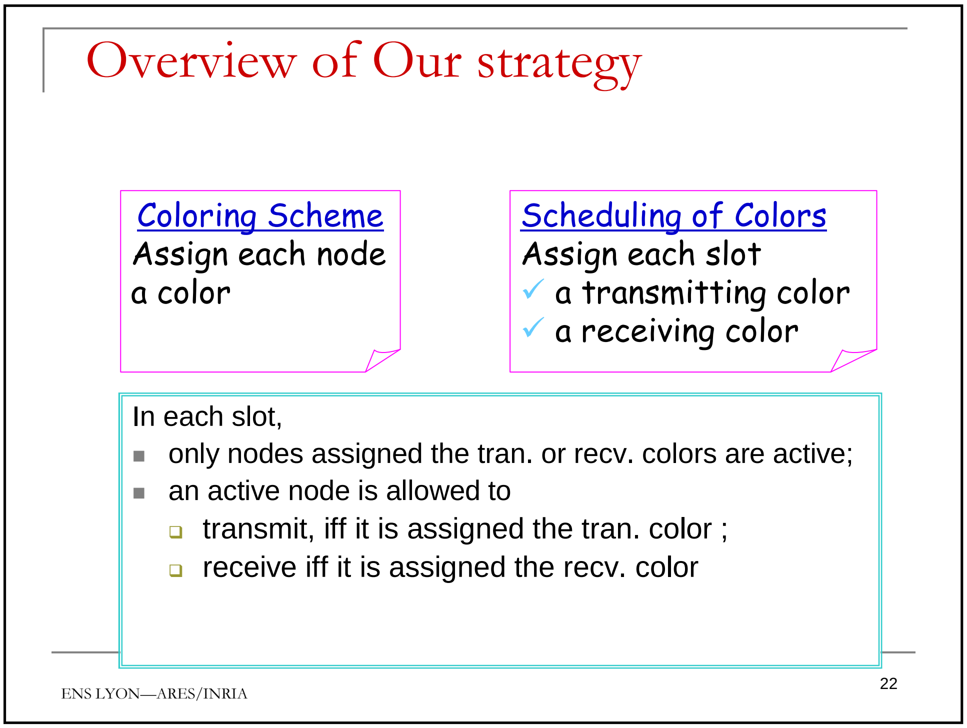
Background

- Problems caused by duty cycling
  - Extra latency
  - Disconnection of links
  - Collisions



Our Work

- Our goal
  - a duty cycling scheme that guarantees
    - the required communication connectivity with a small latency
    - in the presence of collisions and node sleeping.
- Our strategy
  - integrate a variant of TDMA into duty cycling scheme.



Overview of Our strategy

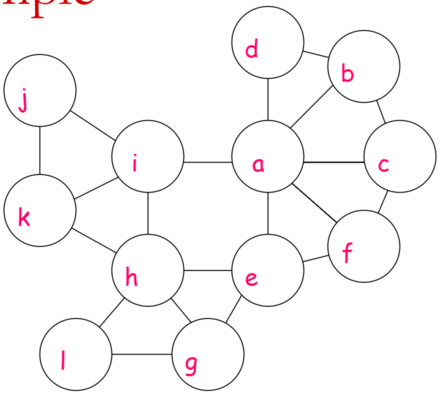
Coloring Scheme
Assign each node a color

Scheduling of Colors
Assign each slot
✓ a transmitting color
✓ a receiving color

In each slot,

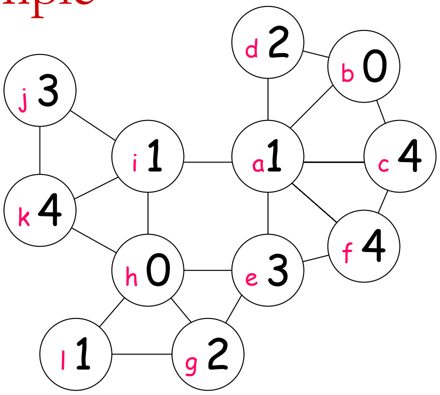
- only nodes assigned the tran. or recv. colors are active;
- an active node is allowed to
  - transmit, iff it is assigned the tran. color ;
  - receive iff it is assigned the recv. color

An Example



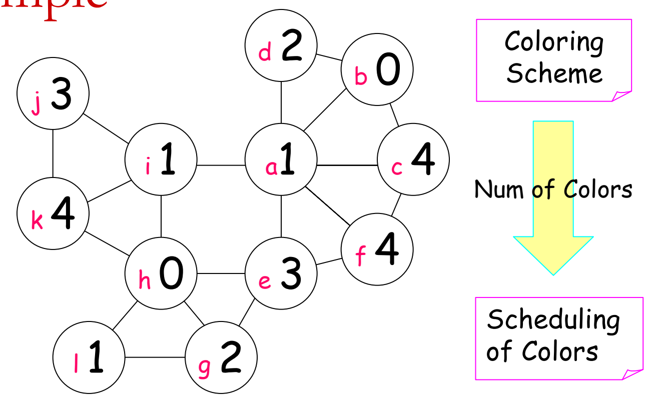
Coloring Scheme

An Example



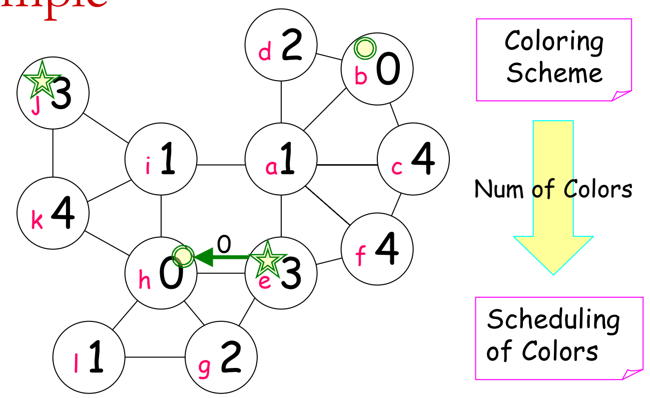
Coloring Scheme

An Example

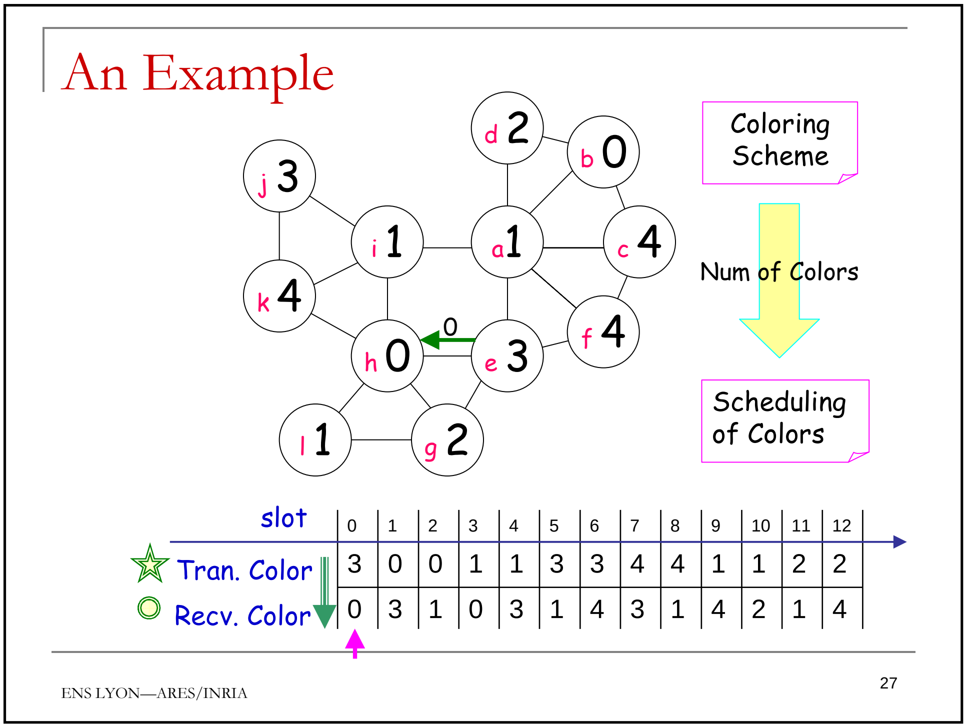


slot	0	1	2	3	4	5	6	7	8	9	10	11	12
Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4

An Example



slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
◎ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4



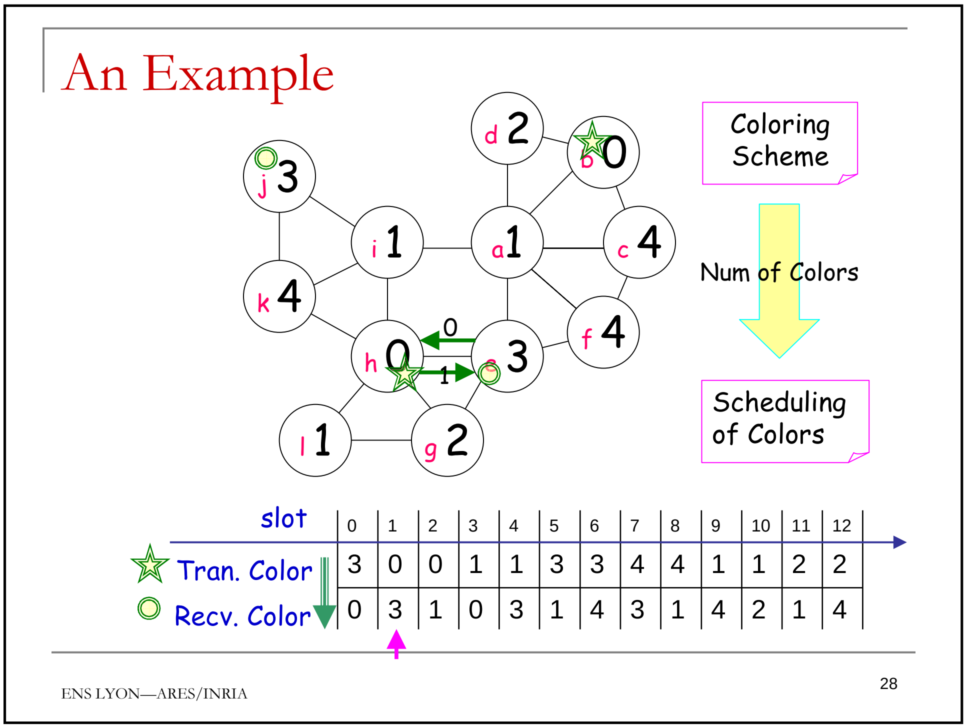
An Example

Coloring Scheme

Num of Colors

Scheduling of Colors

slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
○ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4



An Example

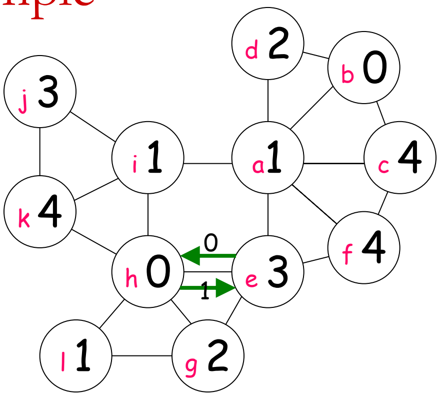
Coloring Scheme

Num of Colors

Scheduling of Colors

slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
○ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4

An Example



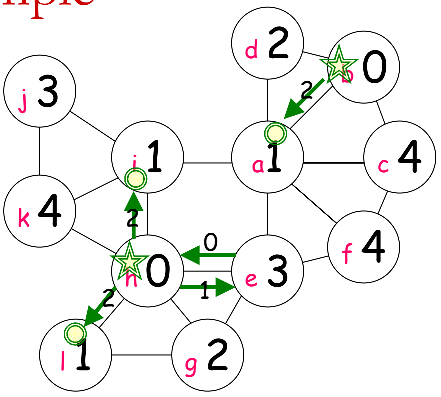
Coloring Scheme

Num of Colors

Scheduling of Colors

slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
◎ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4

An Example

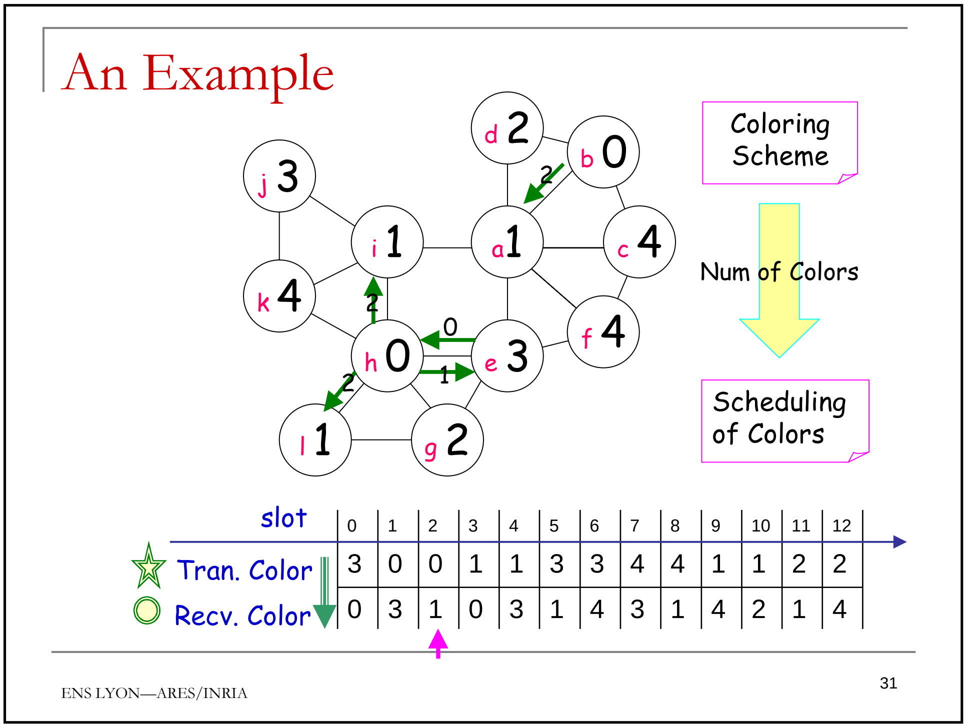


Coloring Scheme

Num of Colors

Scheduling of Colors

slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
◎ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4



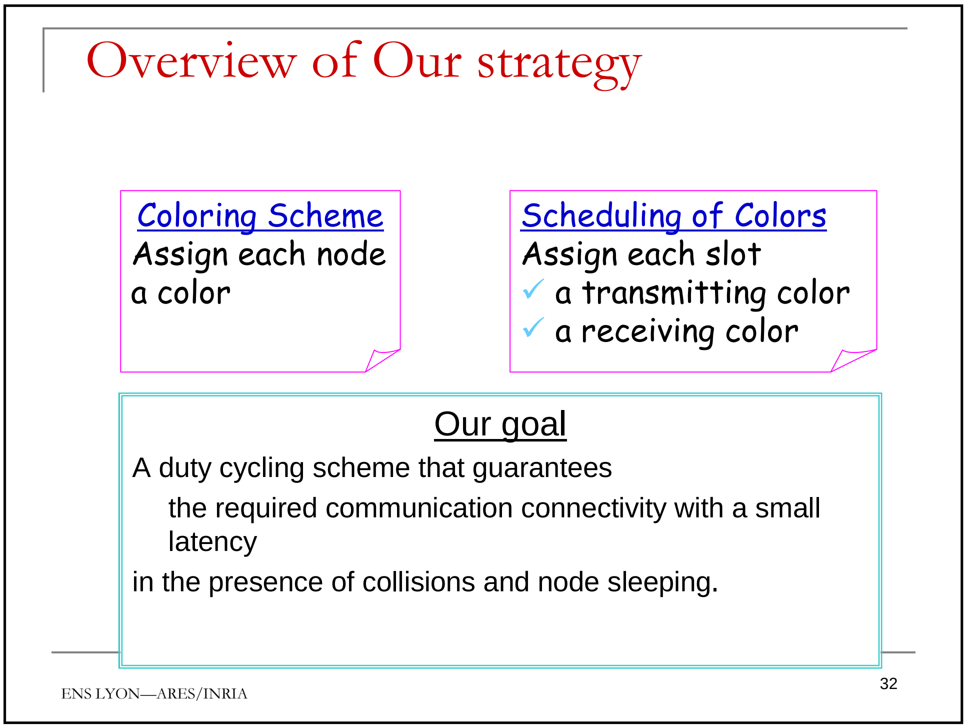
An Example

Coloring Scheme

Num of Colors

Scheduling of Colors

slot	0	1	2	3	4	5	6	7	8	9	10	11	12
☆ Tran. Color	3	0	0	1	1	3	3	4	4	1	1	2	2
○ Recv. Color	0	3	1	0	3	1	4	3	1	4	2	1	4



Overview of Our strategy

Coloring Scheme
Assign each node a color

Scheduling of Colors
Assign each slot
- ✓ a transmitting color
- ✓ a receiving color

Our goal

A duty cycling scheme that guarantees

the required communication connectivity with a small latency

in the presence of collisions and node sleeping.

Coloring Scheme

- Coloring Scheme
  - A new coloring definition
    - Why we need a new definition?
  - Analyses on the number of required colors
  - Our coloring heuristics

Traditional coloring scheme

- Graph labeling with channel separation constraints:
  - integers parameters *d*1, *d*2, … , d_k ares used to describe the channel separation constraints;
  - in particular, d_i is the minimum spacing between the subchannels assigned to nodes that are distance *i* from each other.

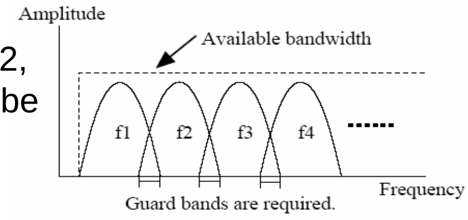
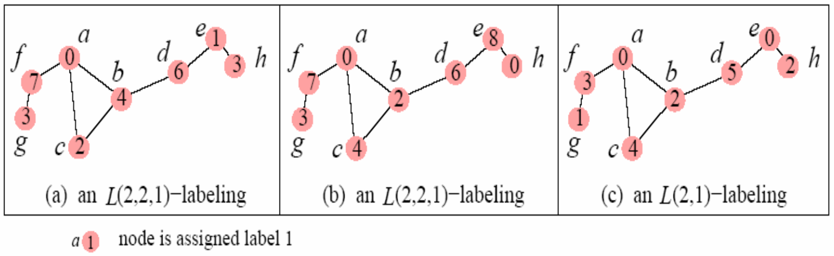
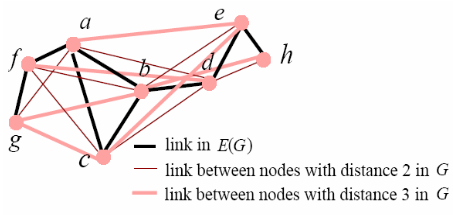


Figure 1. Adjacent channel interference

Graph labeling example



(a) an $L(2,2,1)$–labeling (b) an $L(2,2,1)$–labeling (c) an $L(2,1)$–labeling



a
e
f
b
d
h
g
c
link in E(G)
link between nodes with distance 2 in G
link between nodes with distance 3 in G

Traditional coloring definitions

- L(1,1)-coloring, L(d,k)-coloring (e.g. [6][3])
  - provides an entirely collision-free schedule.
  - Given a network G, for any two nodes x,y

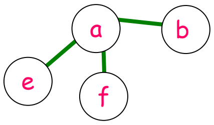
$$|\text{color}(x)-\text{color}(y)| >= \begin{cases} d \text{ if } y \in N_G(x) \\ k \text{ if } \exists z \in N_G(x)\ y \in N_G(z) \end{cases}$$

[6] I. Chlamtac and S. S. Pinter. Distributed nodes organization algorithm for channel access in a multihop dynamic radio network. *IEEE Transactions on Computers*, 1987.
[3] R. Battiti, A. A. Bertossi, and M. A. Bonuccelli. Assigning codes in wireless networks: bounds and scaling properties. *Wireless Networks*, 1999.

Traditional coloring definitions

- L(1,1)-coloring, L(d,k)-coloring (e.g. [6][3])
  - provides an entirely collision-free schedule.
  - Given a network G, for any two nodes x,y

$$|\text{color}(x)-\text{color}(y)| >= \begin{cases} d & \text{if } y \in N_G(x) \\ k & \text{if } \ni z \in N_G(x)\ y \in N_G(z) \end{cases}$$



L(d,k)-coloring: |color(a)-color(b)| >=d,
|color(e)-color(b)| >=k

Traditional coloring definitions

- L(1,1)-coloring, L(d,k)-coloring
  - provides an entirely collision-free schedule
- However, a large number of colors are required in dense networks.
  - Best upper bound [5] : $\Delta G^2+(d-1)\ \Delta G$, where ΔG,is the de

Can we use a smaller number of colors to preserve communication connectivity?

[5] G. Chang, W. Ke, D. Kuo, D. Liu, and R. Yeh. On l(d,1)-labeling of graphs. *Discrete Mathematics*, 220:57-66, 2000.

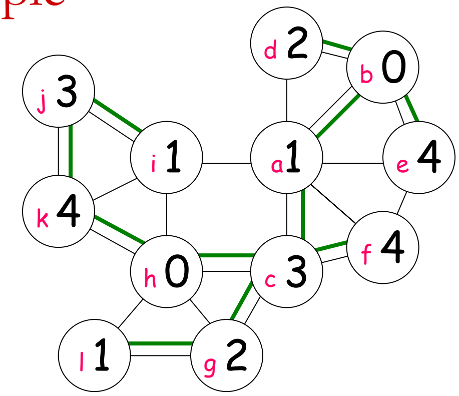
Traditional coloring definitions

- L(1,1)-coloring, L(d,k)-coloring
  - provides an entirely collision-free schedule
- However, a large number of colors are required in dense networks.
  - Best upper bound [5] : ΔG2+(d-1) ΔG, where ΔG,is the degree of G.

Not always necessary

[5] G. Chang, W. Ke, D. Kuo, D. Liu, and R. Yeh. On l(d,1)-labeling of graphs. *Discrete Mathematics*, 220:57-66, 2000.

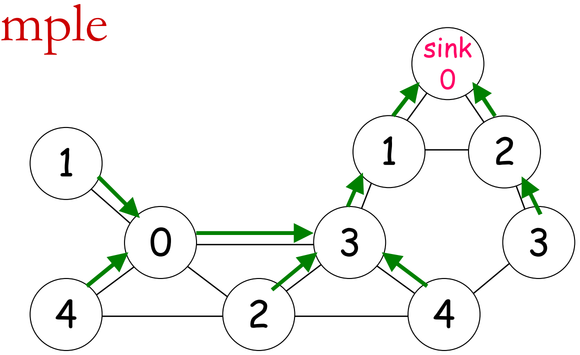
An Example



<u>Communication between any pair of nodes</u>

Strong connectivity,
e.g. a spanning tree

An Example

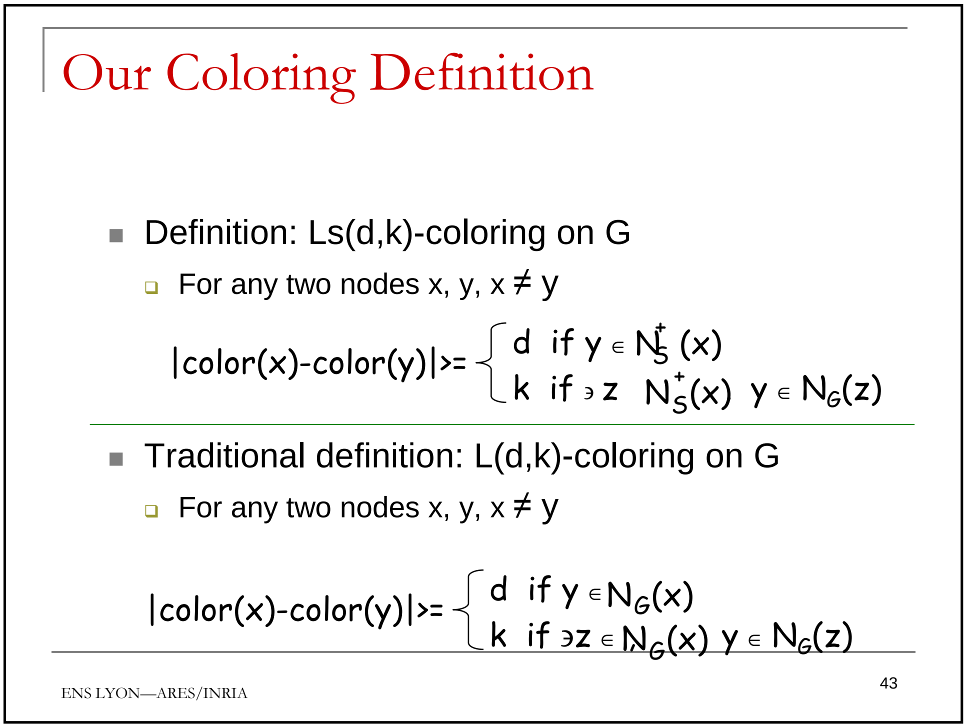


Data gathering

connectivity from each node to sink,
e.g. a tree rooted at the sink with edges towards the sink

Our Coloring Definition

- Given a network G,
  - Parameter: subgraph S⊆G
    - S represents applications' requirement on connectivity, e.g.
      - Spanning tree
      - Directed tree
  - We define Ls(d,k)-coloring
    - To guarantee links in S are collision free



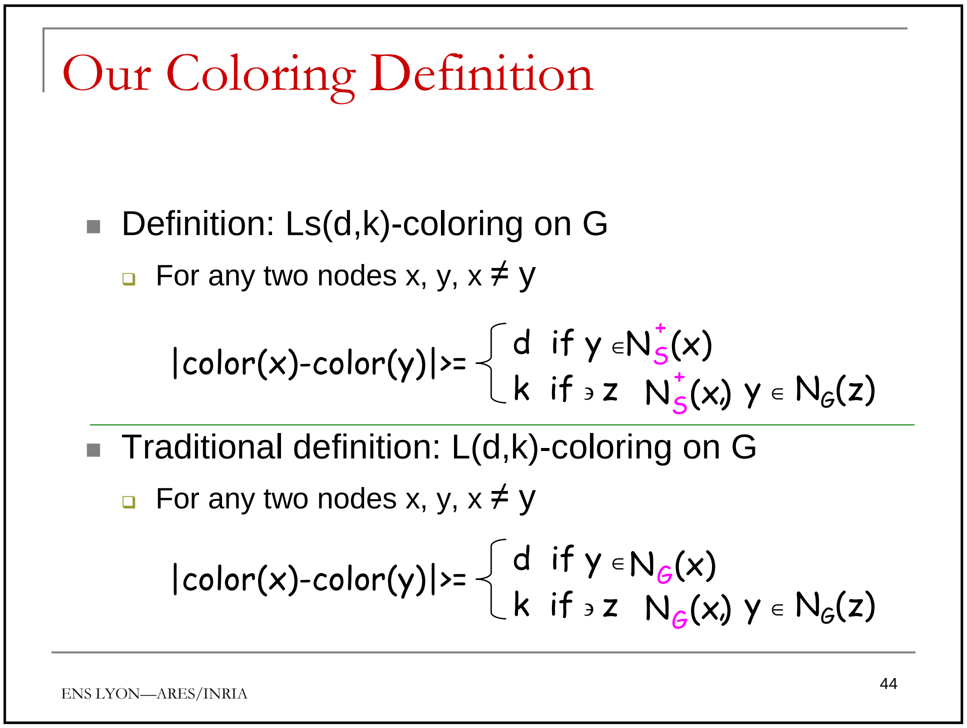
Our Coloring Definition

- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|\mathrm{color}(x)-\mathrm{color}(y)| >= \begin{cases} d & \text{if } y \in N_S^{+}(x) \\ k & \text{if } \ni z \ N_S^{+}(x) \ y \in N_G(z) \end{cases}$$

- Traditional definition: L(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|\mathrm{color}(x)-\mathrm{color}(y)| >= \begin{cases} d & \text{if } y \in N_G(x) \\ k & \text{if } \ni z \in N_G(x) \ y \in N_G(z) \end{cases}$$



Our Coloring Definition

- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|\mathrm{color}(x)-\mathrm{color}(y)| >= \begin{cases} d & \text{if } y \in N_S^{+}(x) \\ k & \text{if } \ni z \ N_S^{+}(x) \ y \in N_G(z) \end{cases}$$

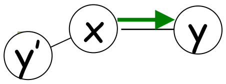
- Traditional definition: L(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|\mathrm{color}(x)-\mathrm{color}(y)| >= \begin{cases} d & \text{if } y \in N_G(x) \\ k & \text{if } \ni z \ N_G(x) \ y \in N_G(z) \end{cases}$$

Our Coloring Definition

- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|color(x)-color(y)|>= \begin{cases} d \text{ if } y \in N^{+}_{S}(x) \\ k \text{ if } \ni z\ N^{+}_{S}(x)\ y \in N_G(z) \end{cases}$$



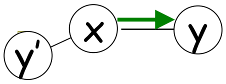
$y \in N^{+}_{S}(x)$

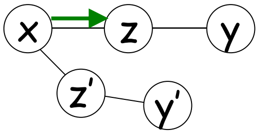
Link in S

Our Coloring Definition

- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|color(x)-color(y)|>= \begin{cases} d \text{ if } y \in N^{+}_{S}(x) \\ k \text{ if } \ni z\square\ N^{+}_{S}(x)\ y \in N_G(z) \end{cases}$$





$y \in N^{+}_{S}(x)$

$z \in N^{+}_{S}(x)$, y in $N_G(z)$

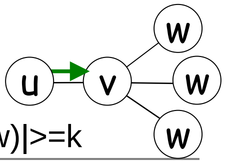
Link in S

Our Coloring Definition

- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|color(x)-color(y)|>= \begin{cases} d & \text{if } y \in N_S^{+}(x) \\ k & \text{if } z□ \; N_S^{+}(x) y \in N_G(z) \end{cases}$$

For any link <u→v> in S, u's color is distinguished in $N_G(v)$



- ✓ v ∈ $N_S^{+}(u)$ → |color(u)-color(v)|>=d
- ✓ for all w ∈ $N_G(v)$, w ≠ u,

  v ∈ $N_S^{+}(u)$w ∈ $N_G(v)$ → |color(u)-color(w)|>=k

Our Coloring Definition

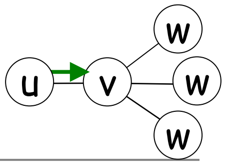
- Definition: Ls(d,k)-coloring on G
  - For any two nodes x, y, x ≠ y

$$|color(x)-color(y)|>= \begin{cases} d & \text{if } y□ N_S^{+}(x) \\ k & \text{if}□z□ \; N_S^{+}(x) \; y□N_G(z) \end{cases}$$

Equivalent

For any link <u→v> in S, u's color is distinguished in $N_G(v)$

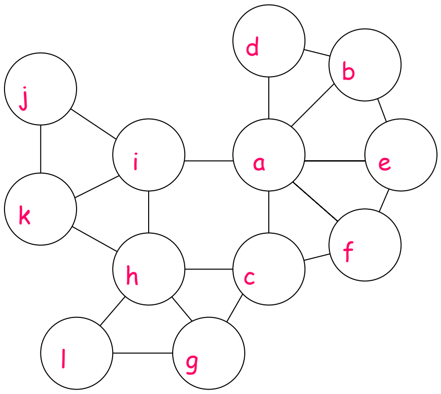
By setting d and k appropriately, communication from u to v can be guaranteed



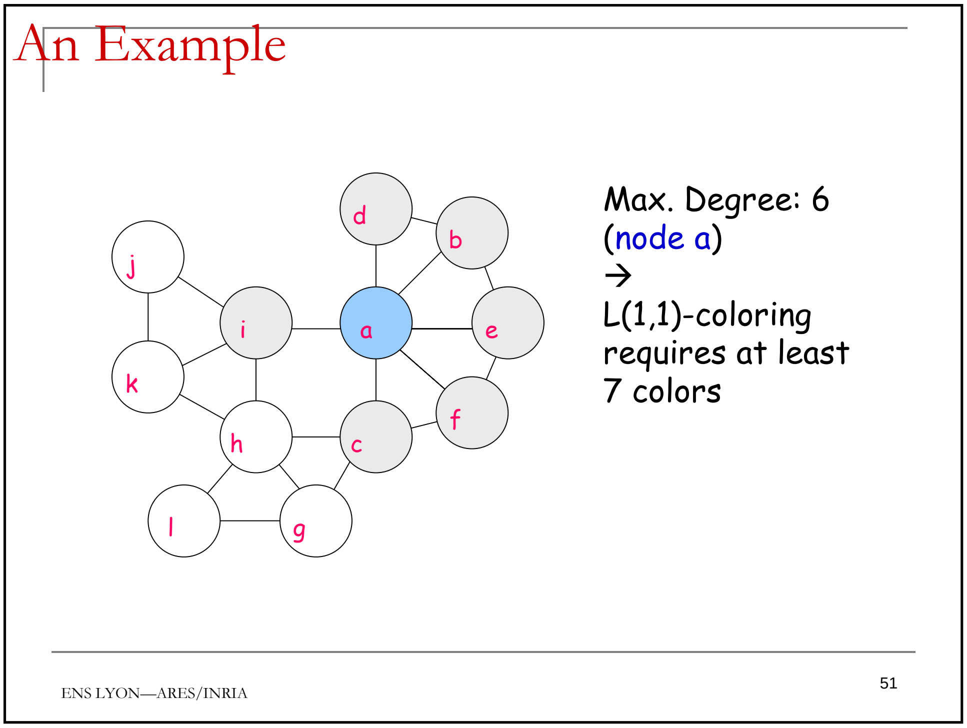
Coloring Scheme

- Coloring Scheme
  - A new coloring definition
  - Analyses on the number of colors required by Ls(d,1)
  - Heuristic

An Example

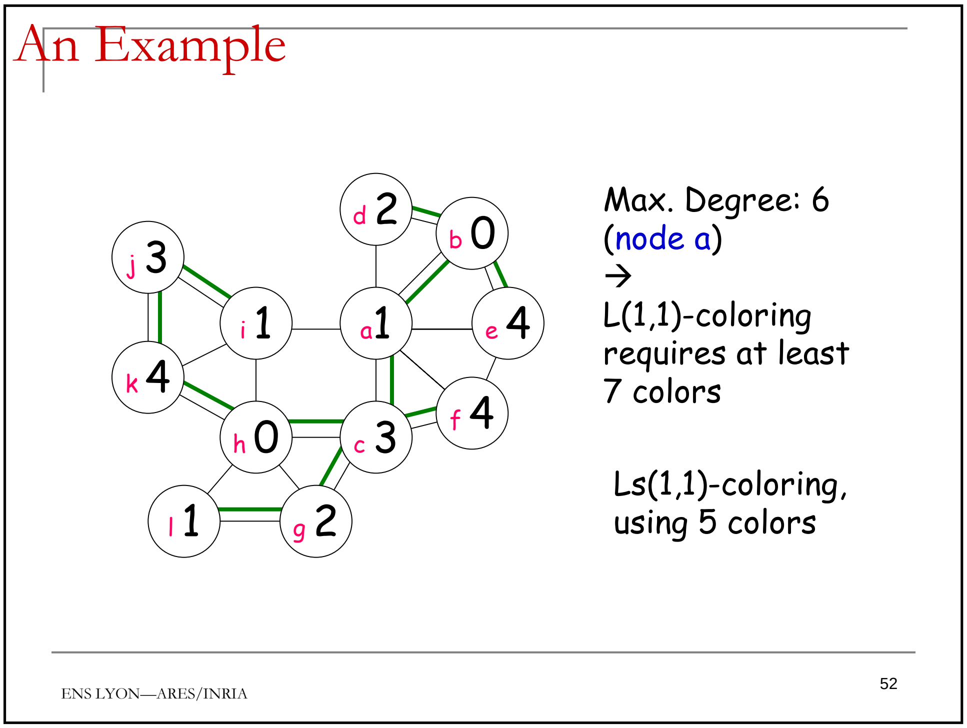


Max. Degree: 6 (node a)
→
L(1,1)-coloring requires at least 7 colors



An Example

Max. Degree: 6
(node a)
→
L(1,1)-coloring
requires at least
7 colors

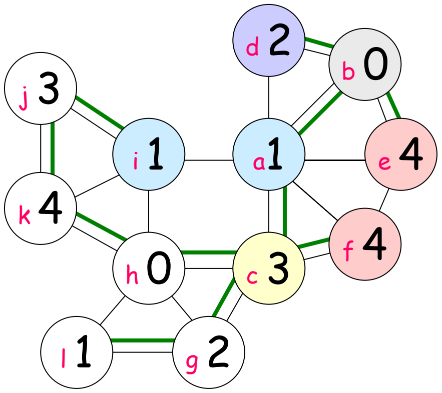


An Example

Max. Degree: 6
(node a)
→
L(1,1)-coloring
requires at least
7 colors

Ls(1,1)-coloring,
using 5 colors

An Example



Max. Degree: 6
(node a)
→
L(1,1)-coloring requires at least 7 colors

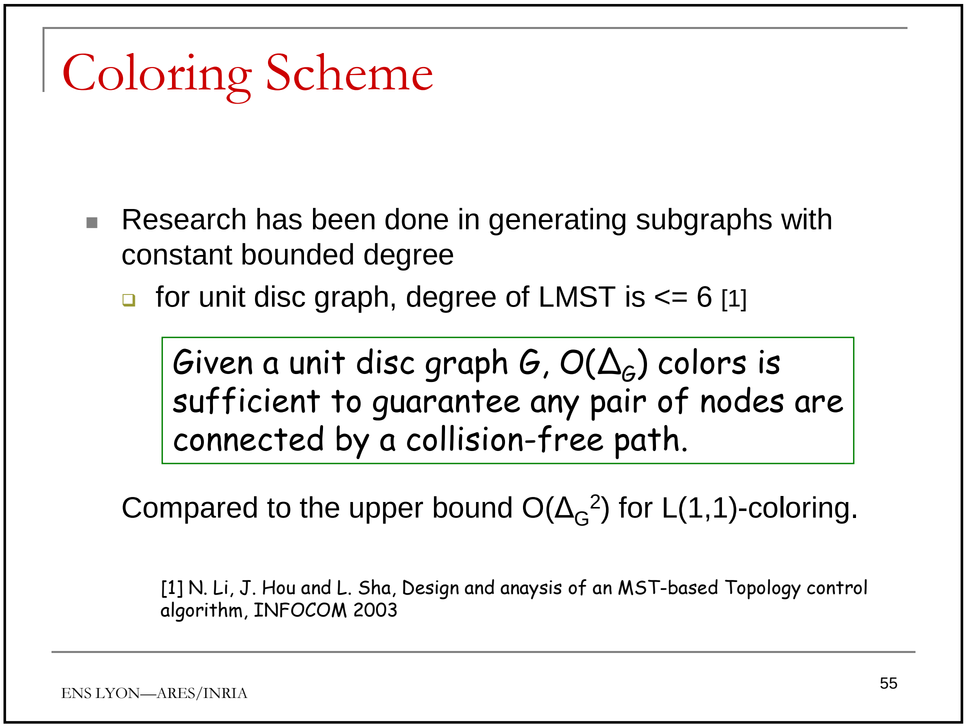
Ls(1,1)-coloring, using 5 colors

Coloring Scheme

- An upper bound on the number of colors required by Ls(d,1)

$$\min\{\Delta_G^2, \Delta_S + 2\Delta_G\Delta_S\} + (d-1)\Delta_S$$

  - Δ_G : degree of G,
  - Δ_S: degree of S
- When G is dense,
  - Upper bound: $2\Delta_S \Delta_G + d\,\Delta_S = O(\Delta_S \Delta_G)$
- If Δ_S is a bounded by a constant
  - Upper bound: $O(\Delta_G)$



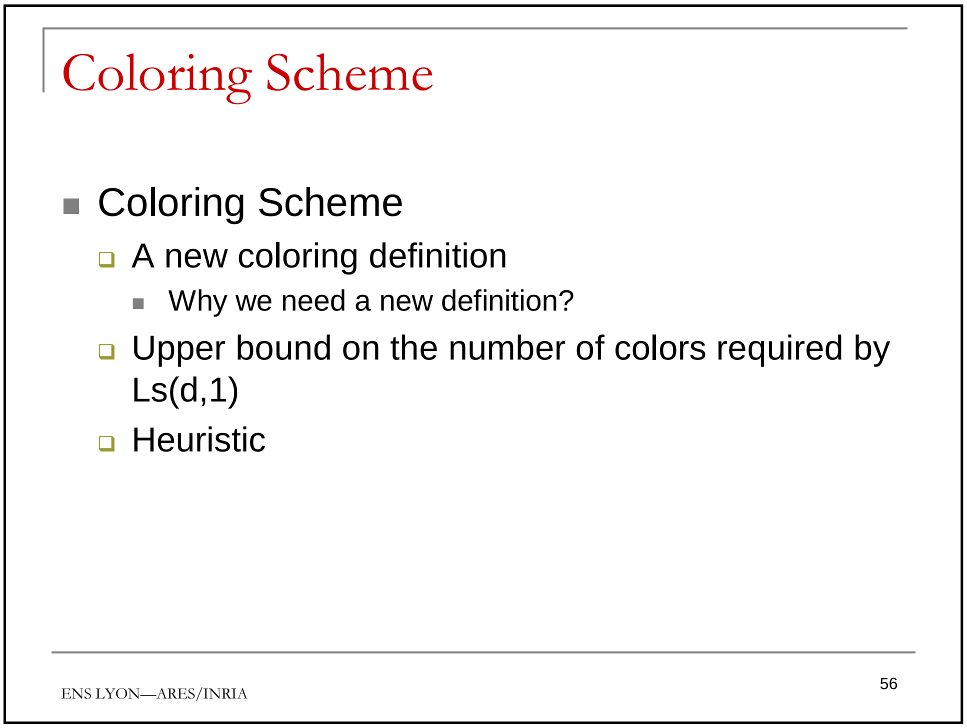
Coloring Scheme

- Research has been done in generating subgraphs with constant bounded degree
  - for unit disc graph, degree of LMST is <= 6 [1]

> Given a unit disc graph G, O(Δ_G) colors is sufficient to guarantee any pair of nodes are connected by a collision-free path.

Compared to the upper bound O(Δ_G^2) for L(1,1)-coloring.

[1] N. Li, J. Hou and L. Sha, Design and anaysis of an MST-based Topology control algorithm, INFOCOM 2003



Coloring Scheme

- Coloring Scheme
  - A new coloring definition
    - Why we need a new definition?
  - Upper bound on the number of colors required by Ls(d,1)
  - Heuristic

Coloring Scheme

- NP-Complete, as $L_G(1,1)$ is NP-Complete
- Heuristic for Ls(1,1) (follows traditional heuristic)
  - Assign priority for each node, e.g. ID, or degree in S and breaking ties by ID.
  - Coloring based on priority
    - The node with highest priority gets color 0
    - Each node waits until all those with higher priority in its two hop-neighborhood are colored.
      - It assigns itself a minimum color that does not invalidate Ls(1,1) constraints.
  - # of used colors are evaluated by simulations

Overview of Our strategy

Coloring Scheme

Assign each node a color ($L_S(1,1)$-coloring)

Schedule of Colors

Assign each slot

✓ a transmitting color

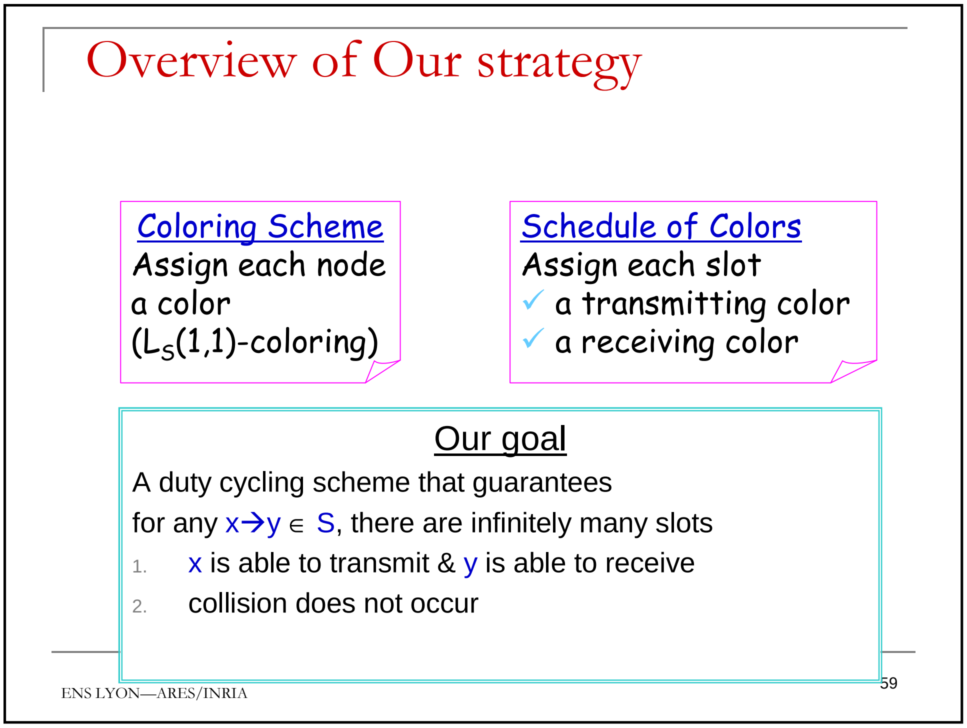
✓ a receiving color

Our goal

A duty cycling scheme that guarantees

the required communication connectivity

in the presence of collisions and node sleeping.



Overview of Our strategy

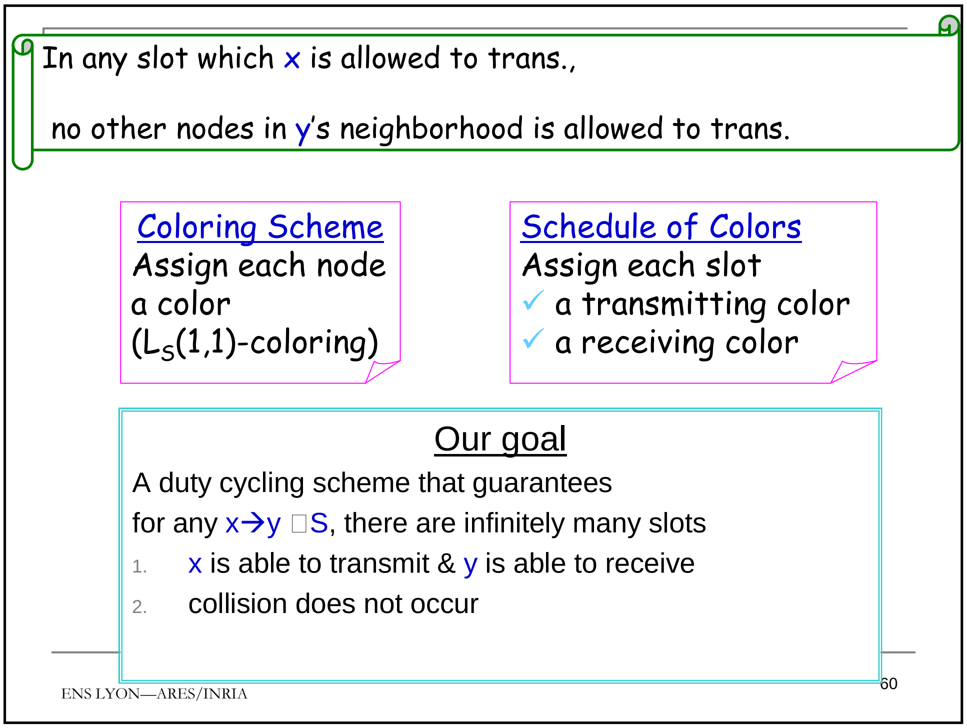
Coloring Scheme
Assign each node a color ($L_S(1,1)$-coloring)

Schedule of Colors
Assign each slot
- ✓ a transmitting color
- ✓ a receiving color

Our goal

A duty cycling scheme that guarantees for any x→y ∈ S, there are infinitely many slots

1. x is able to transmit & y is able to receive
2. collision does not occur



In any slot which x is allowed to trans.,

no other nodes in y's neighborhood is allowed to trans.

Coloring Scheme
Assign each node a color ($L_S(1,1)$-coloring)

Schedule of Colors
Assign each slot
- ✓ a transmitting color
- ✓ a receiving color

Our goal

A duty cycling scheme that guarantees for any x→y □S, there are infinitely many slots

1. x is able to transmit & y is able to receive
2. collision does not occur

Our goal can be achieved if

✓ for any two colors a and b, there are infinitely many slots in which a is the transmitting color and b is the receiving color.

Coloring Scheme

Assign each node a color ($L_S(1,1)$-coloring)

Schedule of Colors

Assign each slot

✓ a transmitting color

✓ a receiving color

Our goal

A duty cycling scheme that guarantees

for any $x \rightarrow y \in S$, there are infinitely many slots

1. x is able to transmit & y is able to receive
2. collision does not occur

Color Scheduling Scheme

- Technically, given the number of colors K, the schedule can be constructed from any permutation of K(K-1) pairs of different colors

e.g.

Tran color	0	0	...	0	1	1	...	1	......	K-1	K-1	...	K-1
Recv color	1	2	...	K-1	0	2	...	K-1	......	0	1	...	K-2

- The fraction of slots in which a node is active 2/K.

Color Scheduling Scheme

- Technically, given the number of colors K, the schedule can be constructed from any permutation of K(K-1) pairs of different color

e.g.

Tran color	0	0	...	0	1	1	...	1	......	K-1	K-1	...	K-1
Recv color	1	2	...	K-1	0	2	...	K-1	......	0	1	...	K-2

color 0: switch K-1 times in K(K-1) slots

other color: swtich K times in K(K-1) slots

  - Nodes need energy to switch between different modes → reduce the number of switches.

Color Scheduling Scheme

- A lower bound on the number of switches in K(K-1) slots: $\left\lfloor \frac{K}{2} \right\rfloor$
  - K is # of colors.
- We propose a schedule of colors that achieves this lower bound.

Color Scheduling Scheme

When S is much sparser than G, energy consumption can be further reduced by

scheduling each node x active only in slots in which communication is possible:

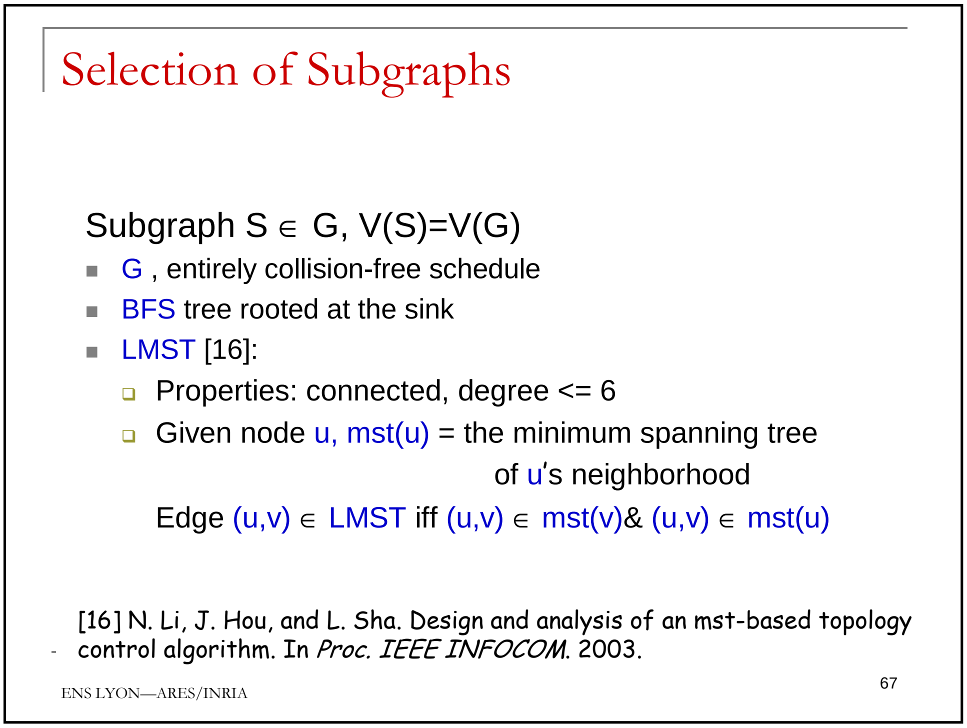
- x has the tran. color & one of x's neighbors (in S) has the recv. color or
- one of x's neighbors (in S) has the tran. color & x has the recv. color

The fraction of slots in which node x stays active

$$\frac{\delta_S^{+}(x)+\delta_S^{-}(x)}{K(K-1)} \leq \frac{2\Delta_S}{K(K-1)}$$

Simulations

- We consider several levels of connectivity,
  - represented by different types of subgraphs
- For each type of subgraph:
  - # of colors used by our coloring scheme
  - the latency under our duty cycling scheme



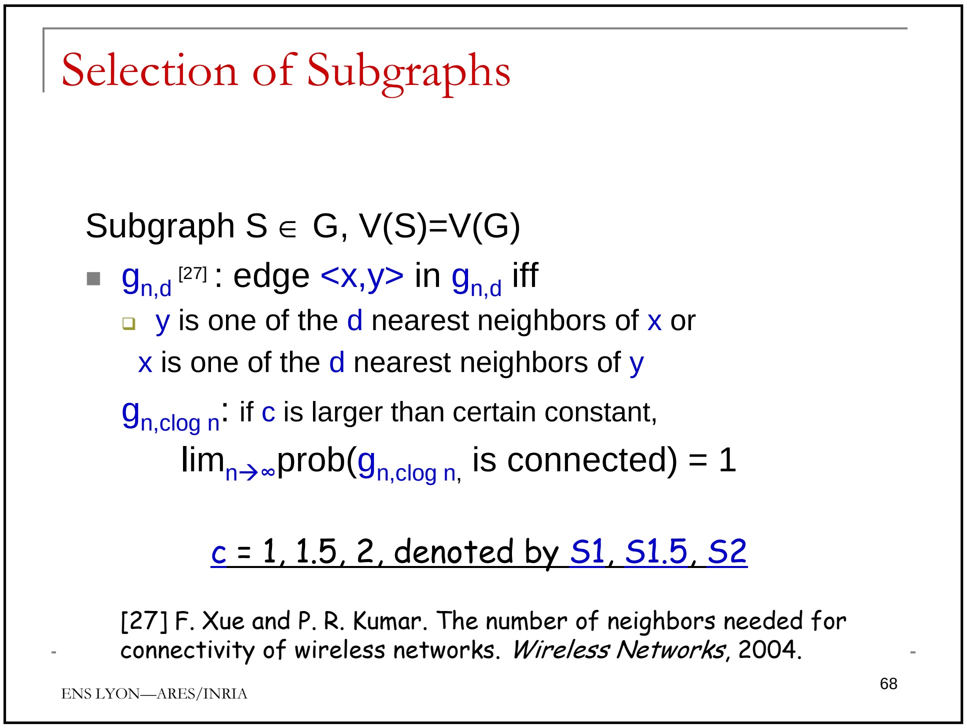
Selection of Subgraphs

Subgraph $S \in G$, $V(S)=V(G)$

- G , entirely collision-free schedule
- BFS tree rooted at the sink
- LMST [16]:
  - Properties: connected, degree <= 6
  - Given node u, mst(u) = the minimum spanning tree of u's neighborhood

  Edge $(u,v) \in$ LMST iff $(u,v) \in mst(v)$& $(u,v) \in mst(u)$

[16] N. Li, J. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. In *Proc. IEEE INFOCOM*. 2003.



Selection of Subgraphs

Subgraph $S \in G$, $V(S)=V(G)$

- $g_{n,d}$ [27] : edge <x,y> in $g_{n,d}$ iff
  - y is one of the d nearest neighbors of x or x is one of the d nearest neighbors of y

  $g_{n,c\log n}$: if c is larger than certain constant,

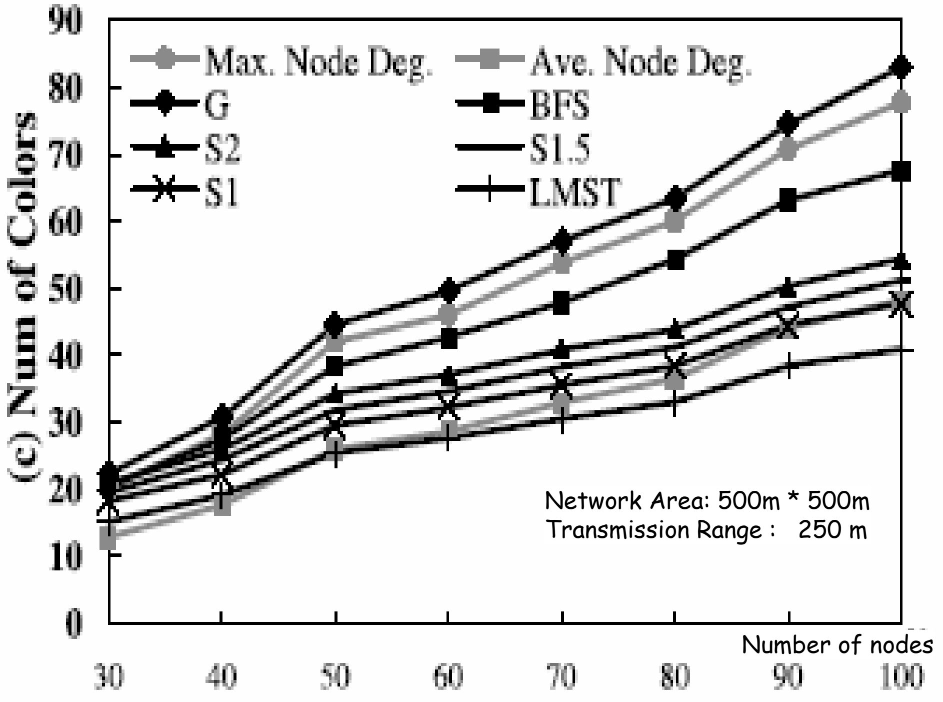
  $\lim_{n\to\infty}$prob($g_{n,c\log n,}$ is connected) = 1

<u>c = 1, 1.5, 2, denoted by S1, S1.5, S2</u>

[27] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 2004.

Simulation Results

- For each type of subgraph,
  - # of colors used by our coloring scheme.



Network Area: 500m * 500m
Transmission Range : 250 m

Simulation Results

- Latency of duty cycling schedule,
  - ❑ Compared to two other duty cycling schemes:

    Sim: nodes follow the same schedule

    Random-Avg[18]: nodes wake up to receive in one of k slots; nodes can transmit in any slot.

In both schemes, collisions are assumed to be resolved in one slot.

[18] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel. Delay efficient sleep scheduling in wireless sensor networks. In *Proc. IEEE INFOCOM,* 2005

Duty Cycling Performance

The same fraction of slots in which sensors are active

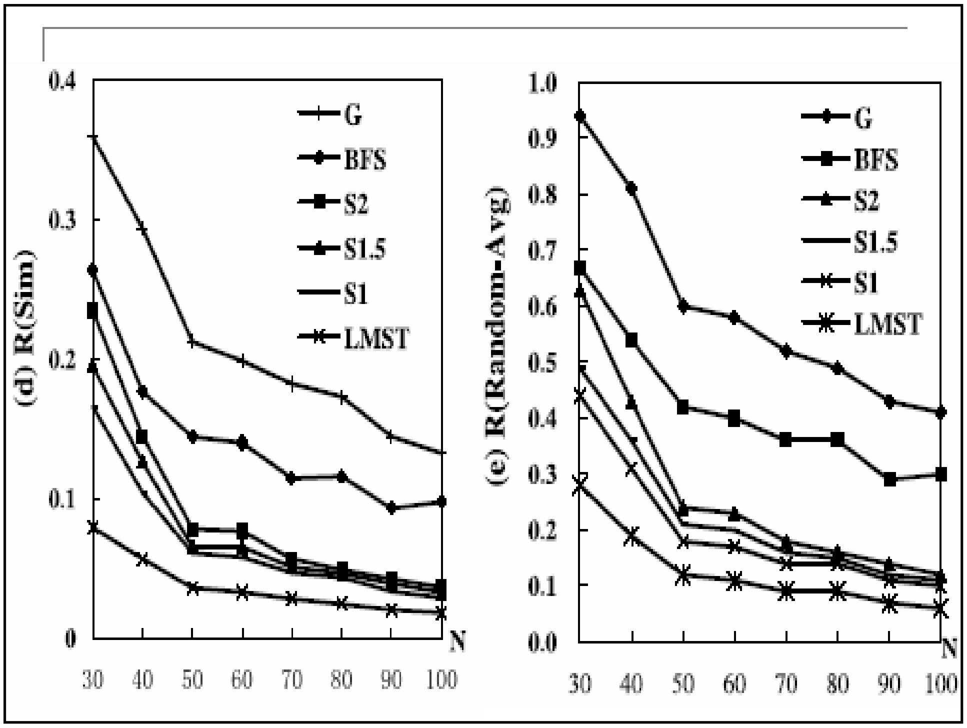
$$R(Sim)=\frac{delay}{delay'\ \Delta_G} \qquad R(Random\text{-}Avg)= \frac{delay}{delay'\ \Delta_G}$$

- delay : the delay under our scheme and
- delay': the delay under Sim or Random-Avg

if a contention-based scheme is used to handle collisions,

Δ_G R(sim) or Δ_GR(Random-Avg):

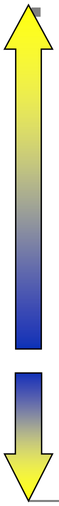
the number of slots within which contention should be resolved to achieve the same performance as our scheme.



Conclusions

- A new coloring definition
  - Analyses on the number of required colors
  - Heuristics
- Duty cycling scheme
- Simulations

Small Technology, Broad Agenda



- Social factors
  - security, privacy, information sharing
- Applications
  - long lived, self-maintaining, dense instrumentation of previously unobservable phenomena
  - interacting with a computational environment
- Programming the Ensemble
  - describe global behavior, synthesis local rules that have correct, predictable global behavior
- Distributed services
  - localization, time synchronization, resilient aggregation
- Networking
  - self-organizing multihop, resilient, energy efficient routing
  - despite limited storage and tremendous noise
- Operating system
  - extensive resource-constrained concurrency, modularity
  - framework for defining boundaries
- Architecture
  - rich interfaces and simple primitives allowing cross-layer optimization
- Components
  - low-power processor, ADC, radio, communication, encryption, sensors, batteries

The Time is Right

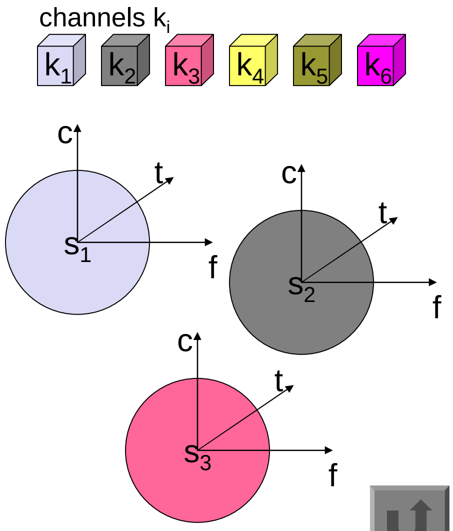
- Don’t be afraid to go out and tackle REAL problems.
- They often reveal interesting challenges.
- The technology is (just barely) ready for it.
- There is much innovation ahead.

Thank You

Questions & Comments?

Multiplexing

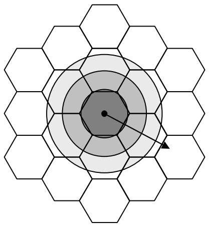
- Multiplex channels (k) in four dimensions
  - space (s)
  - time (t)
  - frequency (f)
  - code (c)
- Goal: multiple use of a shared medium
- Important: guard spaces needed!
- Example: radio broadcast



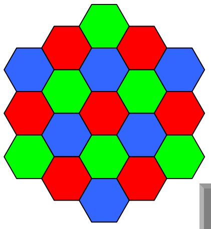
Example for space multiplexing: Cellular network

- VERY Simplified hexagonal model

- Signal propagation ranges: Frequency reuse only with a certain distance between the base stations



- Can you reuse frequencies in distance 2 or 3 (or more)?
- Graph coloring problem
- Example: fixed frequency assignment for reuse with distance 2
- Interference from neighbor cells (other color) can be controlled with transmit and receive filters



Frequency Division Multiplex (FDM)

- Separation of the whole spectrum into smaller frequency bands
- A channel gets a certain band of the spectrum for the whole time

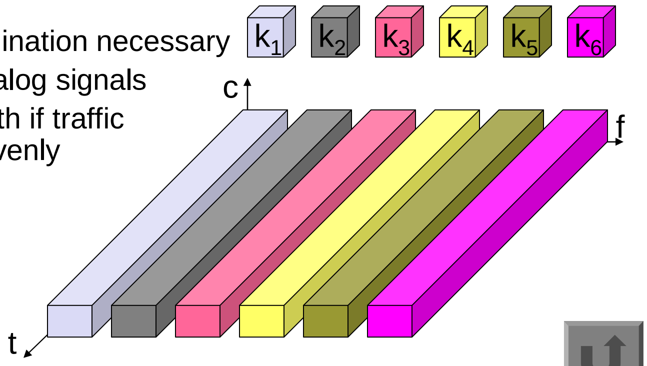
+ no dynamic coordination necessary

+ works also for analog signals

– waste of bandwidth if traffic is distributed unevenly

– inflexible

- Example: broadcast radio



Time Division Multiplex (TDM)

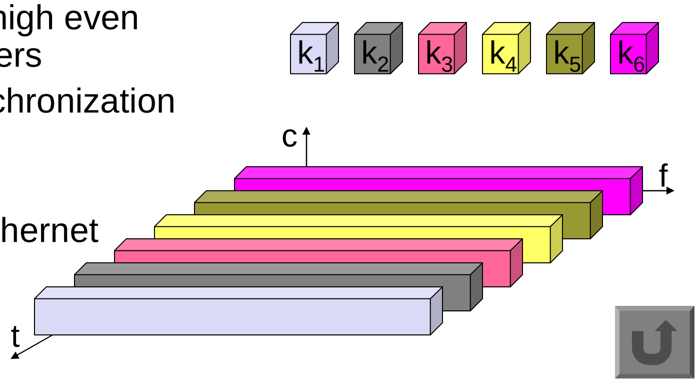
- A channel gets the whole spectrum for a certain amount of time

+ only one carrier in the medium at any time

+ throughput high even for many users

– precise synchronization necessary

- Example: Ethernet



Time and Frequency Division Multiplex

- Combination of both methods
- A channel gets a certain frequency band for some time

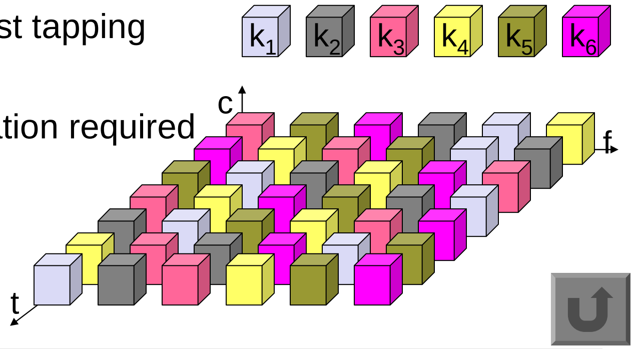
+ protection against frequency selective interference

+ protection against tapping

+ adaptive

– precise coordination required

- Example: GSM



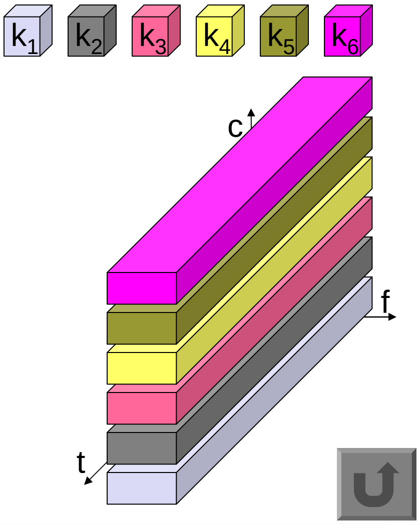
Code Division Multiplex (CDM)

- Each channel has a unique code
- All channels use the same spectrum at the same time

+ bandwidth efficient
+ no coordination or synchronization
+ hard to tap
+ almost impossible to jam

– lower user data rates
– more complex signal regeneration

- Example: UMTS
- Spread spectrum
- U. S. Patent 2‘292‘387, Hedy K. Markey (a.k.a. Lamarr or Kiesler) and George Antheil (1942)



Cocktail party as analogy for multiplexing

- Space multiplex: Communicate in different rooms
- Frequency multiplex: Use soprano, alto, tenor, or bass voices to define the communication channels
- Time Frequency: Class room principle
- Code multiplex: Use different languages and hone in on your language. The “farther apart” the languages the better you can filter the “noise”: German/Japanese better than German/Dutch. Can we have orthogonal languages?