A Distributed Policy Scheduling for Wireless Sensor Networks

Infocom 2007

Yu Chen, Eric Fleury

2008/02/21

Outline

- Introduction
- Coloring
- Color scheduling scheme
- Simulation
- Conclusion

Introduction

- Save energy by collision avoidance + duty cycling
- Although duty cycling can save power, it also disrupts network performance
  - Link disconnections
- Duty cycling needs synchronization to solve the above problems

Introduction(2)

- Save energy by integrating two tasks, collision avoidance and duty cycles, into one scheduling of sensors' activities
- This scheme consists of two parts:
  - ***A coloring scheme***
  - ***A color scheduling scheme***
- To reduce packet latency and save energy, guarantee the communication connectivity of links only in a sparse connected subgraph

Coloring

- Traditional coloring: L(d,k)
  - A function that assigns to each node an integer such that if x and y are adjacent, then $|color(x) - color(y)| \geq d$; if there is a two-edge path, then $|color(x) - color(y)| \geq k$
  - To guarantee an entirely collision-free schedule in WSN
- In this paper, reduce the number of required colors

Why $L_s(d,k)$?

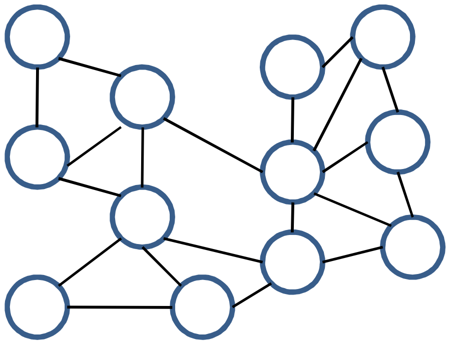
- As colors represent resources in networks, e.g. time slots in TDMA and codes in CDMA, due to high node density or constrained system resources.
- Communication among all pairs of nodes only requires collision-free links forming strongly connected graph
- Data gathering only requires that each node is connected to the sink by a collision-free path

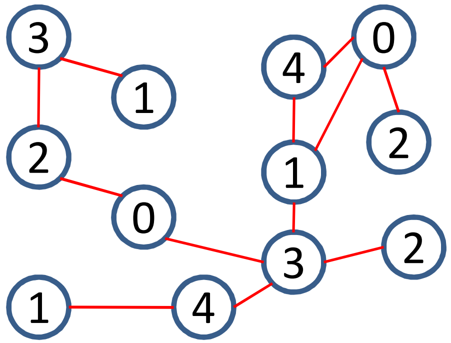
$L_s(d,k)$

- We don't need to consider the entire graph, just consider its subgraph
- [21]: relative neighborhood graphs; [22]: local minimal spanning trees
- Network topologies can be changed by adjusting nodes' transmission power and the number of colors can be reduced by only guaranteeing required connectivity

L_s(d,k)-coloring (1)

$$|color(x) - color(y)| \geq \begin{cases} d \text{ if } y \in N_S^+(x) \\ k \text{ if } \exists z \in N_S^+(x), y \in N_G(z) \end{cases}$$





index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
T[20]=	3	0	0	1	1	3	3	4	4	1	1	2	2	4	4	0	0	2	2	3
R[20]=	0	3	1	0	3	1	4	3	1	4	2	1	4	2	0	4	2	0	3	2

L_s(d,k)-coloring (2)

$$
\begin{aligned}
N_{S,G}(x) &\equiv N_S^+(x) \cup \{y \mid \exists z \in N_S^+(x), y \in N_G(z)\} \\
D_{S,G}(x) &\equiv N_{S,G}(x) \cup \{y \mid x \in N_{S,G}(y)\} \\
&= N_S^+(x) \cup \{y \mid \exists z \in N_S^+(x), y \in N_G(z)\} \cup \\
&\quad N_S^-(x) \cup \{y \mid \exists z \in N_G(x), y \in N_S^-(z)\}
\end{aligned}
$$

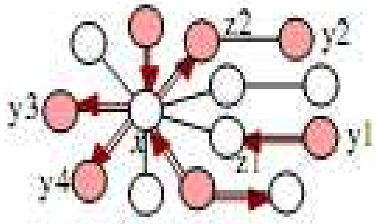


Fig. 2. Example of $N_{S,G}(x)$ and $D_{S,G}(x)$

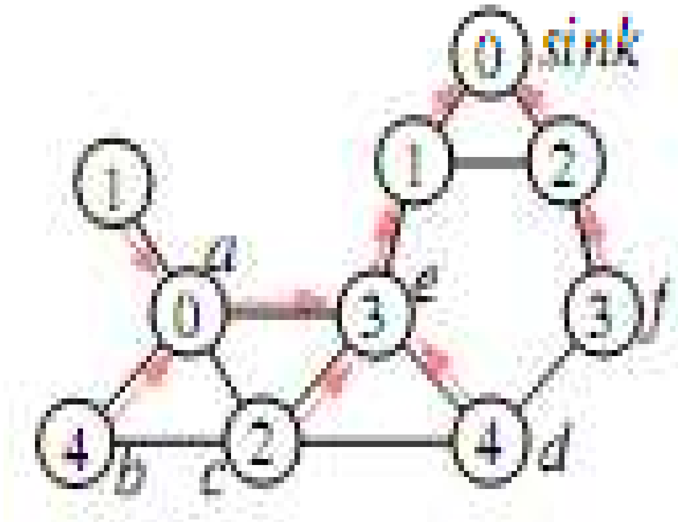
$L_s(1,1)$-coloring

$$wait_x \equiv \{z | z \in D_{S,G}(x) \wedge (priority(z) > priority(x))\}$$

- Distributed algorithm
- On node x:
  - when received y's message <y,color(y)>
  - If $y \in N_G(x)$, it broadcasts it
  - If $y \in wait(x)$, remove y from wait(x), and record color(y)
  - Until wait(x) is empty, it chooses the smallest color from remaining colors, then broadcasts itself

$L_T(1,1)$-coloring

- Data gathering from nodes to sink
- In BFS order



Color scheduling (1)

- Nodes are scheduled in (T,R); in slot i, a node is allowed to transmit iff its color is T[i%F], otherwise it turns off its radio to sleep
- To guarantees the connectivity of a link

<a,b>∈ S (fully connected schedule)

T=	0	0	...	0	1	1	...	1	...	K-1	K-1	...	K-1
R=	1	2	...	K-1	0	2	...	1	...	0	1	...	K-2

Color scheduling (2)

- To reduce the number of switching between sleep and active modes
- The following scheme is optimal:

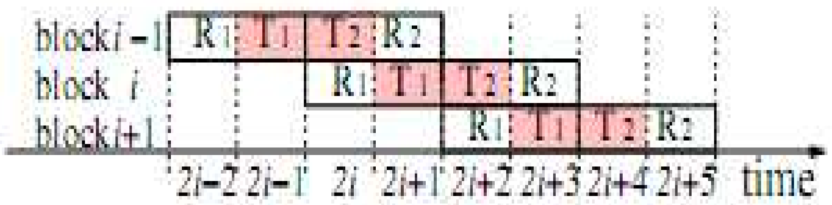


Fig. 10. Organize slots into blocks

Color scheduling (3)

- Block i is assigned to color $\sigma[i\%|\sigma|]$

$$\forall k \in [0, |\sigma|-1], R[2k] = T[2k+1] = T[(2k+2)\%2|\sigma|] = R[(2k+3)\%2|\sigma|] = \sigma[k]$$

- If $\exists k,\ (\sigma[k]=x \wedge \sigma[k+1]=y)$ or $(\sigma[k]=y \wedge \sigma[k+1]=x)$., then the schedule is fully connected.
- Hence, the main work is to construct σ

Color scheduling (4)

if $K = 4p + 1$, $p > 0$ **then**

$\forall r \in [1, p]$, $\beta(K, r, 2p + 1 - r)$ is constructed.

else if $K = 4p + 3$, $p > 0$ **then**

$\forall r \in [1, p]$, $\beta(K, r, 2p + 1 - r)$ is constructed.

$\alpha(K, 2p + 1)$ is constructed.

endif

σ is a concatenate of all the constructed sequences.

Example of σ construction

$\alpha(K,d) \equiv 0,\ d,\ 2d,\ \ldots,\ (K-1)d$

$\beta(K,d_1,d_2) \equiv 0,d_1,d_1+d_2,(d_1+d_2)+d_1,2(d_1+d_2),\ 2(d_1+d_2)+d_1,$
$\ldots,(K-1)(d_1+d_2),(K-1)(d_1+d_2)+d_1$

$K=9$; $\sigma = \beta(9,1,4) \circ \beta(9,2,3)$, where

$\beta(9,1,4)$=0,1,5,6,1,2,6,7,2,3,7,8,3,4,8,0,4,5

$\beta(9,2,3)$=0,2,5,7,1,3,6,8,2,4,7,0,3,5,8,1,4,6

$K=11$; σ=$\beta(11,1,4) \circ \beta(11,2,3) \circ \alpha(11,5)$, where

$\beta(11,1,4)$=0,1,5,6,10,0,4,5,9,10,3,4,8,9,2,3,7,8,1,2,6,7

$\beta(11,2,3)$=0,2,5,7,10,1,4,6,9,0,3,5,8,10,2,4,7,9,1,3,6,8

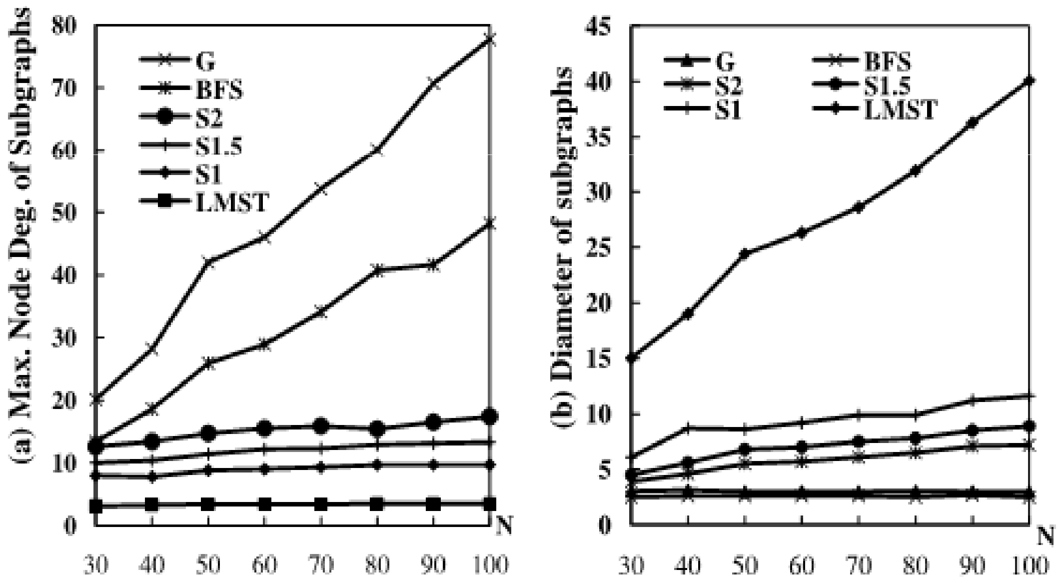
$\alpha(11,5)$ =0,5,10,4,9,3,8,2,7,1,6

Simulation(1)

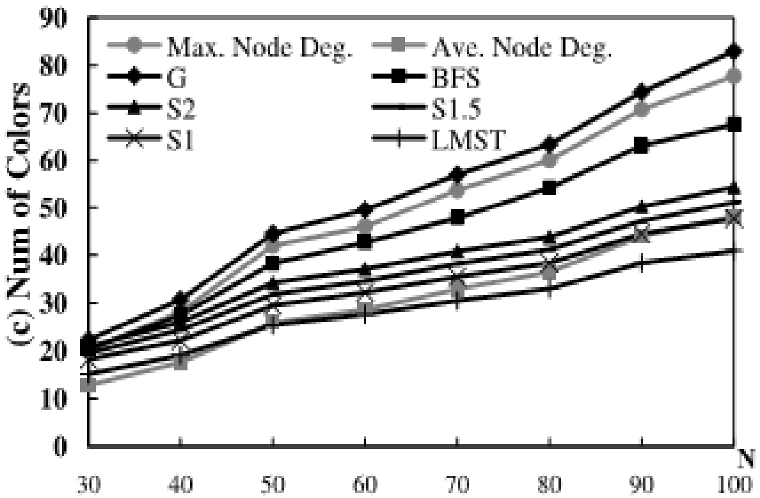
- Consider 4 levels of connectivity

Connectivity	Subgraph S	Denotation
Full Reserved	G	**G**
Strong Connectivity	LMST	**LMST**
Strong Connectivity w.h.p.	$\mathcal{G}_{n,d\log n,R}$, $d = 1, 1.5, 2$	**S1, S1.5, S2**
Connectivity to Sink	BFS rooted at sink	**BFS**

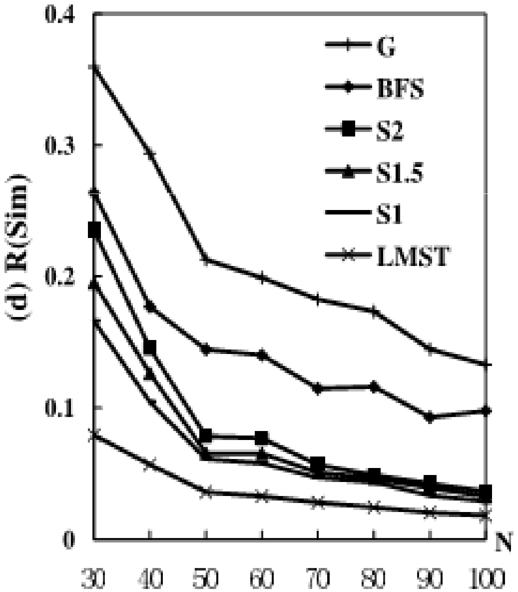
Simulation(2)

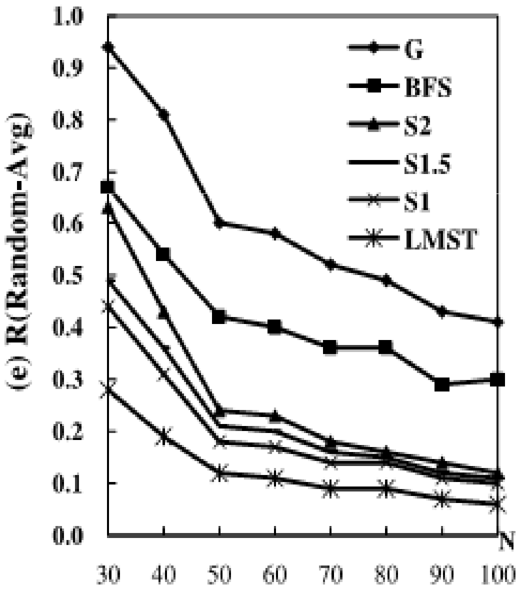


Simulation(3)



Simulation(4)





Conclusion

- According to the simulation, if this coloring is used by collision avoidance scheme
  - much less colors than traditional colorings
  - much less packet latency than traditional colorings