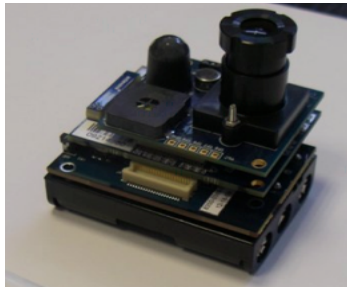
Designing and implementing a criticality-based duty-cycled MAC for low-latency mission-critical surveillance applications

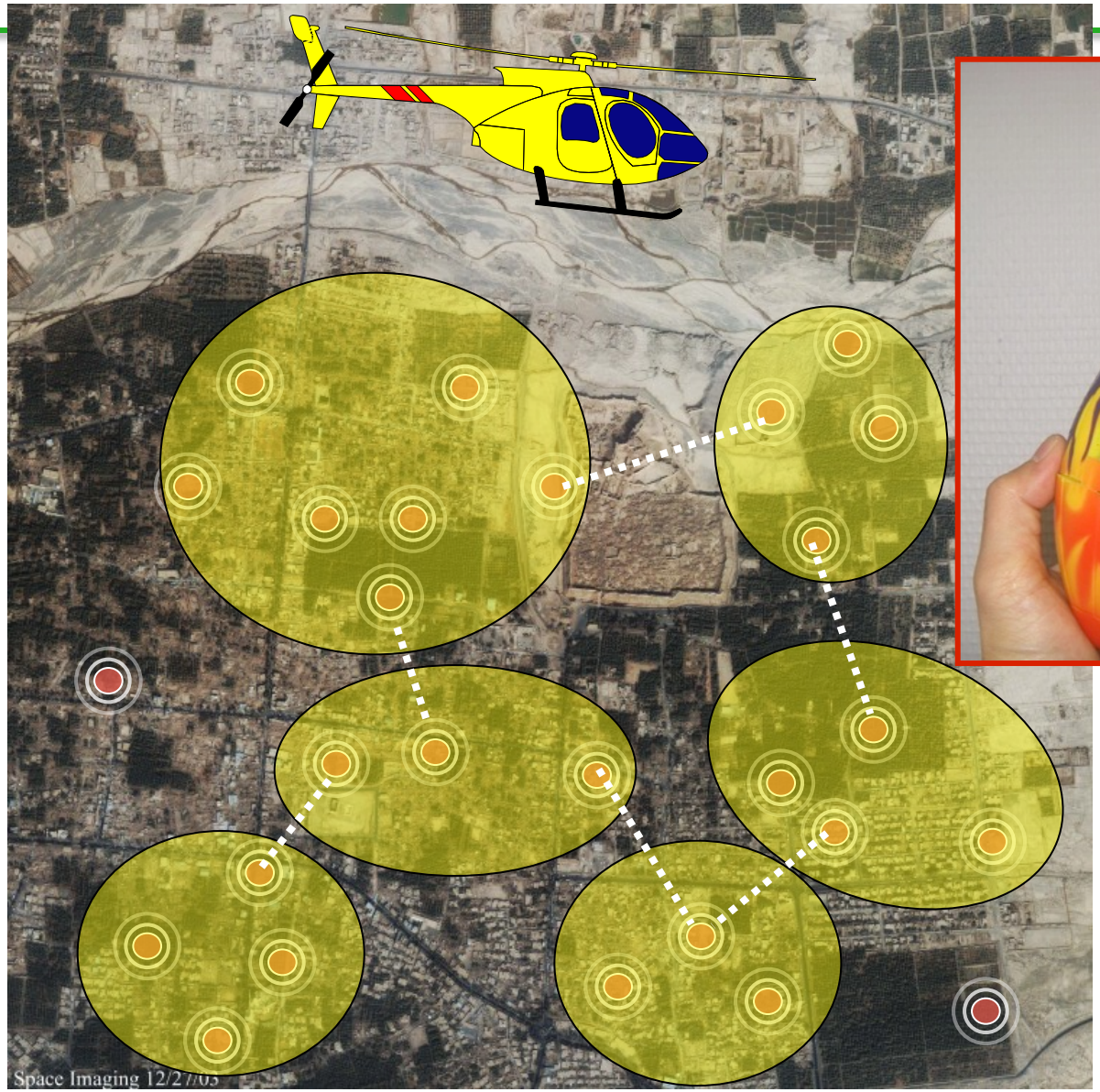
E. Muhammad and C. Pham

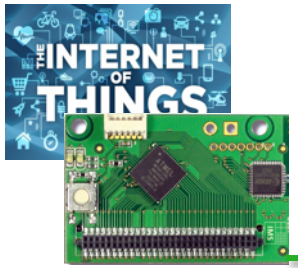
ISCC 2014
24th June, Madeira, Portugal

Prof. Congduc Pham
http://www.univ-pau.fr/~cpham
Université de Pau, France

Search&Rescue, security



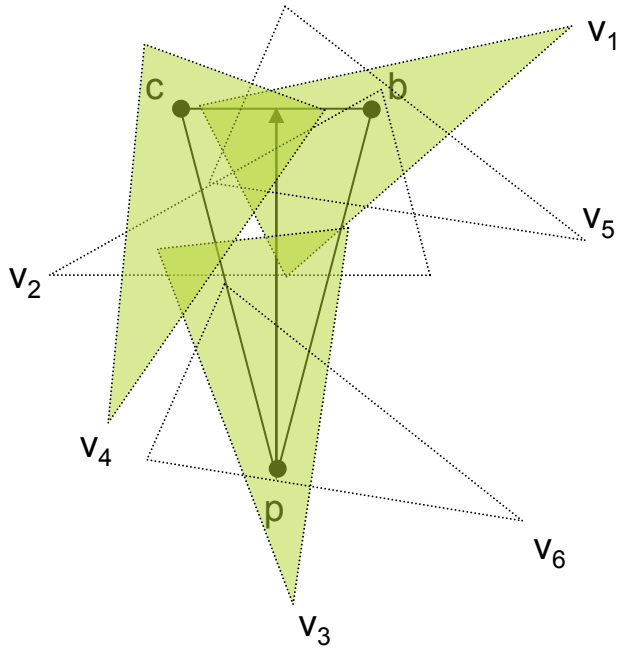


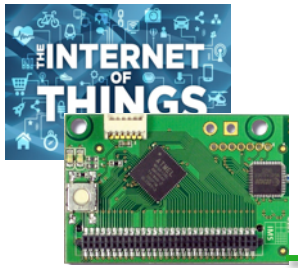


Node's cover set

Co(V)= {
{V },
{V_1, V_3, V_4},
{V_2, V_3, V_4},
{V_3, V_4, V_5},
{V_1, V_4, V_6},
{V_2, V_4, V_6},
{V_4, V_5, V_6}
}

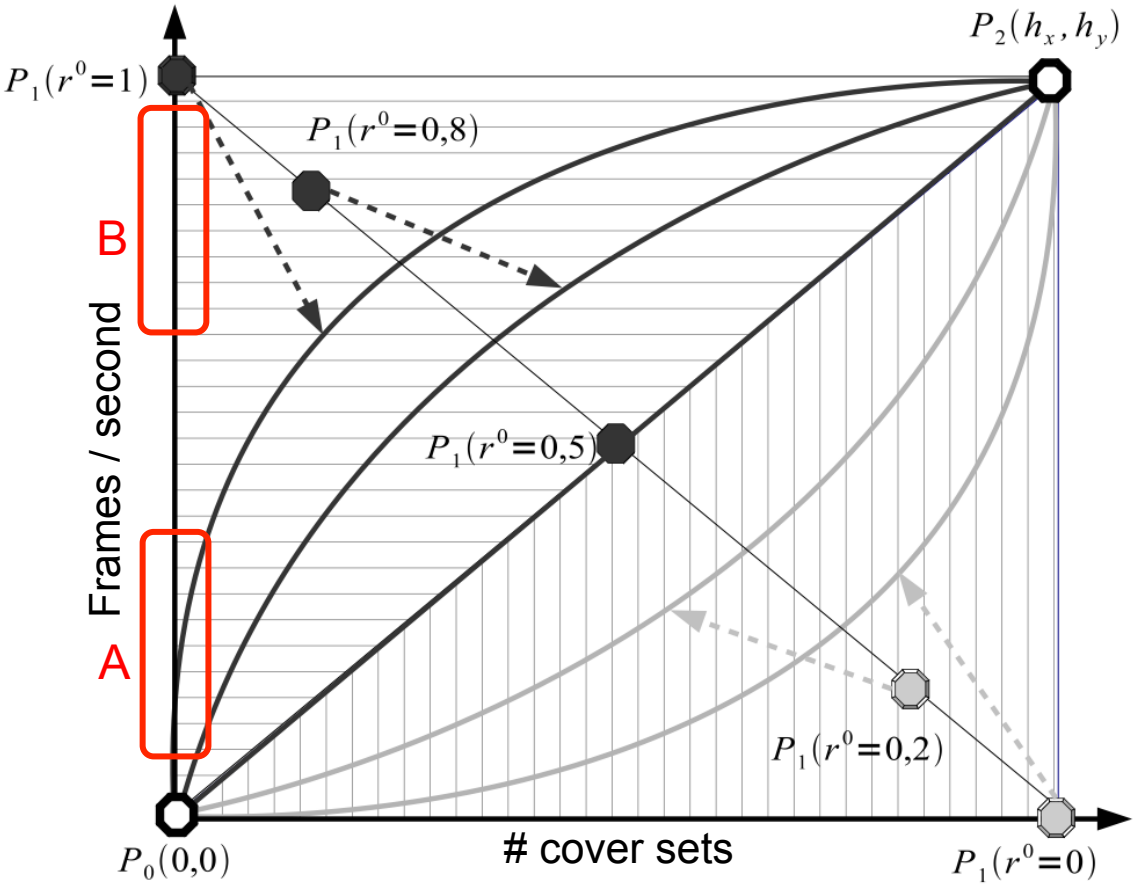
|Co(V)| = 7





Criticality model

- r^0 can vary in [0,1]
- Behavior functions (BV) defines the capture speed according to r^0
- $r^0 < 0.5$
  - Concave shape BV
- $r^0 > 0.5$
  - Convex shape BV
- We propose to use Bezier curves to model BV functions



Criticality model

- r^0 can vary in [0,1]
- Behavior functions (BV) defines the capture speed according to r^0
- $r^0 < 0.5$
  - Concave shape BV

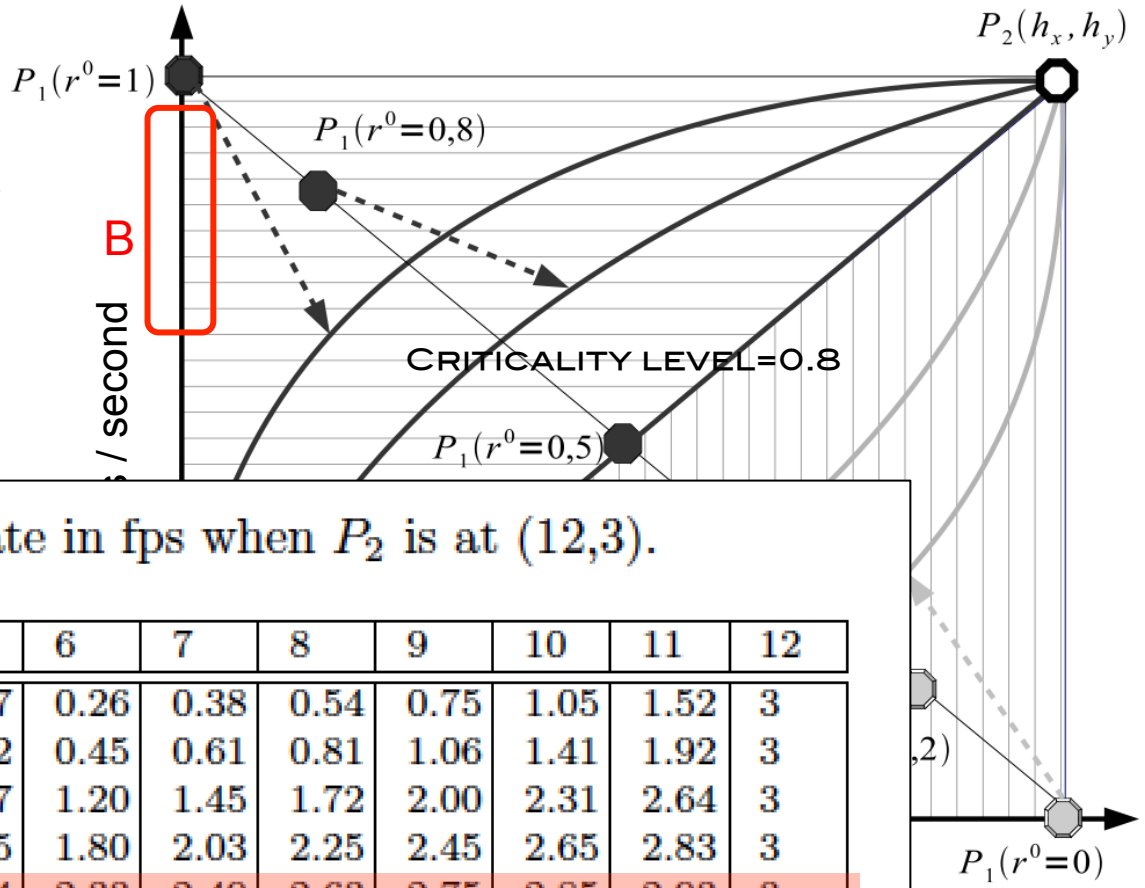
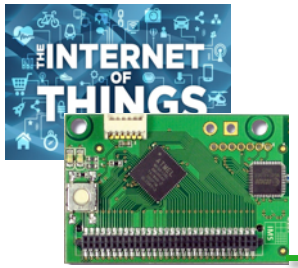


Table 1: Capture rate in fps when P_2 is at (12,3).

r^0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.01	0.02	0.05	0.10	0.17	0.26	0.38	0.54	0.75	1.05	1.52	3
.1	0.03	0.08	0.14	0.22	0.32	0.45	0.61	0.81	1.06	1.41	1.92	3
.4	0.17	0.35	0.55	0.75	0.97	1.20	1.45	1.72	2.00	2.31	2.64	3
.6	0.36	0.69	1.00	1.28	1.55	1.80	2.03	2.25	2.45	2.65	2.83	3
.8	0.75	1.24	1.61	1.90	2.14	2.33	2.49	2.63	2.75	2.85	2.93	3
1	1.48	1.95	2.25	2.46	2.62	2.74	2.83	2.90	2.95	2.98	2.99	3



Criticality model

- R^0 can vary in [0
- Behavior funct
  (BV) defines the
  capture speed
  according to R^0
- $R^0 < 0.5$
  - Concave shape

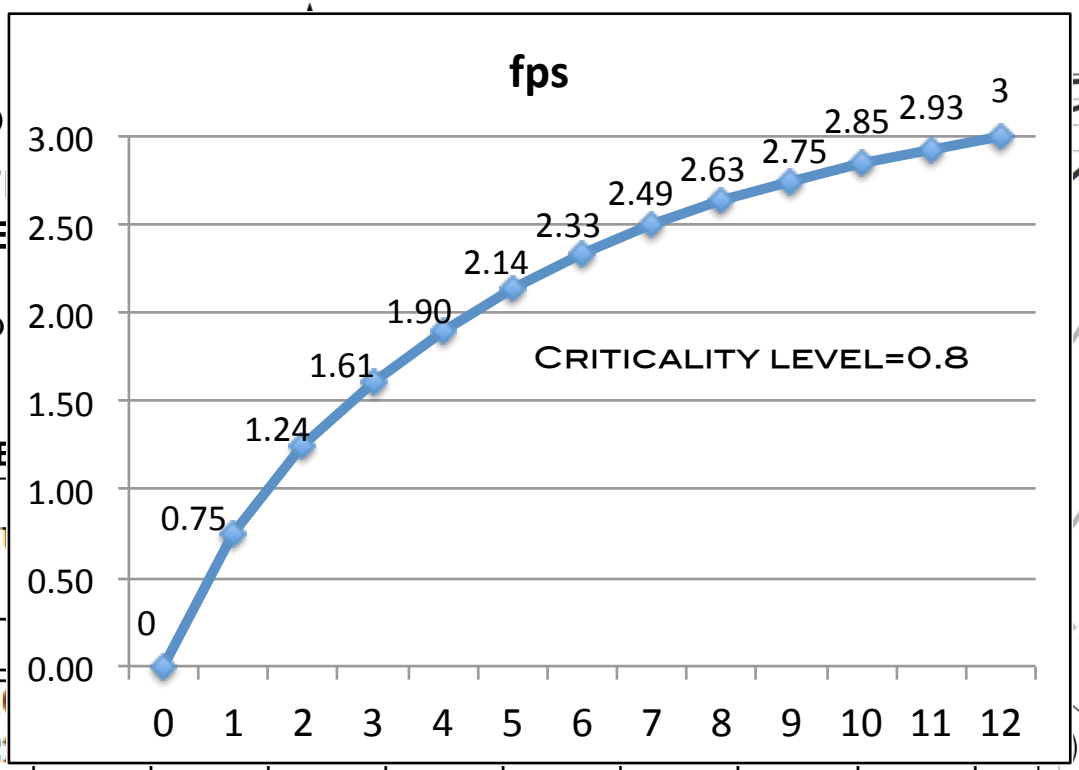
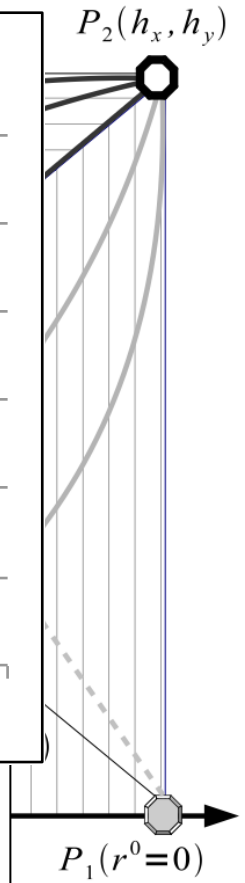
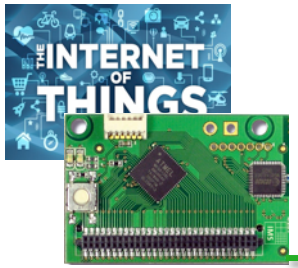


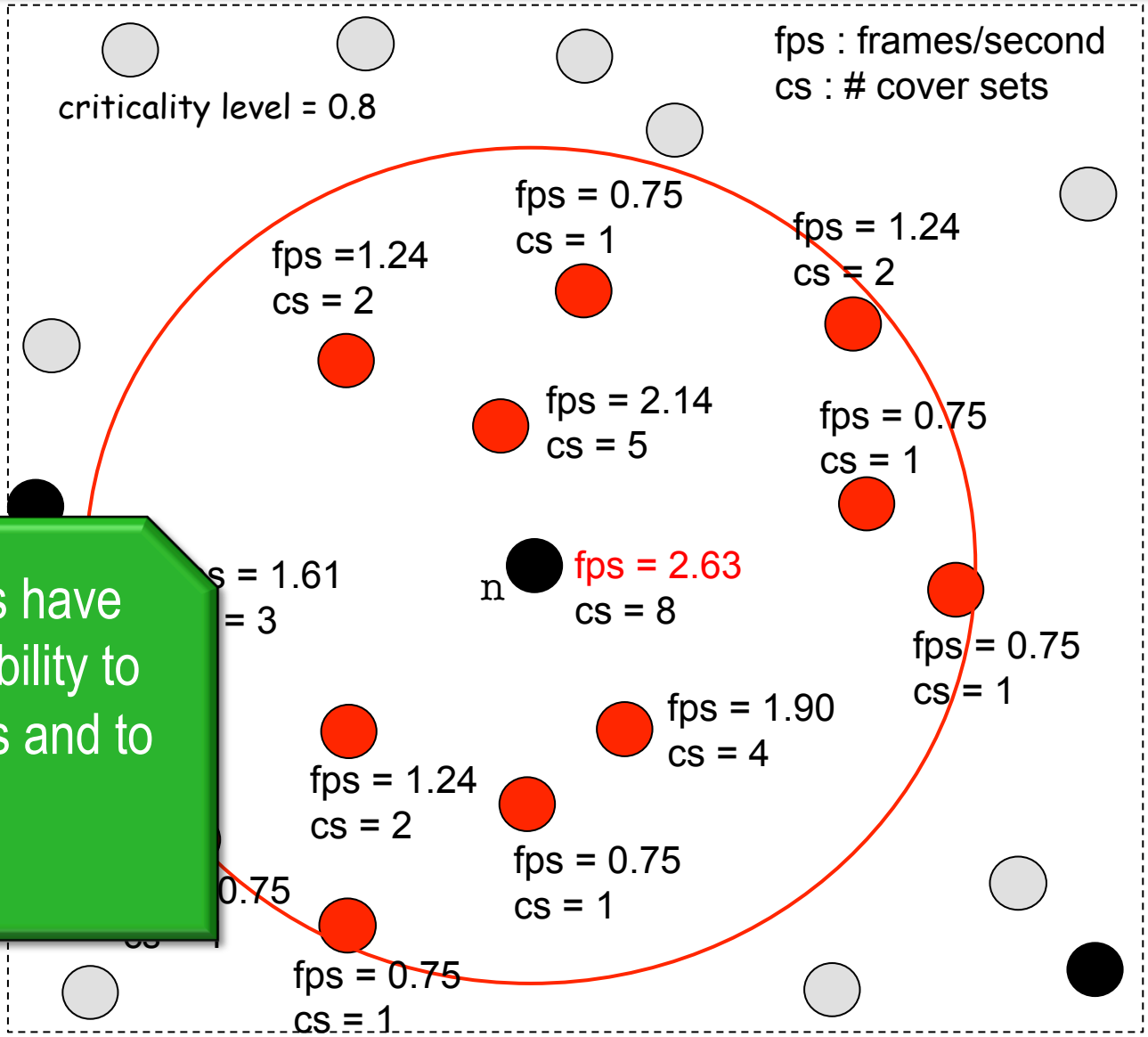
Table 1: Capt

r^0	1	2	3	4								
0	0.01	0.02	0.05	0.1								
.1	0.03	0.08	0.14	0.2								
.4	0.17	0.35	0.55	0.75	0.97	1.20	1.45	1.72	2.00	2.31	2.64	3
.6	0.36	0.69	1.00	1.28	1.55	1.80	2.03	2.25	2.45	2.65	2.83	3
.8	0.75	1.24	1.61	1.90	2.14	2.33	2.49	2.63	2.75	2.85	2.93	3
1	1.48	1.95	2.25	2.46	2.62	2.74	2.83	2.90	2.95	2.98	2.99	3

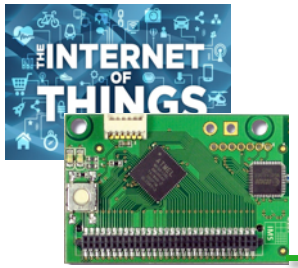




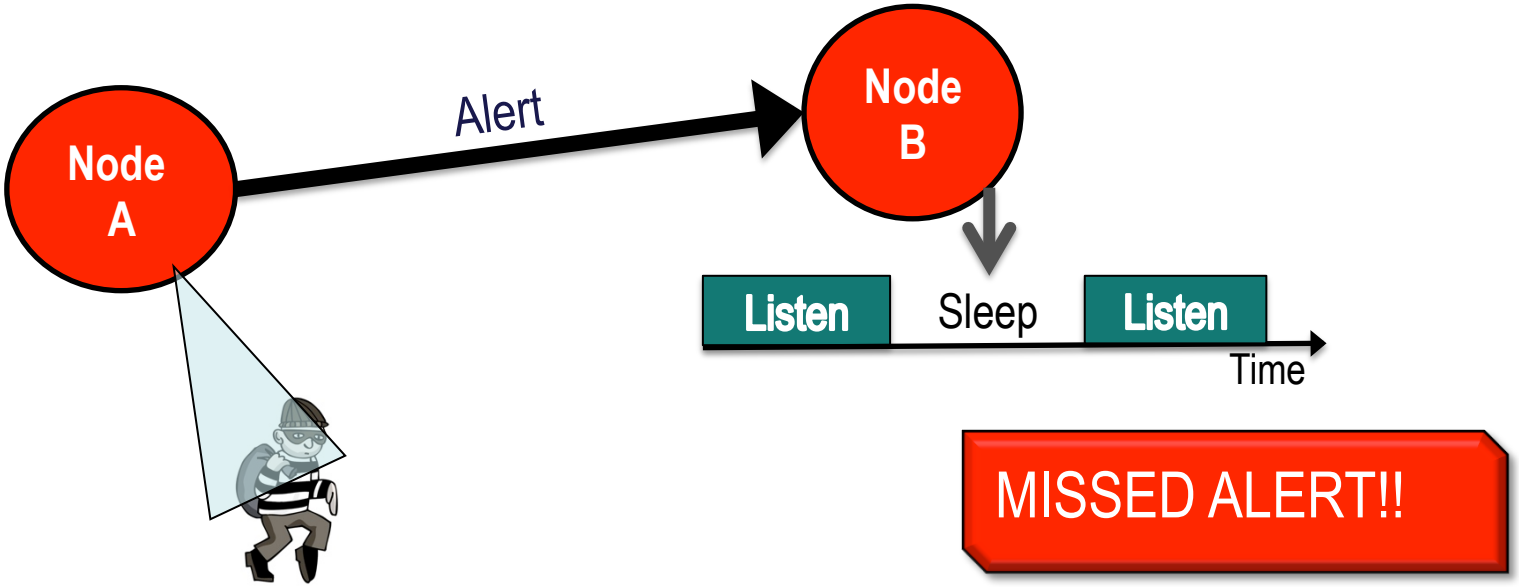
Criticality-based scheduling



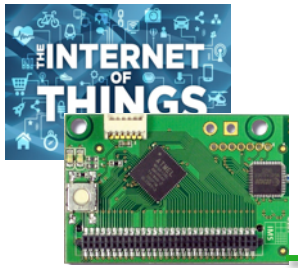
Sentry nodes have higher probability to detect events and to send alerts



Duty-Cycling Issues

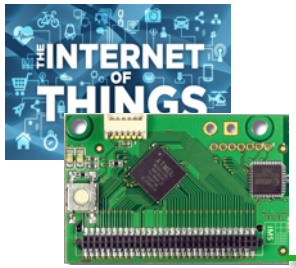


- Sensor's activity usually has duty-cycle behavior to save energy
- Radio & MAC layer activities represent a large part of energy consumption

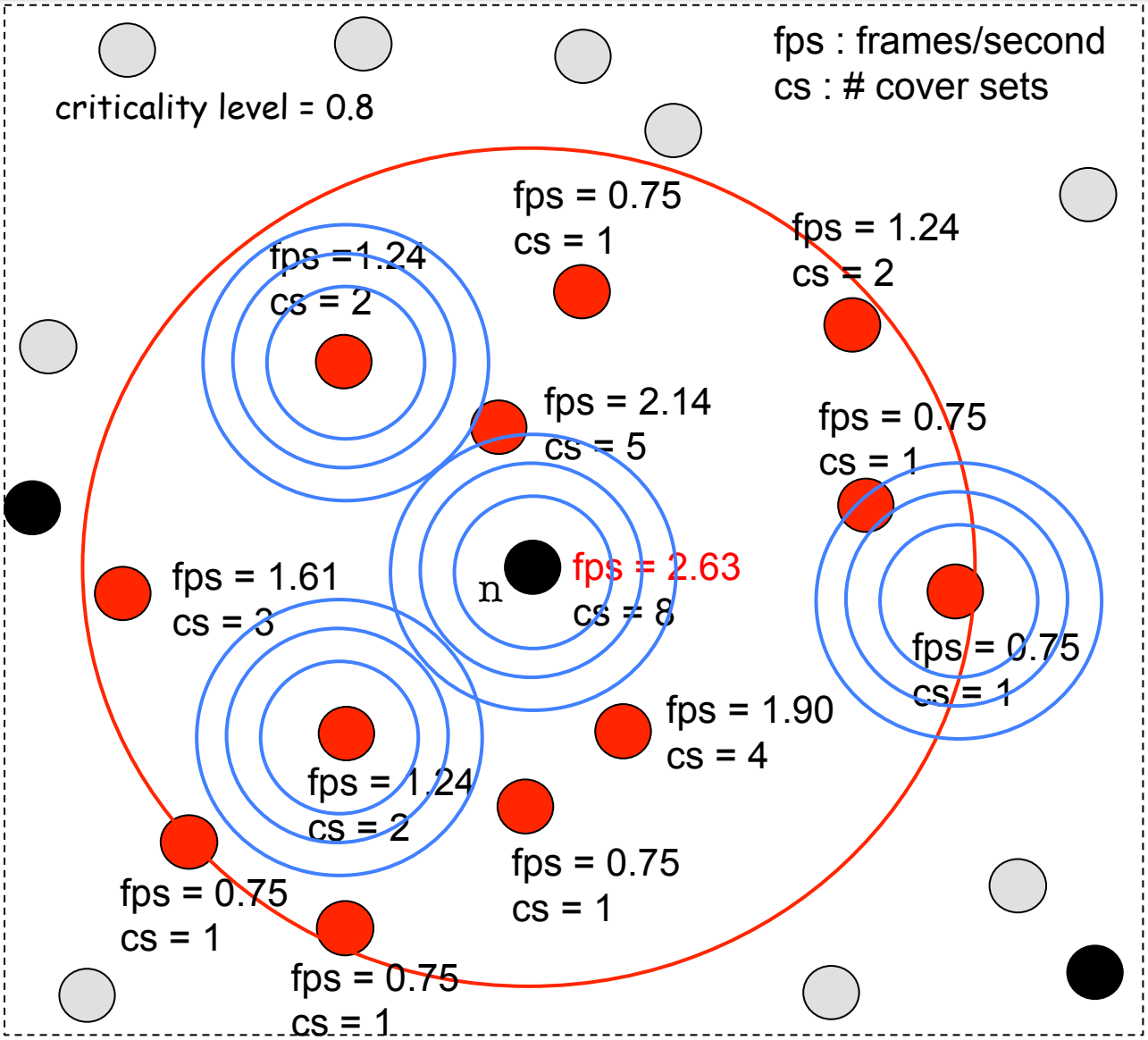


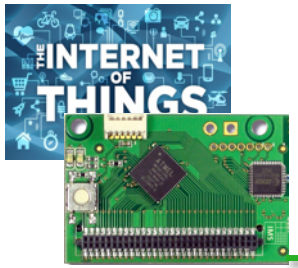
Adaptive duty-cycled MAC protocol

- Static duty-cycle MAC can not adapt to application's needs nor to surveillance's criticality
- Synchronized duty-cycle MAC approaches do not scale well
- Adaptive criticality-based MAC
  - Adapts the active period of follower nodes according to a sentry's activity
  - Take into account # of cover-set to preserve coverage constraints

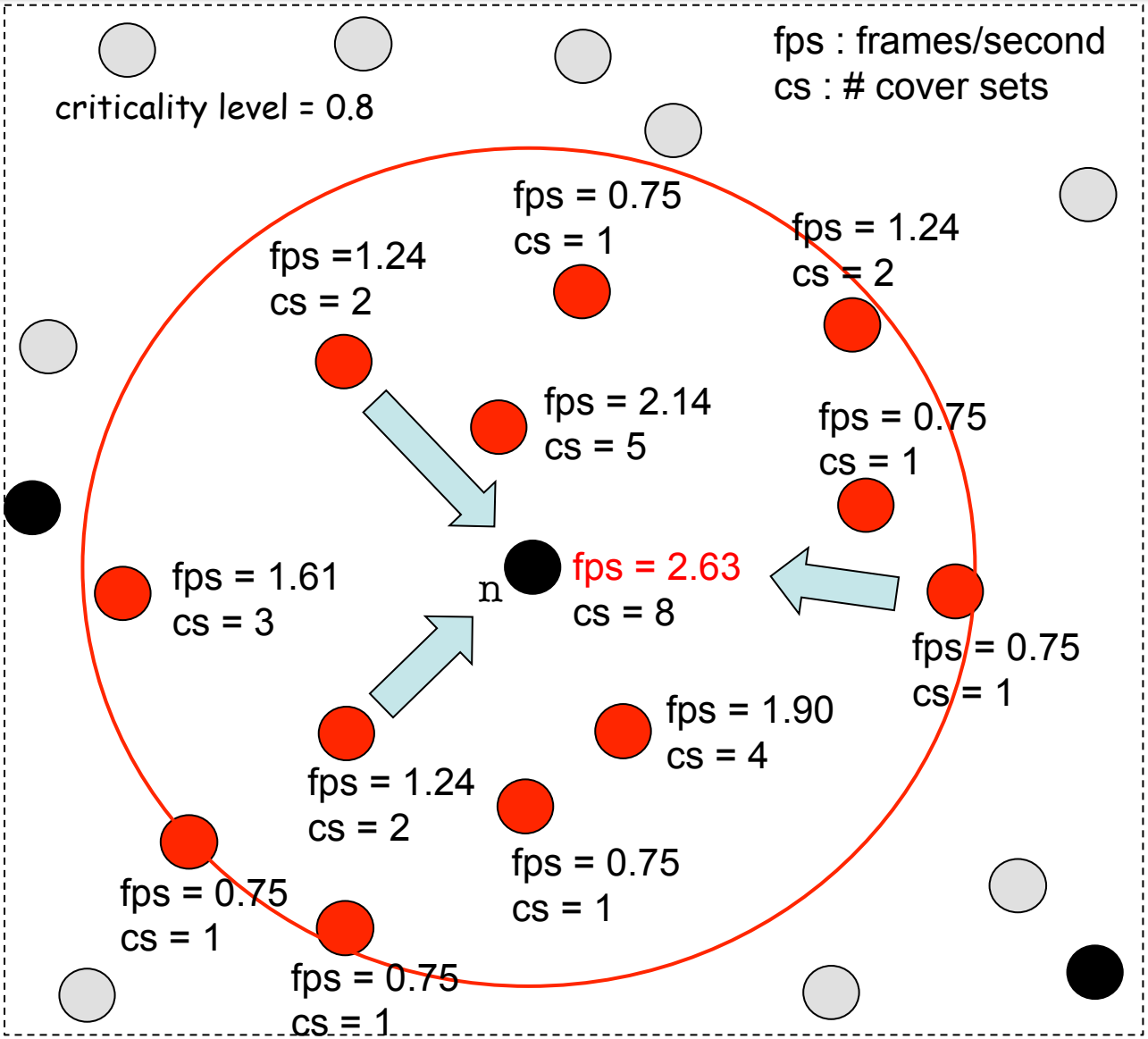


INFO BROADCAST

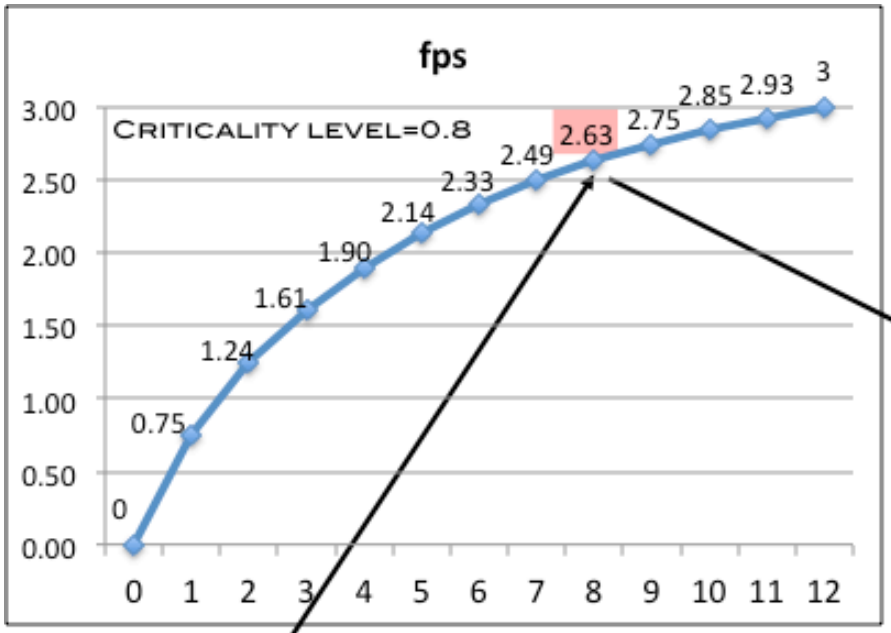




Follower-Sentry association

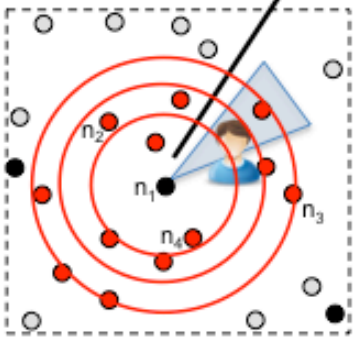


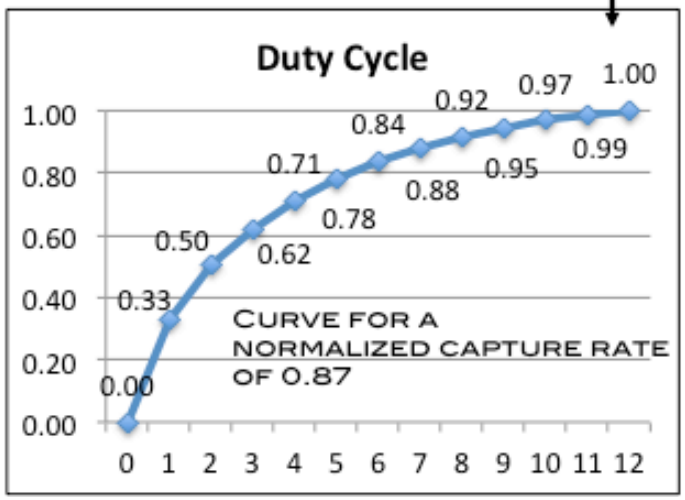
Duty-cycle computation at follower nodes



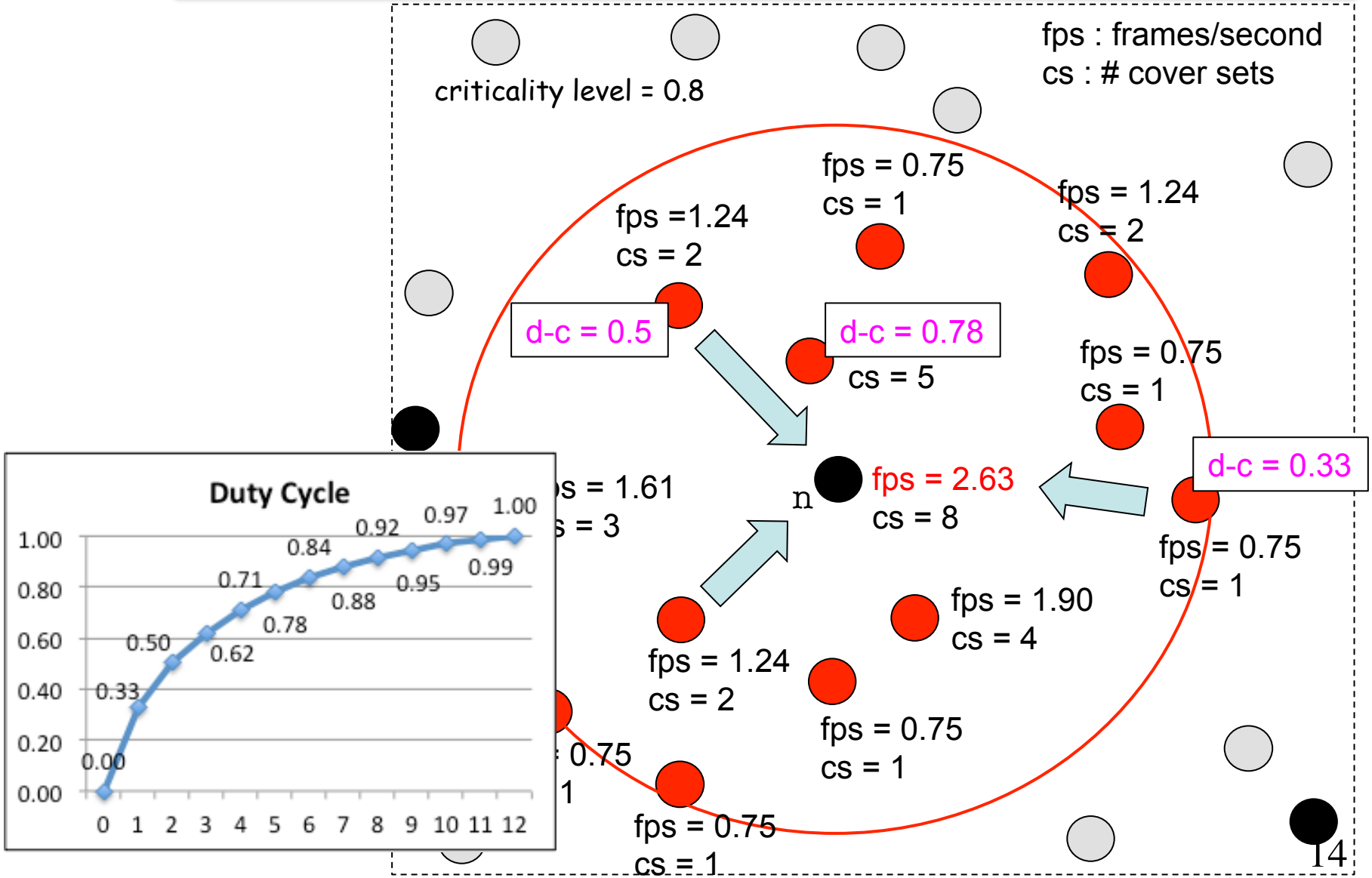
2.63/3.00=0.87

can be viewed as a new criticality level for follower nodes

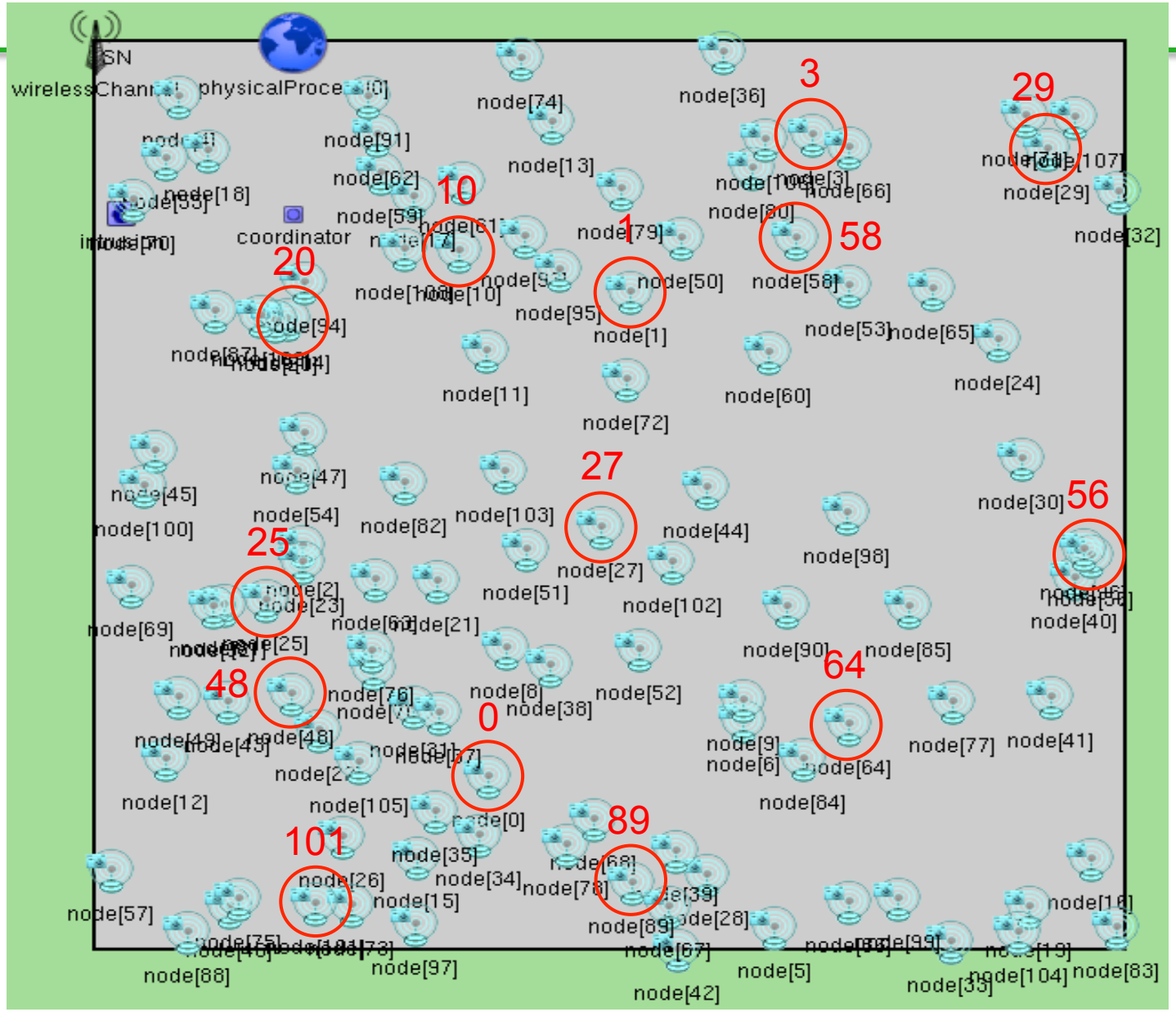




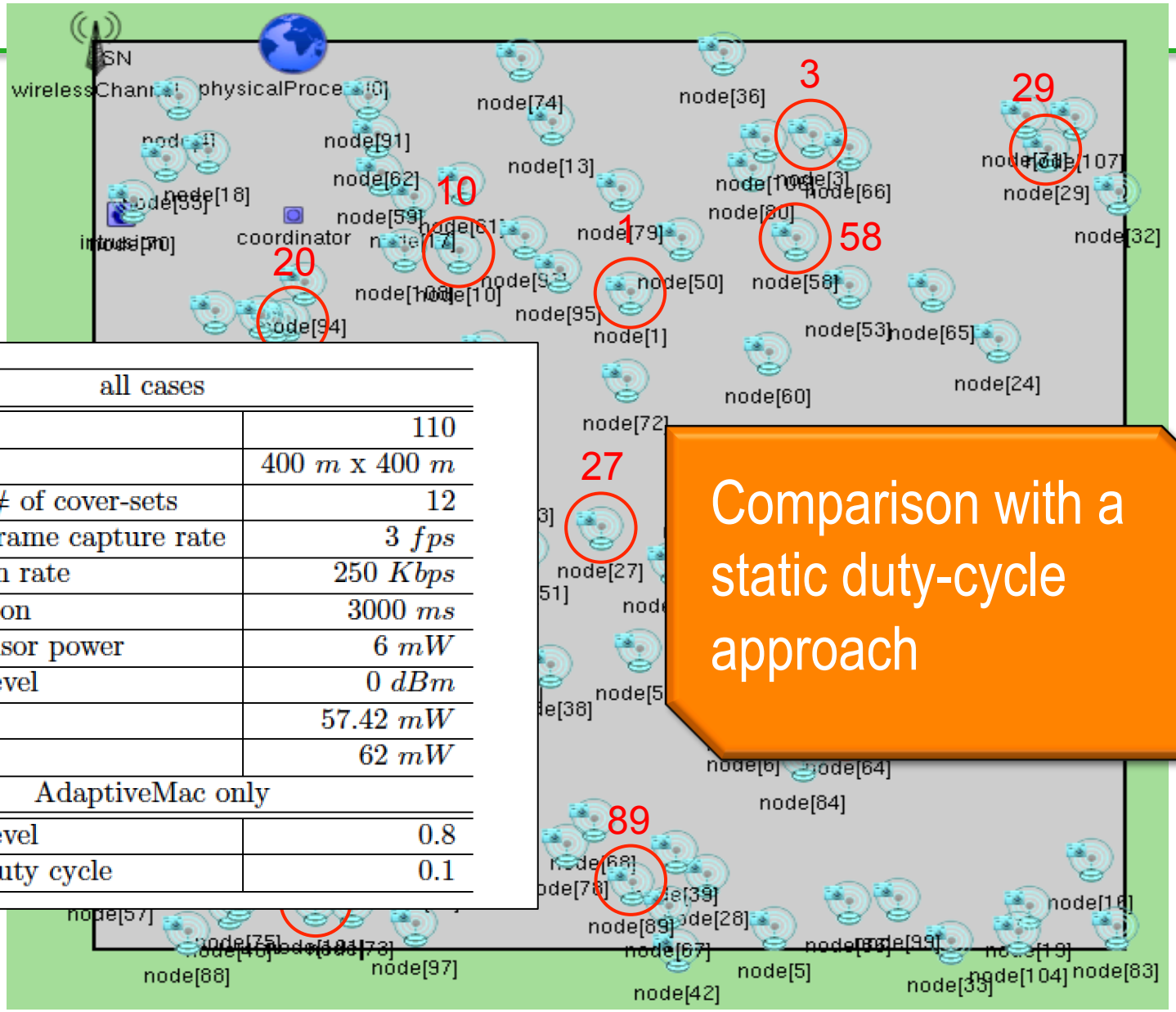
DUTY-CYCLE OF FOLLOWERS



Simulation study

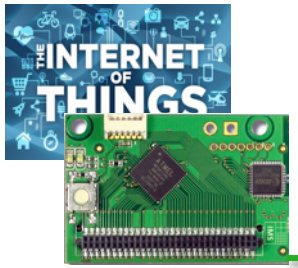


Simulation study

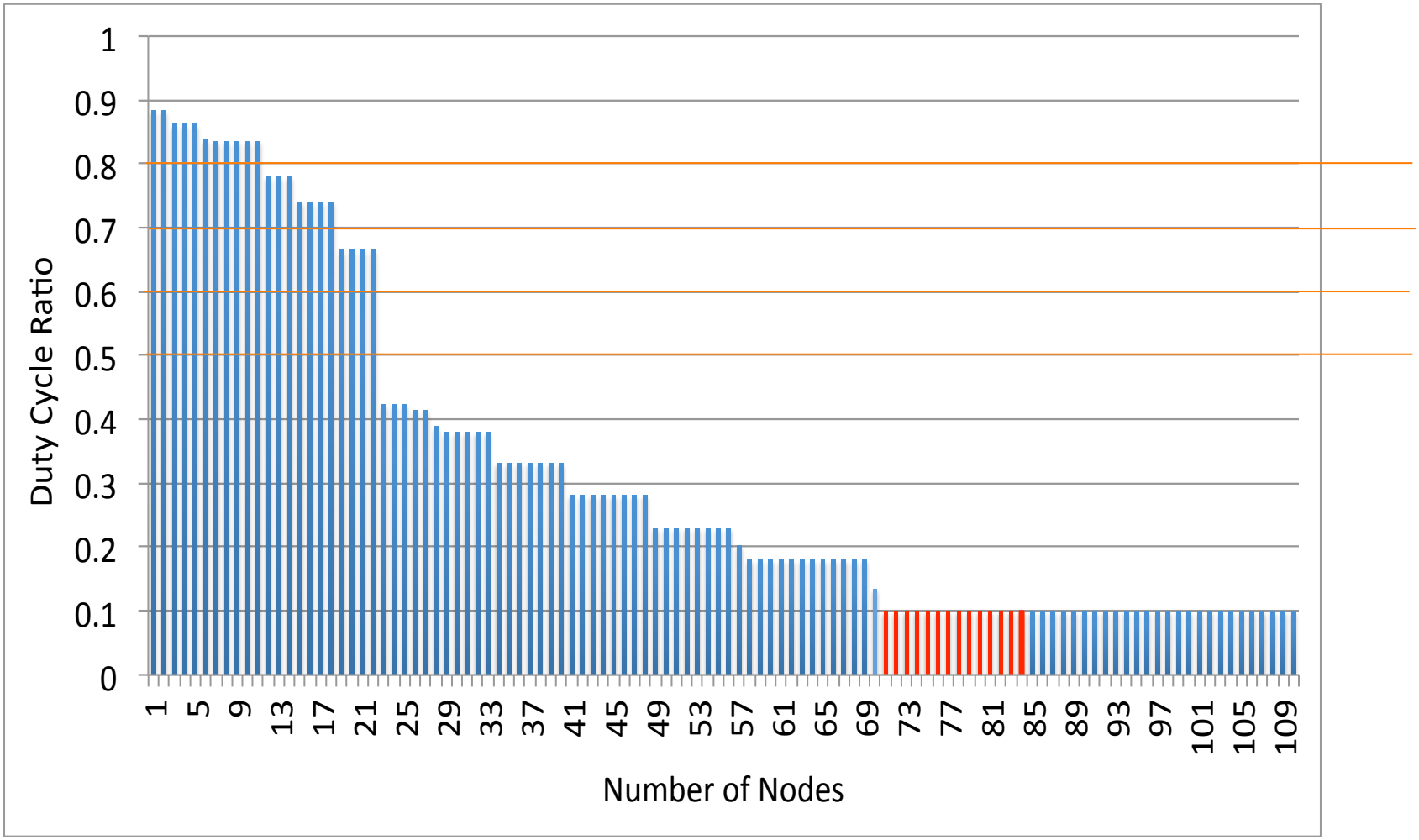


all cases	
# of nodes	110
field size	400 m x 400 m
maximum # of cover-sets	12
maximum frame capture rate	3 fps
transmission rate	250 $Kbps$
cycle duration	3000 ms
baseline sensor power	6 mW
Tx power level	0 dBm
Tx power	57.42 mW
Rx power	62 mW
AdaptiveMac only	
criticality level	0.8
minimum duty cycle	0.1

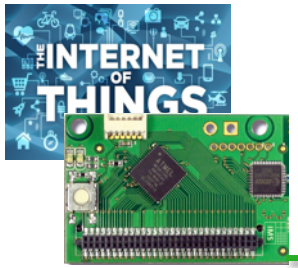
Comparison with a static duty-cycle approach



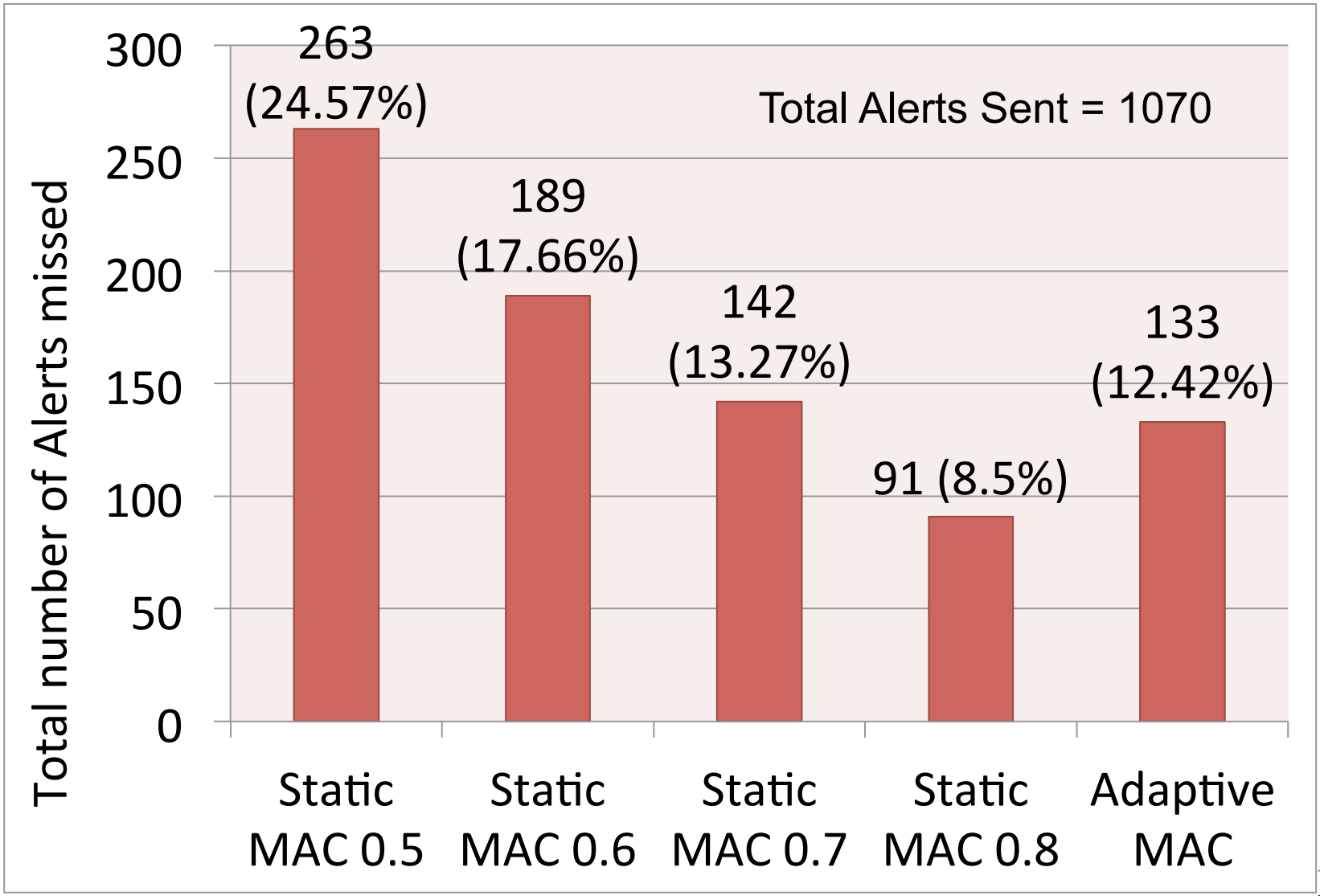
Duty-cycle length

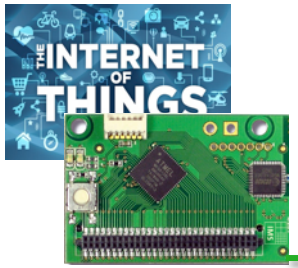


17

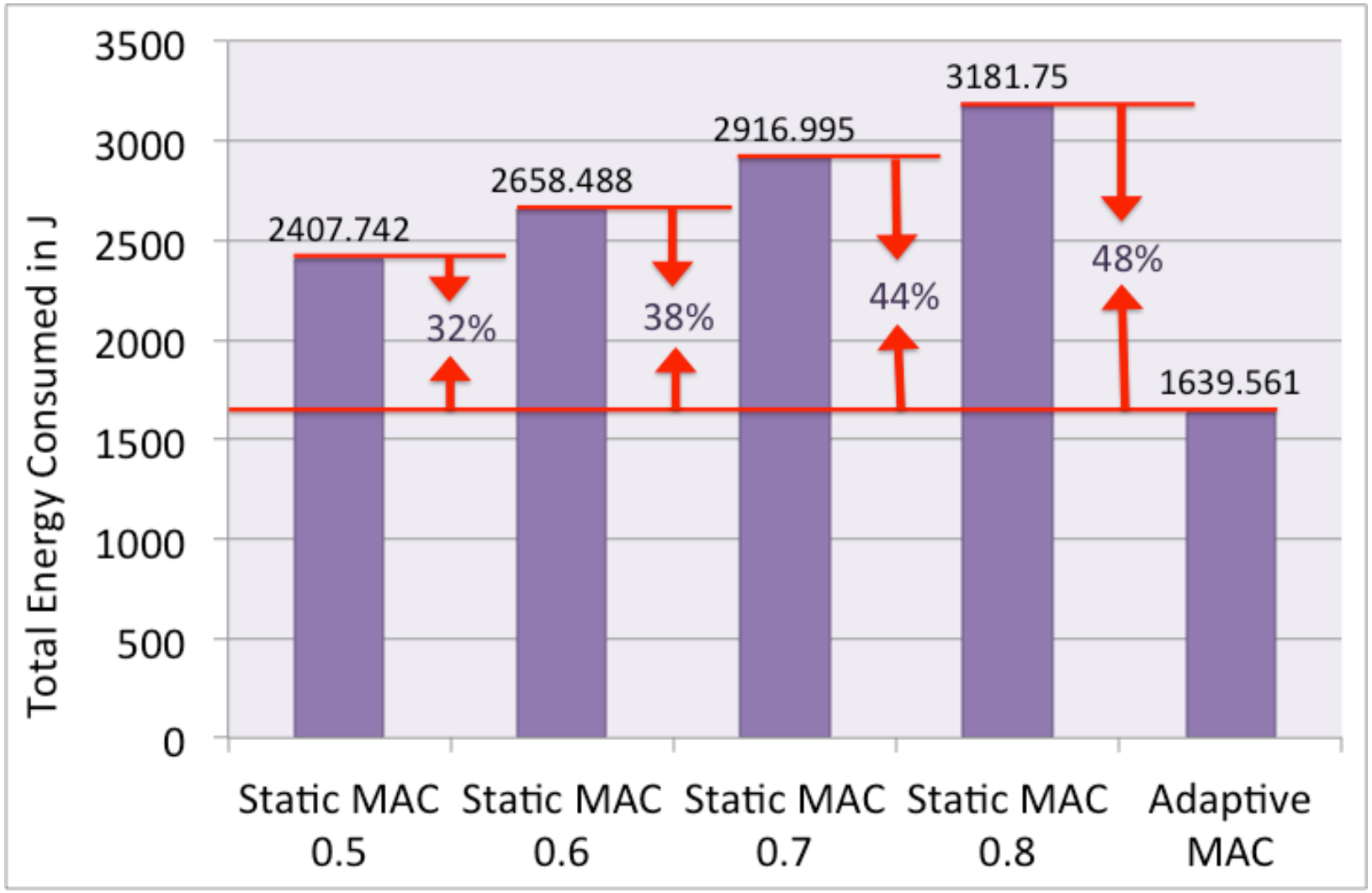


OF MISSED ALERTS

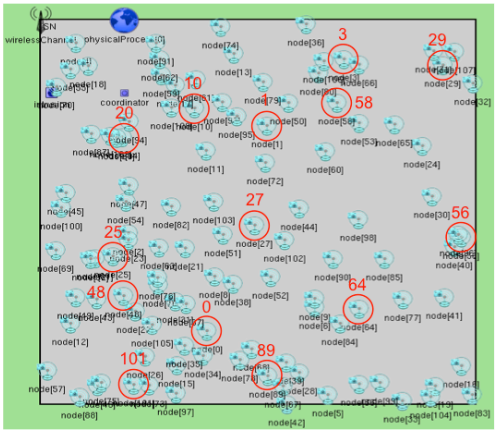




Global energy consumption

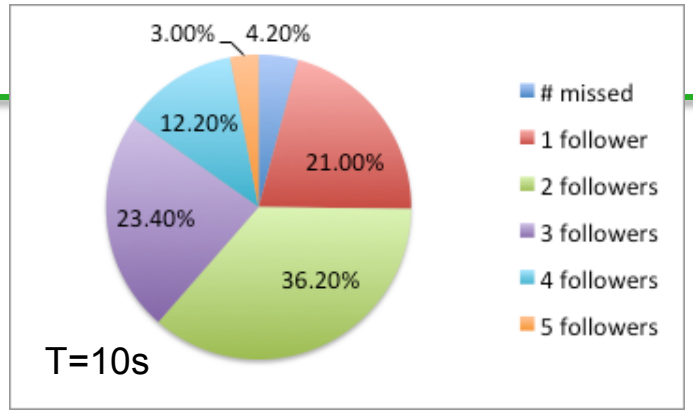


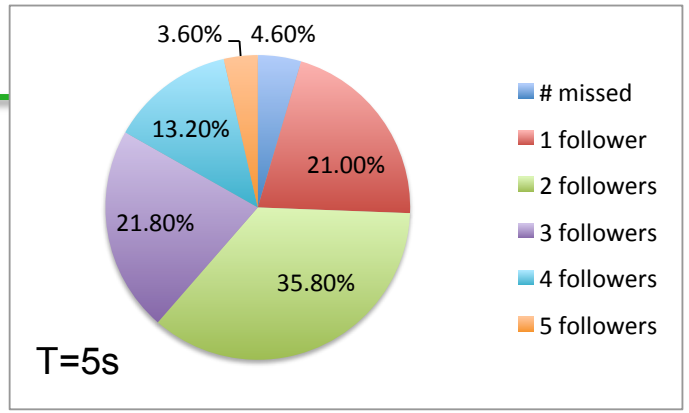
Impact of cycle length

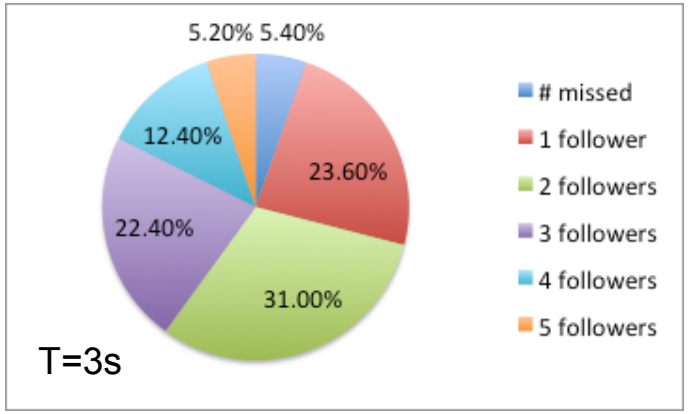


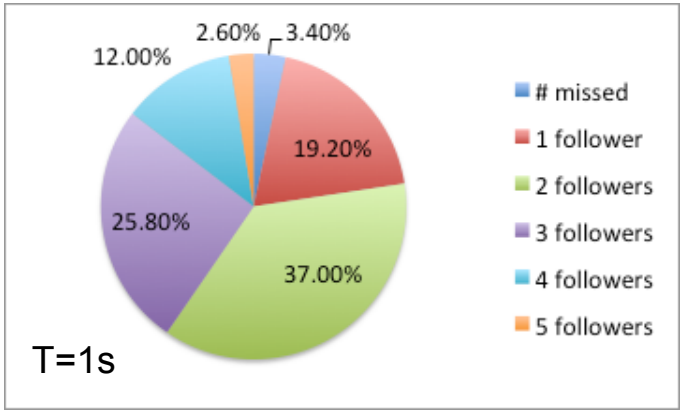
Sentry node 10

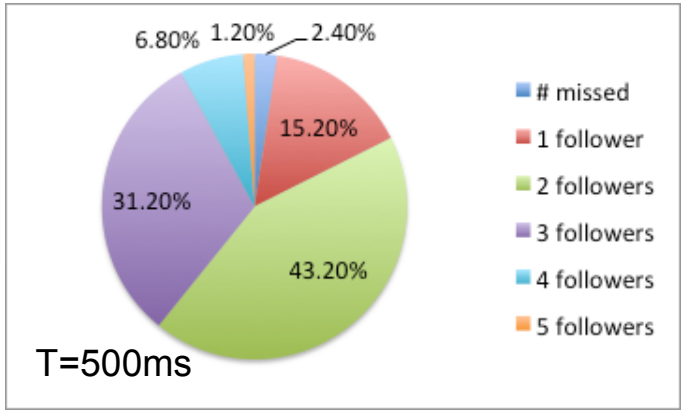
All 5 neighbors are followers

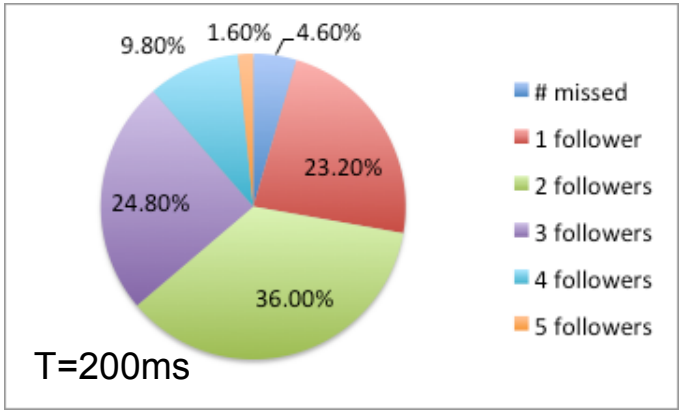


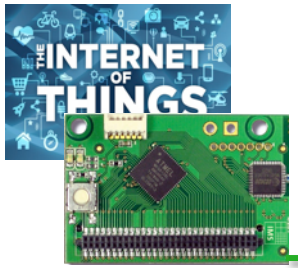






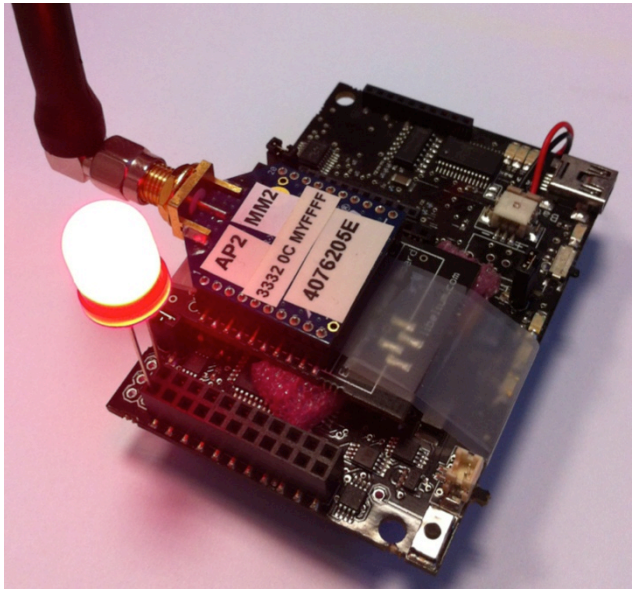




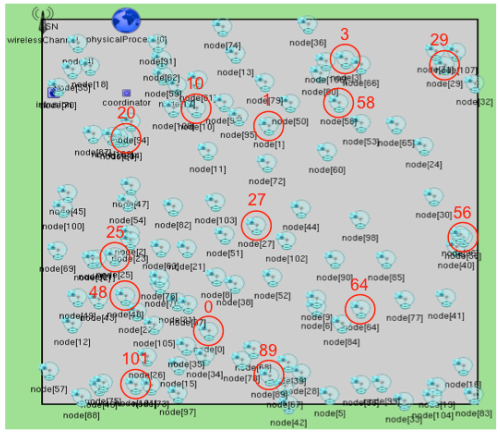


Implementation

- Libelium Waspmote with Xbee module
- Easy to completly power off the radio module
- Sentry is emulated with a Linux machine

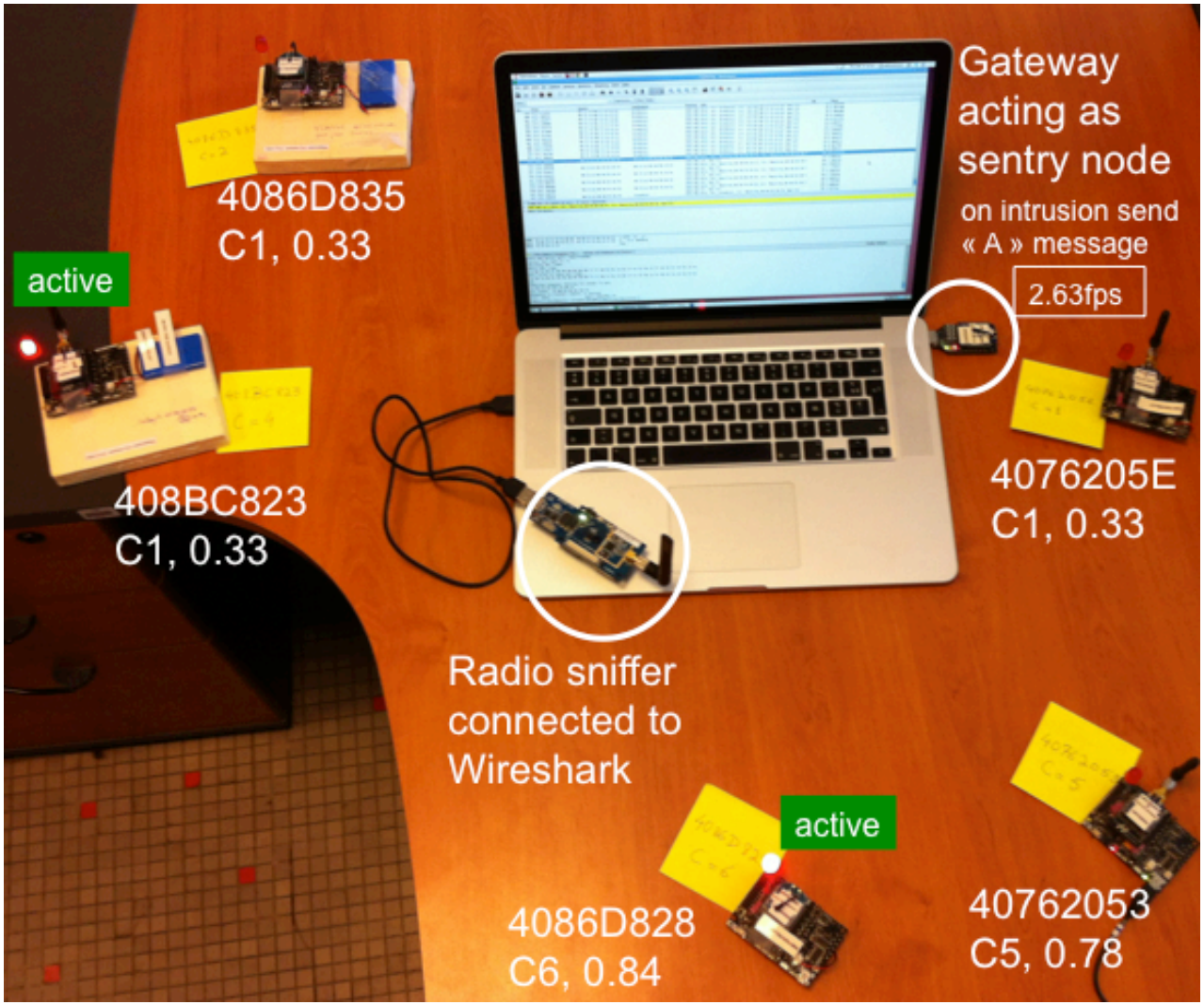


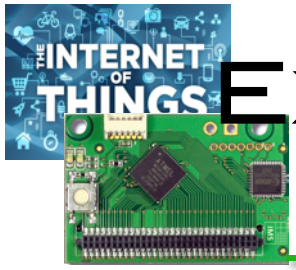
Sentry node 10 configuration



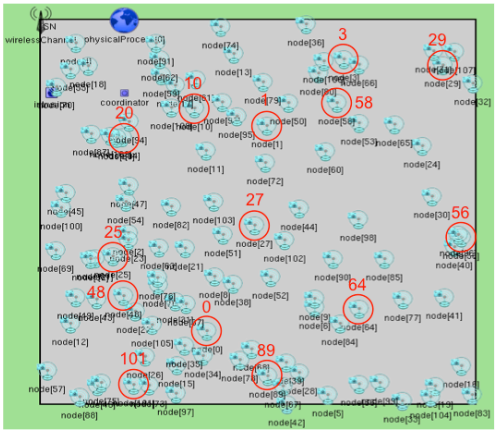
Sentry node 10

All 5 neighbors are followers

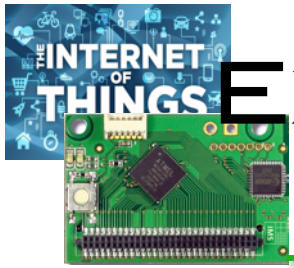




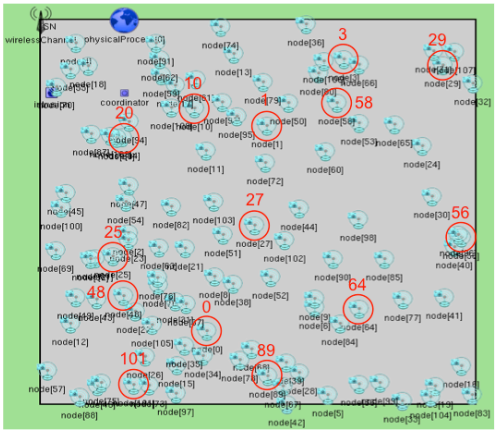
Experiments with Sentry Node 10 Data Trace



```
10 SN.node[10].Application Sending [alert]
18 SN.node[10].Application Sending [alert]
23 SN.node[10].Application Sending [alert]
29 SN.node[10].Application Sending [alert]
35 SN.node[10].Application Sending [alert]
40 SN.node[10].Application Sending [alert]
47 SN.node[10].Application Sending [alert]
54 SN.node[10].Application Sending [alert]
62 SN.node[10].Application Sending [alert]
69 SN.node[10].Application Sending [alert]
79 SN.node[10].Application Sending [alert]
86 SN.node[10].Application Sending [alert]
93 SN.node[10].Application Sending [alert]
101 SN.node[10].Application Sending [alert]
107 SN.node[10].Application Sending [alert]
113 SN.node[10].Application Sending [alert]
...
```



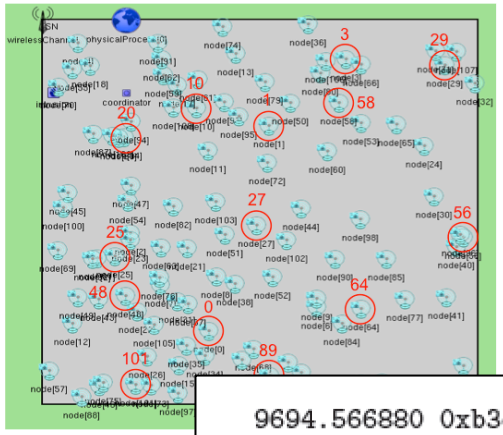
Experiments with Sentry Node 10 Data Trace



```
10 SN.node[10].Application_Sending [alert]
18 SN.node[10].Application
23 SN.node[10].Application
29 SN.node[10].Application
35 SN.node[10].Application
40 SN.node[10].Application
47 SN.node[10].Application
54 SN.node[10].Application
62 SN.node[10].Application
69 SN.node[10].Application
79 SN.node[10].Application
86 SN.node[10].Application
93 SN.node[10].Application
101 SN.node[10].Applicatio
107 SN.node[10].Applicatio
113 SN.node[10].Applicatio
...
```

```
Start time is
Mon Apr 21 15:01:07 2014
10
sleep for 10
Mon Apr 21 15:01:17 2014 : time 10 Intrusion 1 : sending alert
18
sleep for 8
Mon Apr 21 15:01:25 2014 : time 18 Intrusion 2 : sending alert
23
sleep for 5
Mon Apr 21 15:01:30 2014 : time 23 Intrusion 3 : sending alert
29
sleep for 6
Mon Apr 21 15:01:36 2014 : time 29 Intrusion 4 : sending alert
35
sleep for 6
Mon Apr 21 15:01:42 2014 : time 35 Intrusion 5 : sending alert
40
sleep for 5
Mon Apr 21 15:01:47 2014 : time 40 Intrusion 6 : sending alert
47
sleep for 7
Mon Apr 21 15:01:54 2014 : time 47 Intrusion 7 : sending alert
...
```

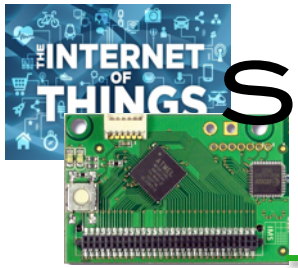
Experiments with Sentry Node 10 Data Trace



```
10 SN.node[10].Application Sending [alert]
18 SN.node[10].Application
23 SN.node[10].Application
29 SN.node[10].Application
35 SN.node[10].Application
40 SN.node[10].Application
47 SN.node[10].Application
54 SN.node[10].Application
62 SN.node[10].Application
69 SN.node[10].Application
79 SN.node[10].Application
86 SN.node[10].Application
```

```
Start time is
Mon Apr 21 15:01:07 2014
10
sleep for 10
Mon Apr 21 15:01:17 2014 : time 10 Intrusion 1 : sending alert
18
sleep for 8
Mon Apr 21 15:01:25 2014 : time 18 Intrusion 2 : sending alert
```

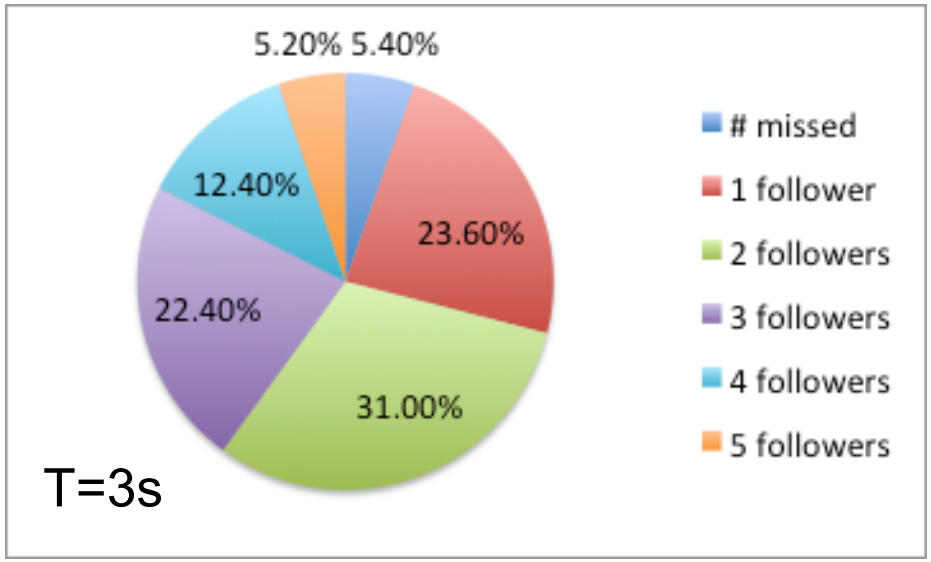
```
9694.566880 0xb3e8                  Dst: Broadcast, Src: 0xb3e8
9694.736000 00:13:a2:00:40:76:20:53 Dst: Broadcast, Src: Maxstrea_00:40:76:20:53   : sending alert
9694.862784 00:13:a2:00:40:86:d8:35 Dst: Broadcast, Src: Maxstrea_00:40:86:d8:35
9702.221312 0xb3e8                  Dst: Broadcast, Src: 0xb3e8
9702.387296 00:13:a2:00:40:86:d8:35 Dst: Broadcast, Src: Maxstrea_00:40:86:d8:35   : sending alert
9702.388820 00:13:a2:00:40:86:d8:28 Dst: Broadcast, Src: Maxstrea_00:40:86:d8:28
9702.390560 00:13:a2:00:40:76:20:5e Dst: Broadcast, Src: Maxstrea_00:40:76:20:5e   : sending alert
9702.393216 00:13:a2:00:40:8b:c8:23 Dst: Broadcast, Src: Maxstrea_00:40:8b:c8:23
9707.064864 0xb3e8                  Dst: Broadcast, Src: 0xb3e8
9707.230816 00:13:a2:00:40:8b:c8:23 Dst: Broadcast, Src: Maxstrea_00:40:8b:c8:23   : sending alert
9707.630624 00:13:a2:00:40:86:d8:28 Dst: Broadcast, Src: Maxstrea_00:40:86:d8:28
9713.010560 0xb3e8                  Dst: Broadcast, Src: 0xb3e8
9713.097024 00:13:a2:00:40:76:20:53 Dst: Broadcast, Src: Maxstrea_00:40:76:20:53   : sending alert
9713.099616 00:13:a2:00:40:8b:c8:23 Dst: Broadcast, Src: Maxstrea_00:40:8b:c8:23
9713.176720 00:13:a2:00:40:86:d8:28 Dst: Broadcast, Src: Maxstrea_00:40:86:d8:28
...
9799.369728 0xb3e8                  Dst: Broadcast, Src: 0xb3e8, Bad FCS
9812.351552 0xb3e8                  Dst: Broadcast, Src: 0xb3e8, Bad FCS
...
```



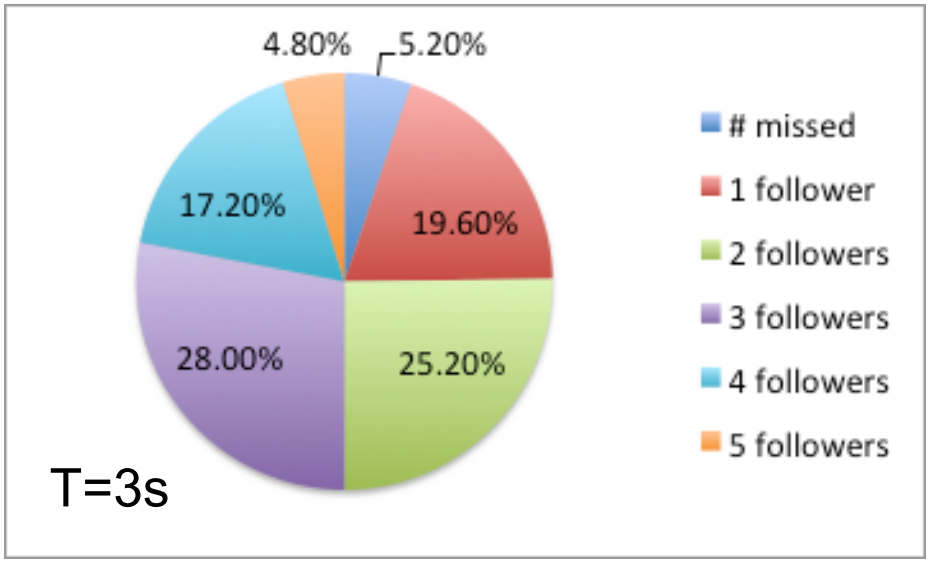
Simulation & Experiment comparison

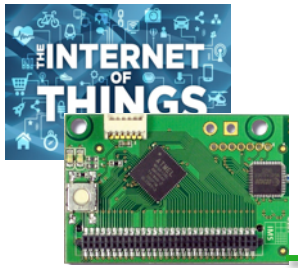
- Cycle length is set to 3000ms

Simulation



Experimentation





Conclusions

- We proposed an adaptive criticality-based MAC protocol to provide duty-cycle support for mission-critical surveillance applications with image sensors
- We linked the node's activity period to the sentry's activity, taking into account both application's criticality and coverage constraints
- Compared with a static duty-cycle approach, our protocol reduces the number of missed alerts and the energy consumption (by 44%) while maintaining the same level of responsiveness
- We also validated our proposition with implementation on WaspMote sensors