

*Methods for
Unsynchronized Duty-Cycling in Specknets*

Steven Wong Kai Juan

Institute for Computing Systems Architecture

School of Informatics, University of Edinburgh, U.K

(k.j.wong@sms.ed.ac.uk)

The Research Consortium in Speckled Computing



The objective of a Speck MAC

“To allow the radio to be turned off sporadically or periodically to save power thus allowing specks to survive longer...”

- *Energy efficiency* - Need Specks to live as long as possible
- *Robustness* - Network does not collapse when some Specks fails
- *Adaptability* - Specks are mobile and Specknet are very dense networks. Must scale.
- *Collision avoidance* - Too many collisions waste energy
- *Throughput* - Delivery ratio and bandwidth of link.
- *Delay or latency* - Need to arrive on time before value is useless
- *Fairness* - All Specks must have a fair share of the bandwidth

PRIORITY

Issues Duty-Cycling using Synchronization

- **Duty-cycling = turning receiver on periodically**
 - Specknet is a mobile ad-hoc network.
 - Synchronize to whom???
 - Need to account for cumulative drifts over long periods (e.g a new packet only every 100sec)
 - Need to synchronize to receive, need to receive to synchronize (How fireflies synchronise if they can't see?)
 - Would the need for synchronisation give raise to other point of failures in the network?
 - What if two or more locally synchronised Specknets meet?

Duty-cycling without Synchronization

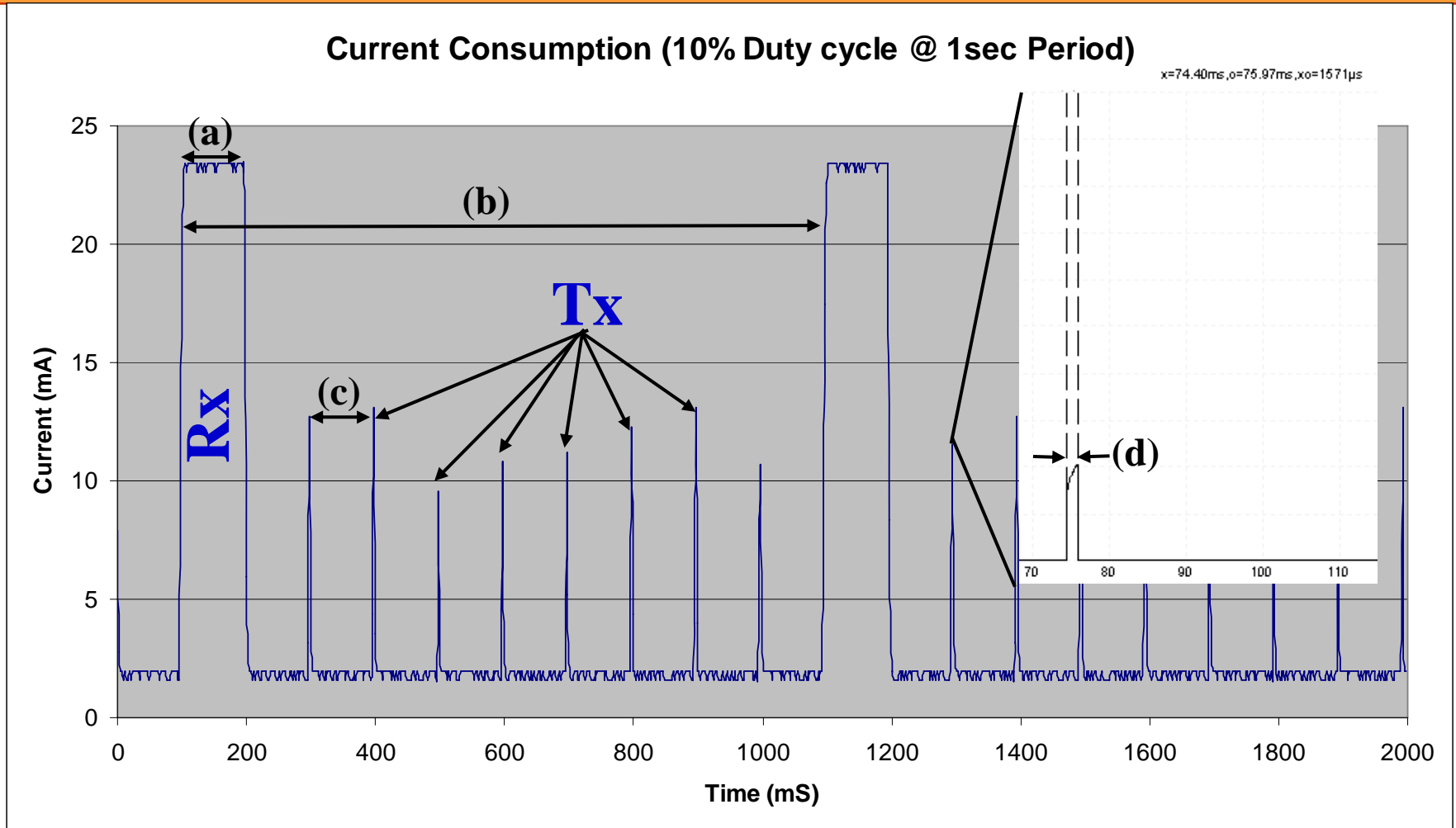
**I will only
listen to her
one day in a
month**

**I will nag (the
same things) at
him everyday of
the month**



Biologically inspired wife-based protocol that works!!!

Actual Power Consumption (Radio+Processor)



a : Rx On period

b : Duty cycle period

c : Retransmission interval

d: Tx On period

Rule of thumb: $c < a - (\text{max. clock skew} * 2)$

Rx Duty Cycle = a/b

Tx Duty Cycle = $d(c/b)$

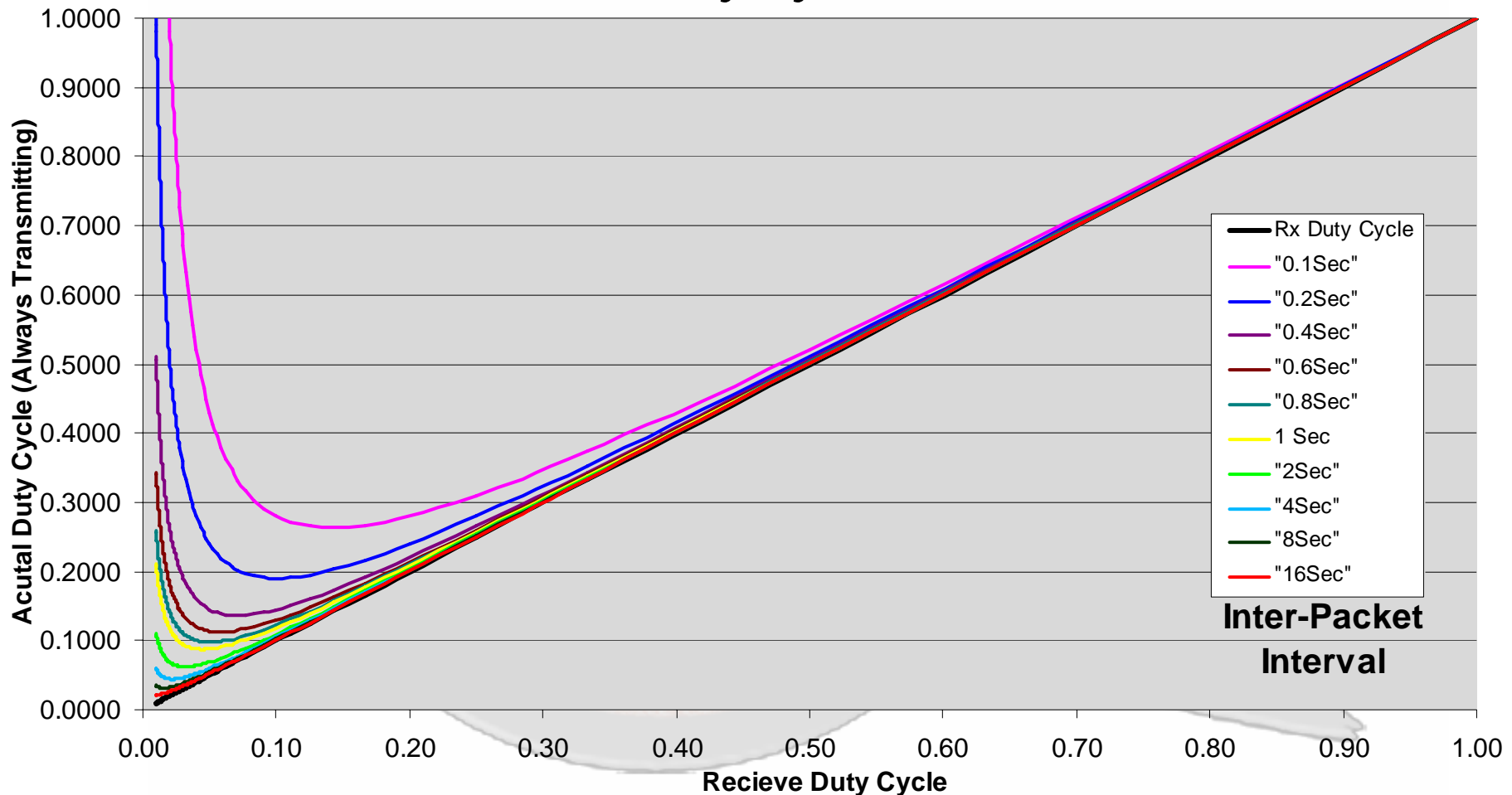
Latency = $b / 2$

Bandwidth = a/b

$d = 1.571\text{ms}$

Transmission Overheads

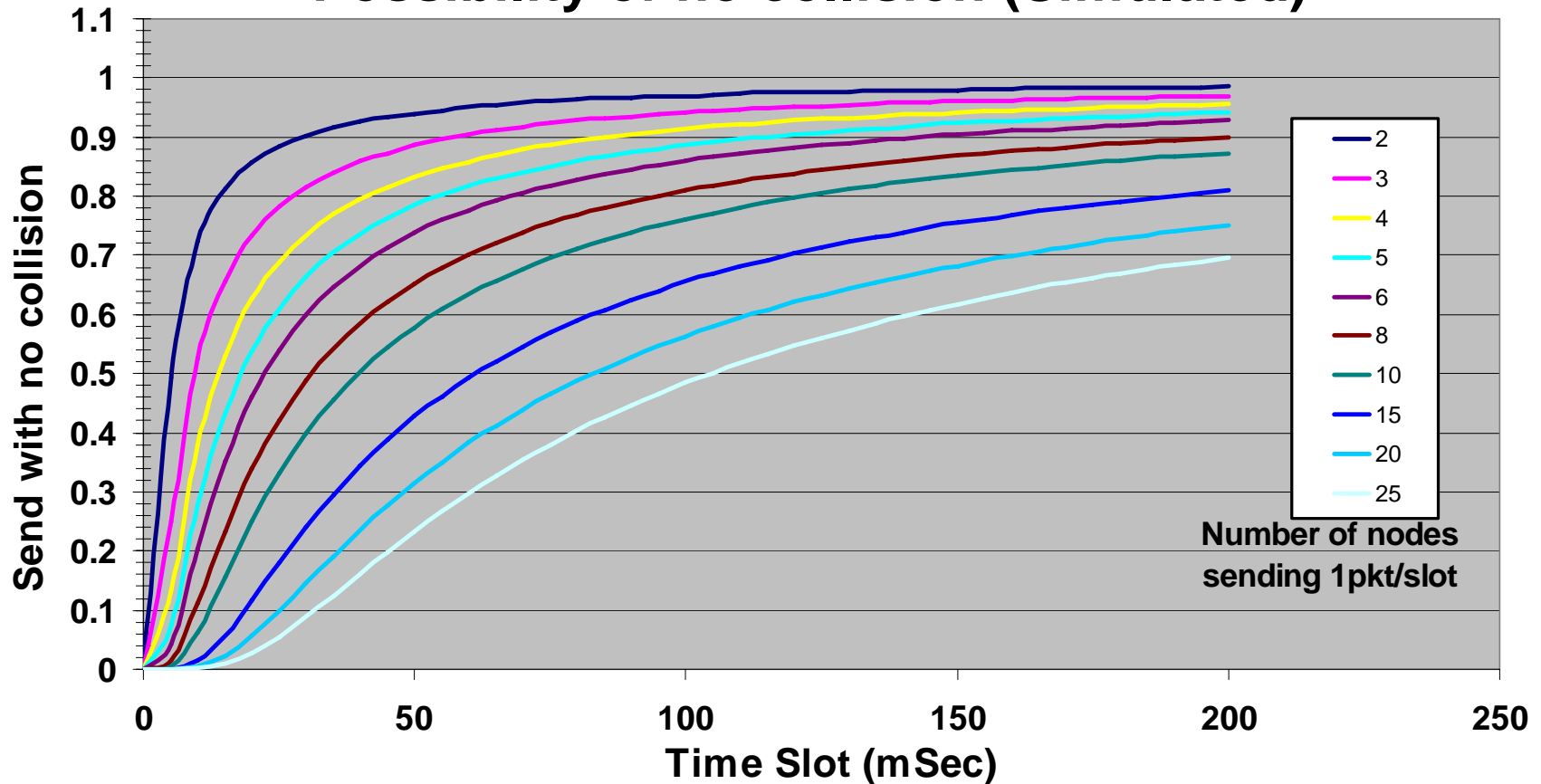
Full Traffic Duty Cycle Overheads



Setting a lower duty-cycle rate can lead to more power wasted!!!

Probability of No Collision

Possibility of no collision (Simulated)



Higher the number of neighbours transmitting = More Collisions

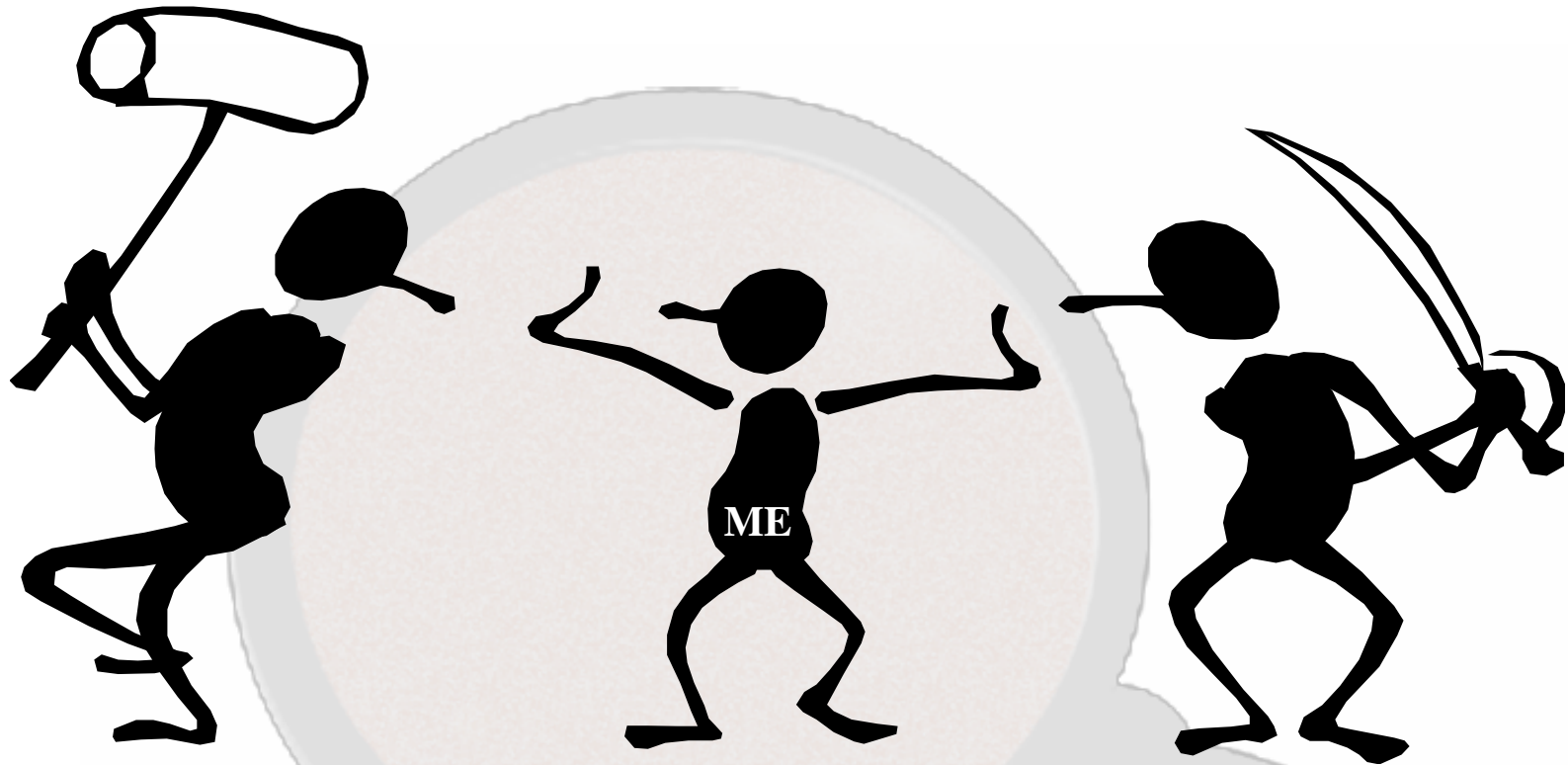
Concluding simMAC

- **All in all...simMAC**
 - is an extremely simple MAC algorithm
 - requires minimal setup time
 - supports mobility
 - works well only when number of packets to be sent are low
- **Look into ways for adaptive duty-cycling**
 - As seen in previous slides, duty-cycling must adapt to traffic in order to save maximal power (profiling???)
- **Support for unicast traffic**
 - Algorithm naturally supports broadcast traffic
 - To support uni-cast traffic efficiently,
 - Specks would send the “time left to turning receiver on” into the header of every retransmission packet.
 - Whenever a Speck want to uni-cast a packet, it will check cache to estimate when destination node will turn receiver on

“large-scale software simulations through small-scale real-time hardware emulations”

- **Use of PerSpeckz64 for ‘automated’ hardware emulations**
 - Support over the air programming for all ProSpeckz
 - Allows for individual ProSpeckz to be turned on/off
 - Allows current consumption to be monitored either from each ProSpeckz or from the whole board using oscilloscope
- **Demonstration scenario**
 - 2 algorithms to be tested.
 - ROMs stored as SEND10.ROM (1sec) and SEND01.ROM (0.1sec)
 - 2 runs for each algorithm. In each case, 6 nodes are chosen randomly to participate in each emulation
 - Statistics are collected and stored on file

FINALLY...



QUESTIONS AND ANSWERS (£4000 Xmas Tree)