Security of MICA*-based / ZigBee Wireless Sensor Networks

Dan Cvrcek, Matt Lewis, and Frank Stajano

Cambridge University
Computer Lab
and myself also
Brno University of Technology
Department of Intelligent Systems

28 December 2008

BUSLab

BUSLab

Our approach

What we did

1. Development of systems for monitoring large civil structures
2. Analysis of off-the-shelf products
3. Interviewing owners and/or operators of the structures
4. Implementing attacks

BUSLab

The kit

- MICAz motes – purchased as part of a Xbow kit
- Stargate – embedded Linux box with a MICAz connector
- USB programmers / receivers for MICAz
- Hardware debugger – Atmel JTAG ICE
- VMware and serveral Linux/Win boxes

Most results applicable to Iris, Tmote, Intel Mote and Intel Mote 2 devices.

They also use TinyOS software or its commercial versions.

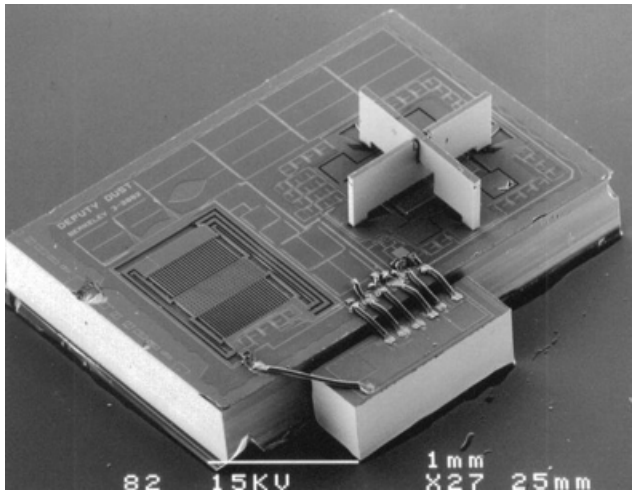
BUSLab

Three-tier architecture

- Your standard three-tier network topology
  - back-end system with database (off-site) Comms: ADSL where possible, GPRS
  - middle: gateway (on-site) Comms: IEEE 802.15.4 / RS-232
  - bottom: sensor nodes (multi-hop ad-hoc network)
- Actual hardware components
  - Crossbow backend system (with PostgreSQL database)
  - Stargate (embedded Linux box as a gateway)
  - Crossbow MICAz (802.15.4 motes)
  - TinyOS 1.1 / XMesh (OS for MICAz motes)

BUSLab

The myth of smart dust . . . and security

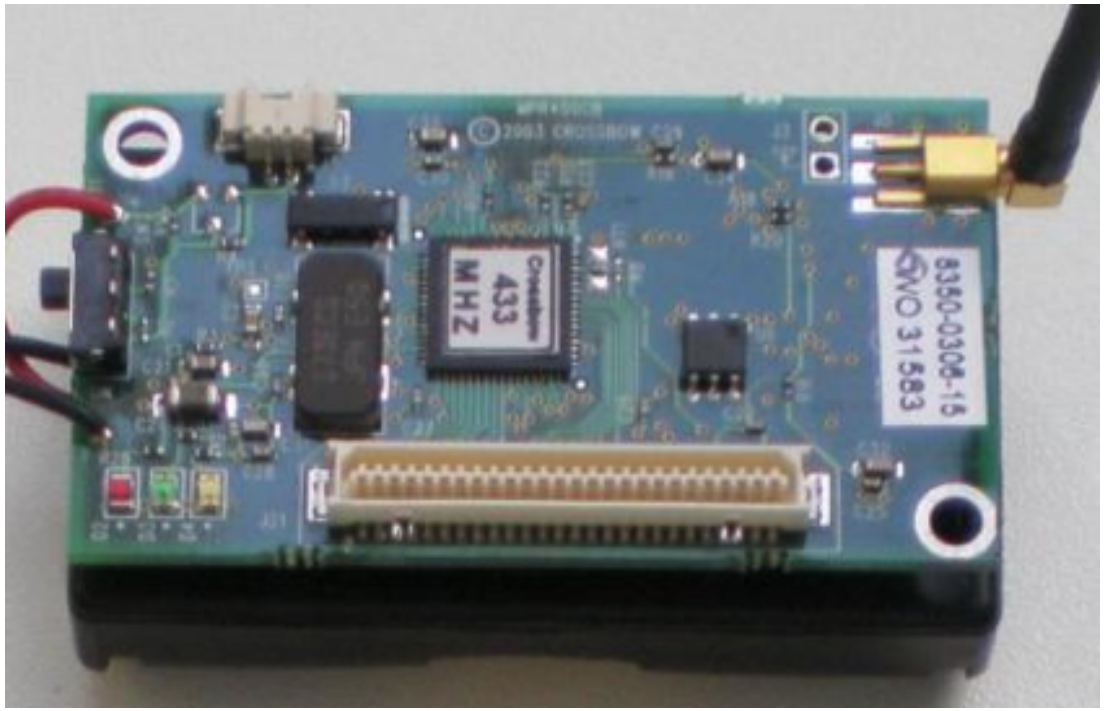


BUSLab

The myth of smart dust . . . and security

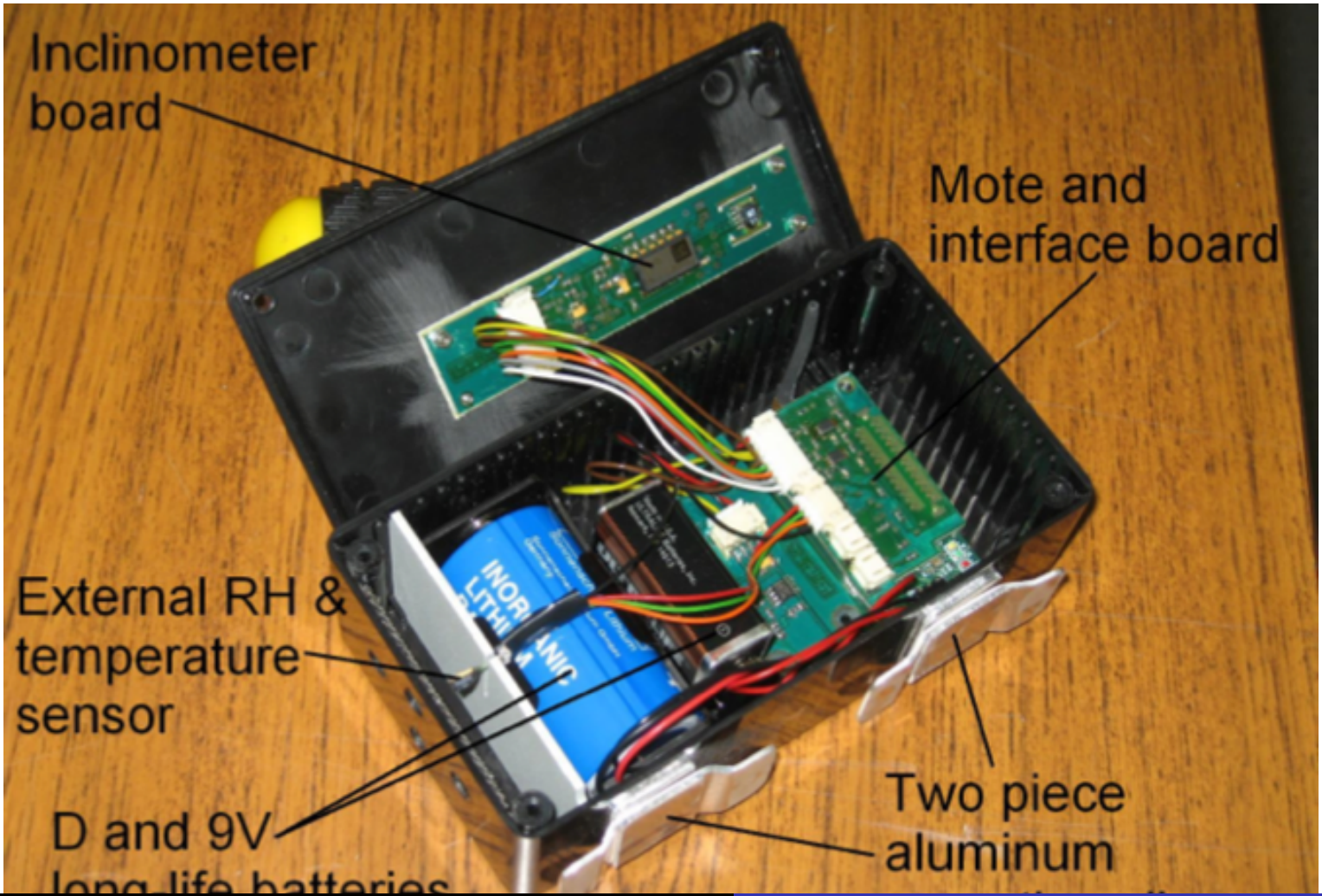
BUSLab

The myth of smart dust . . . and security

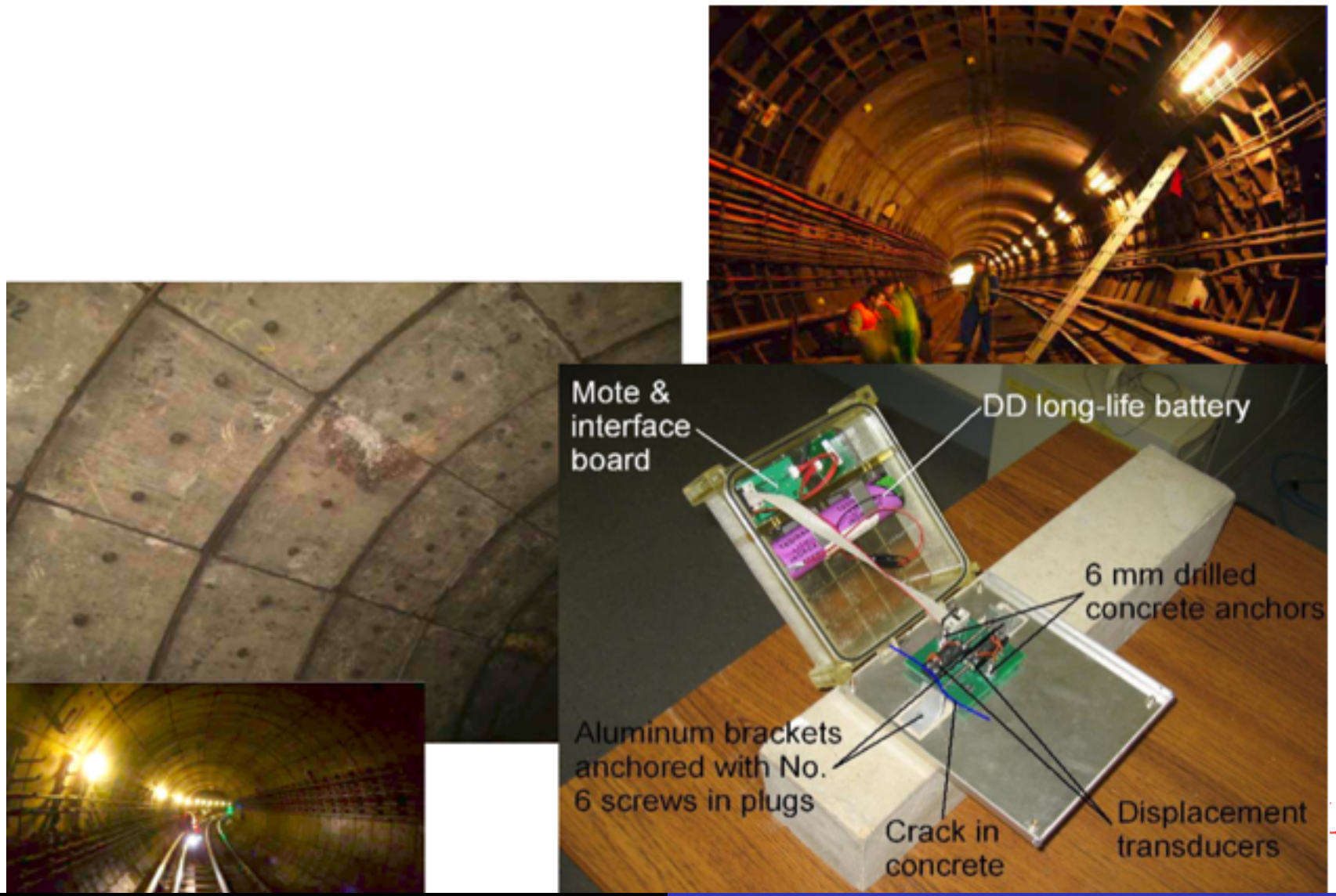


BUSLab

The myth of smart dust . . . and security



BUSLab


Mote &
interface
board
DD long-life battery
6 mm drilled
concrete anchors
Aluminum brackets
anchored with No.
6 screws in plugs
Crack in
concrete
Displacement
transducers

Security properties – by civil structures' operators

What the operators want

- Confidentiality
  - Not really an issue (bridge, underground)
  - Do not rule encryption out but use it by default
- Availability
  - Becomes critical if used be quick response mechanisms
  - Reliability is however a basic requirement

What stroke us was that managers we talked to were not able to predict future applications.
However, overloading existing infrastructures was common (answers to direct questions).

BUSLab

Security properties

- Short-term integrity
  - False negatives – hard to exploit if only few **true** positives!
  - False positives – more disruptive; but viewed with suspicion
- Medium-term integrity
  - Month-scale analysis helps budgeting for maintenance costs
  - Continuous monitoring gives quantitative answers
- Long-term integrity (up to 100+ years)
  - Scale of decades: allows previously impossible research
  - The most valuable data
  - System to be designed for portability and continuous upgrades

BUSLab

Attackers / Risks

This is for now

(imagine a future when you can use ZigBee networks to . . .)

- Terorists?
  - No, they will target bridges and tunnels directly
- Competitors, resourceful attackers?
  - Not likely
- Curious hackers
  - Possibly, to play a bit with the technology

BUSLab

Attackers / Risks

Risk of attacks – not amplified by WSN.

But may be differnet for water pipes.

It is quite easy to get through a pad lock.

Attackers / Risks

Risk of attacks – not amplified by WSN.

But may be differnet for water pipes.

It is quite easy to get through a pad lock.

Classes of attacks

Three basic attack classes

- Remote attacks – from the Internet
- Vicinity attacks – wireless, within communication range
- Physical attacks – physical access to targeted devices

Physical attacks are difficult to do (Becher, Benenson, and Dorseif in 2006)

What about just connecting to a mote's connector??

BUSLab

Classes of attacks

Three basic attack classes

- Remote attacks – from the Internet
- Vicinity attacks – wireless, within communication range
- Physical attacks – physical access to targeted devices

Physical attacks are difficult to do (Becher, Benenson, and Dorseif in 2006)

What about just connecting to a mote's connector??

BUSLab

Code analysis – why?

XMesh – commercial variant of TinyOS

Disassemble XMesh code (15k ASM lines)
and use variable names from TinyOS

```
if (pTable-> flags&NBRFLAG_VALID){
  if (pTable->flag&NBRFLAG_VALID) && (pTable->parent != TOS_
      &&(pTable->parent!=-1) && (pTable->cost !=-1)
      && (pTable->childLiveliness==0))
    computeCost(pTable->cost, pTable->sendEst, pTable->rece
                ulNbrLinkCost, ulNbrTotalCost);
    if (pTable==gpCurrentParent){
      pOldParent=pTable;
      oldParentCost=ulNbrTotalCost;
      oldParentLinkCost=ulNbrLinkCost;
    } else {
      if (ulNbrTotalCost < ulMinTotalCost){
        ulMinTotalCost = ulNbrTotalCost;
        pNewParent = pTable;
```

BUSLab

Routing metrics

- Routing is based on
  1. link cost – ratio of successfully received / sent messages
  2. route cost – sum of link costs to a gateway
- The basis of all computations are message counters
  1. XMesh does not include counters into data messages

Function UpdateNbrCounters

```
sDelta = seqNo - pNbr->seqNo - 1
if (seqNo!=1)  {
    sDelta--  //the only change from TinyOS
}
if (sDelta >= 0) {
    pNbr->missed+=sDelta
    pNbr->received++
} else {  // sDelta was < 0
    if (sDelta < -20) {
       // forget and reinitialise the record
   } else {
        return TRUE
   }

pNbr->lastSeqNo = seqNo
```

BUSLab

XComand vulnerability

XCommand feature

- Xbow's extension of TinyOS
- Direct commands from a gateway to all / selected motes
- No routing but broadcast instead

How to stop rebroadcasting? Messages have msgIDs

- Each mote keeps internal counter (e.g., moteCntr)
- Mote does not process if $moteCntr \geq msgID$

The problem is *moteCntr* is not the last seen *msgID* but the number of processed XCommand messages

BUSLab

XComand vulnerability

XCommand feature

- Xbow's extension of TinyOS
- Direct commands from a gateway to all / selected motes
- No routing but broadcast instead

How to stop rebroadcasting? Messages have msgIDs

- Each mote keeps internal counter (e.g., moteCntr)
- Mote does not process if $moteCntr \geq msgID$

The problem is *moteCntr* is not the last seen *msgID* but the number of processed XCommand messages

BUSLab

Attacks

Hardware limits

Memory is limited $\Rightarrow$ all data structures are size-limited!

- Counters
- Tables
- Crypto information

Not always necessary

BUSLab

Jamming

Jamming

Power used by attacker power wasted by victim

Or can it be used for “smart” attacks?

Jamming

Jamming

Power used by attacker $>$ power wasted by victim

Or can it be used for “smart” attacks?

Jamming

Jamming

Power used by attacker = power wasted by victim

Or can it be used for “smart” attacks?

Jamming

Jamming

Power used by attacker $<$ power wasted by victim

Or can it be used for “smart” attacks?

Jamming

Jamming

Power used by attacker $<<$ power wasted by victim

Or can it be used for “smart” attacks?

BUSLab

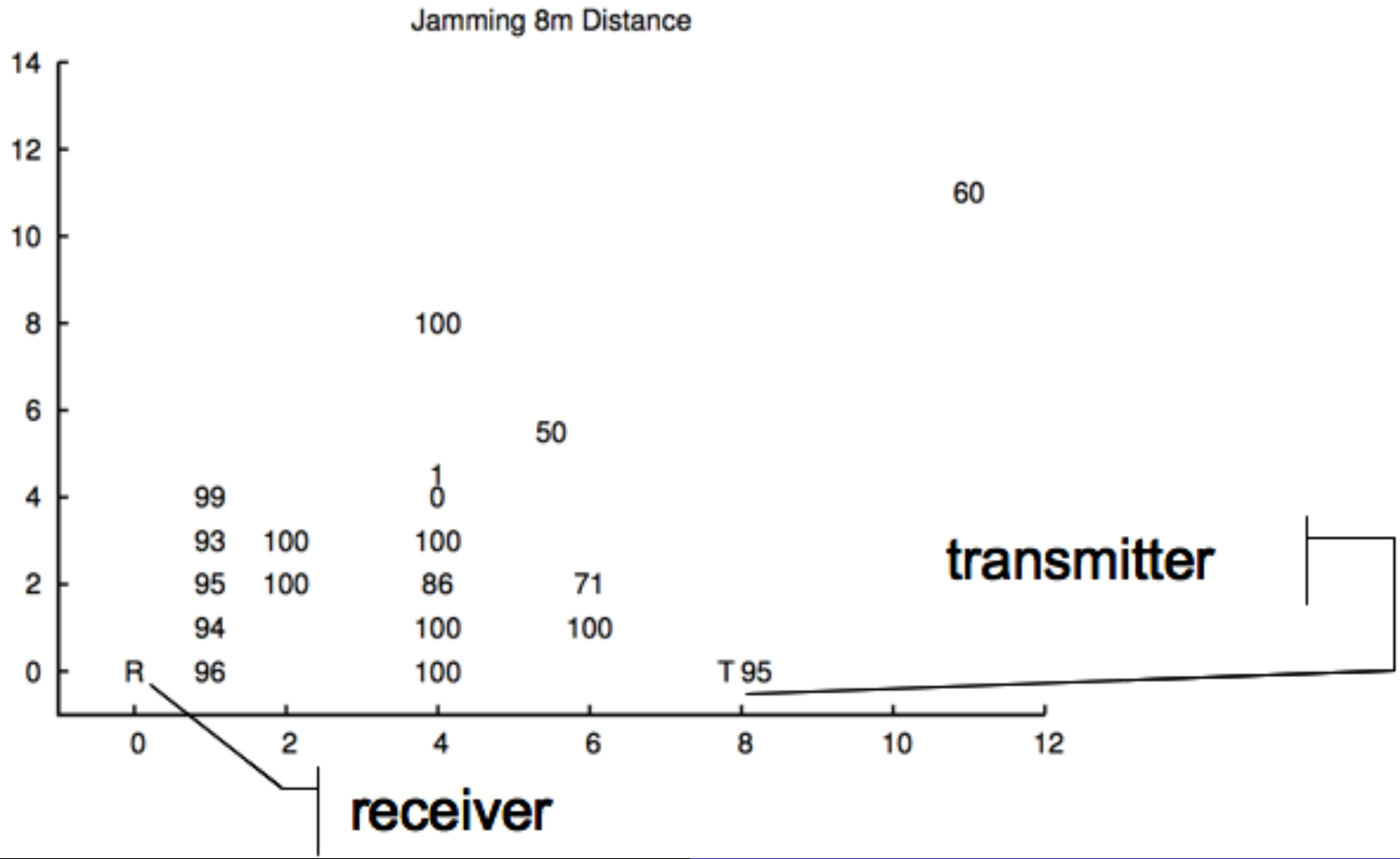
Jamming

Jamming

Power used by attacker power wasted by victim

Or can it be used for “smart” attacks?

Jamming


Jamming 8m Distance
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
R
T
60
100
50
1
0
99
93
95
94
96
100
100
100
86
100
100
71
100
95
transmitter
receiver

Selective jamming

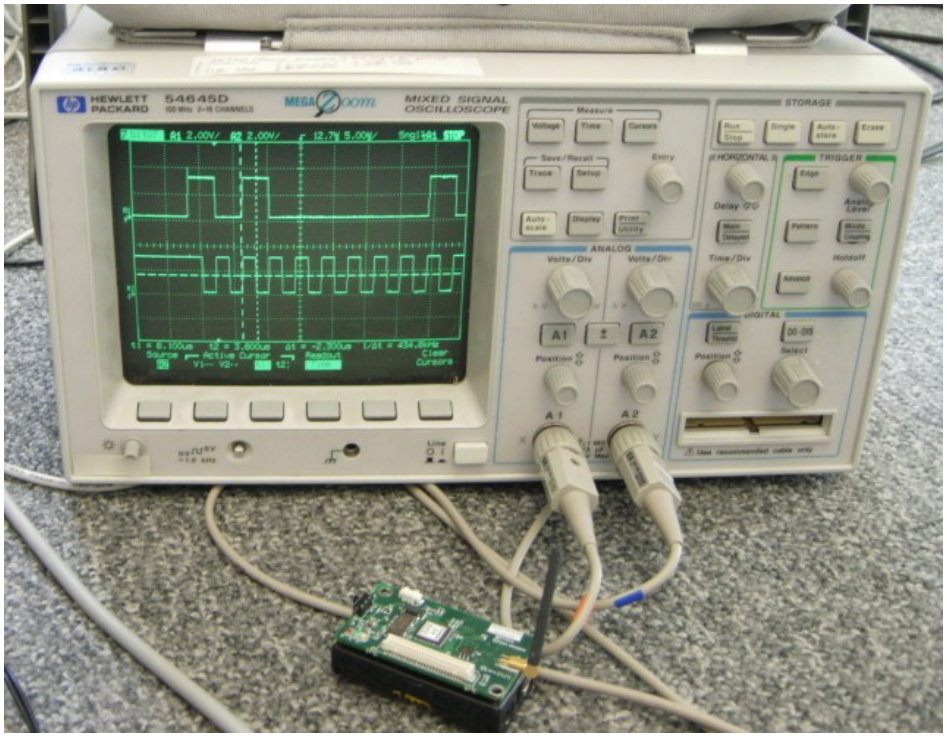
- Jamming always possible – no news
- More interesting – selective jamming
  - Jam packets from mote *X*
  - Jam packets for mote *Y*
  - Jam packets of type *Z*
  - Jam packets with content *C*
- Simple but effective implementation
  - compile criterion into attacking mote's code
  - Deploy the mote and let it listen
  - If a message meets the criterion, jam

Low cost selective jamming

Can we use just one mote to listen and jam selectively?
Problem – IEEE 802.15.4 radio chips work over buffers.

Low cost selective jamming

Solution – debug mode

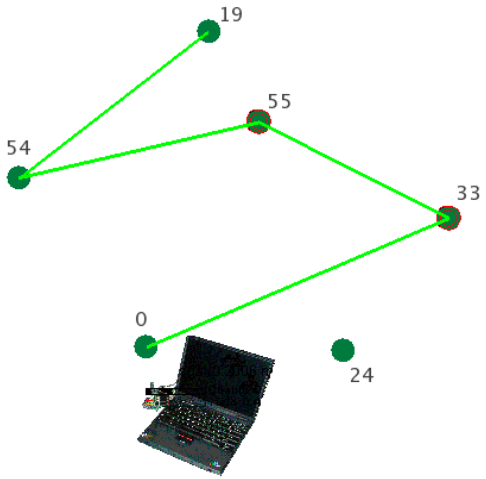


BUSLab

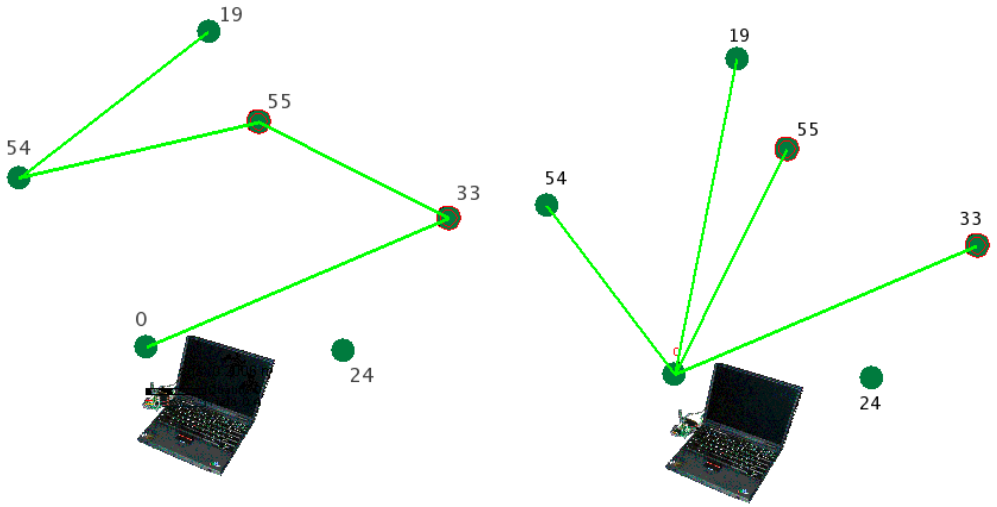
Low cost selective jamming

The final code well below 100 lines of NesC (C macro language)

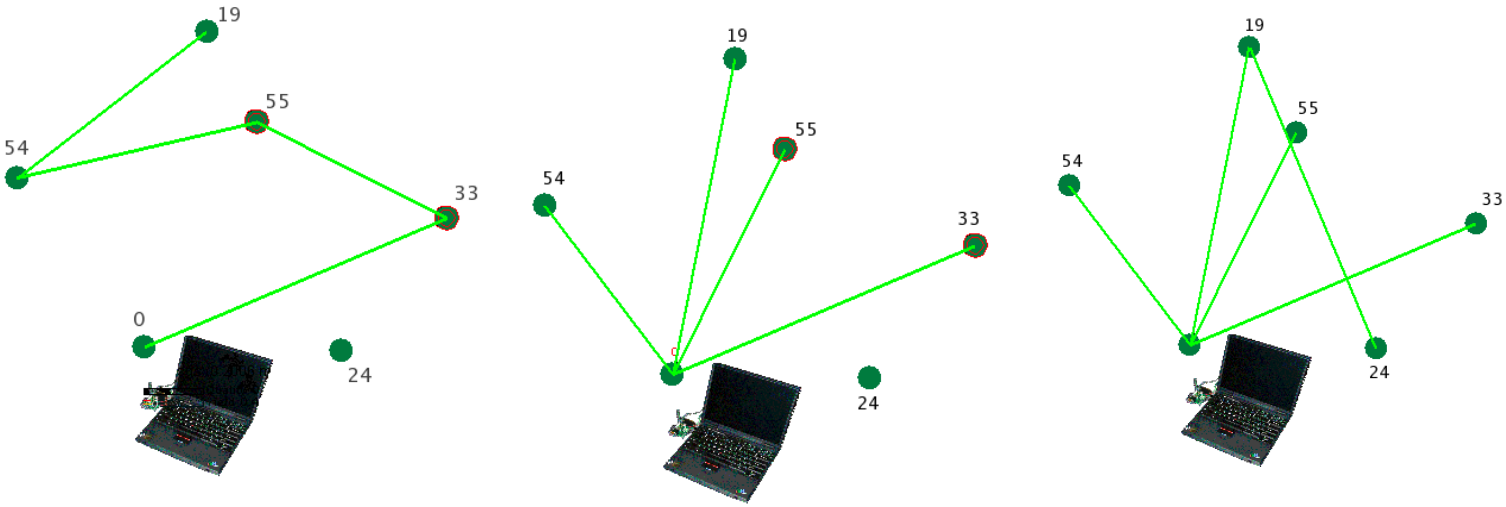
Jamming – transmitting fixed bytes on the same channel.



BUSLab



BUSLab



BUSLab

Protection?

Routing table manipulation

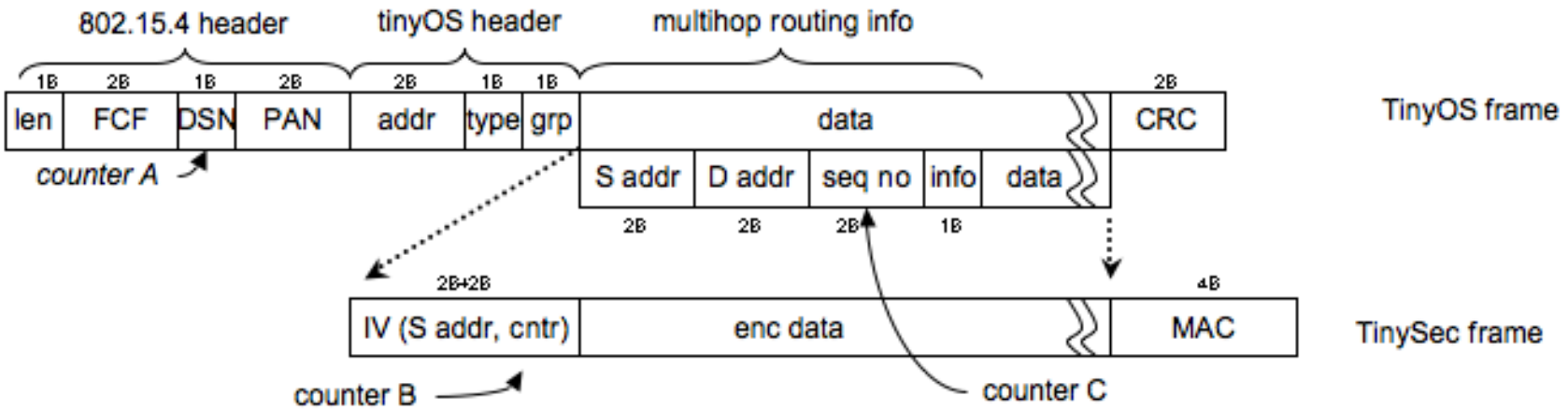
- XMesh nodes broadcast neighbour lists with link quality info
- Nodes update own routing tables by according to messages (no crypto)
- Scapy extension to manipulate/inject messages
- One cen indirectly rewrite routing table of victim

"Sleep deprivation attack"

1. Create a routing loop between two victims
2. Watch them zip messages back and forth (if you want)
3. Let them drain batteries (duty cycle $< 1\ \% \rightarrow >> 10\ \%$)

BUSLab

Frame formats



BUSLab

Msg replays - counter overflow

Messages rejected if the msg counter < expected value
BUT
If the msg counter > expected value, it will be accepted
(also if msg counter = NOUGHT)

Counter size – 16 bits $\Rightarrow$ let's cause overflow

- Inject fake message the victim will forward
  (data msgs no counter)
- Create a routing loop (quickly eating up counter values)
- Jam ACK frames – the victim will retransmit

Combine any of the three.

Or, as the routing table size is 16,
just add as many new neighbours.

BUSLab

Msg replays - counter overflow

Messages rejected if the msg counter < expected value
BUT
If the msg counter > expected value, it will be accepted
(also if msg counter = NOUGHT)

Counter size – 16 bits $\Rightarrow$ let's cause overflow

- Inject fake message the victim will forward
  (data msgs no counter)
- Create a routing loop (quickly eating up counter values)
- Jam ACK frames – the victim will retransmit

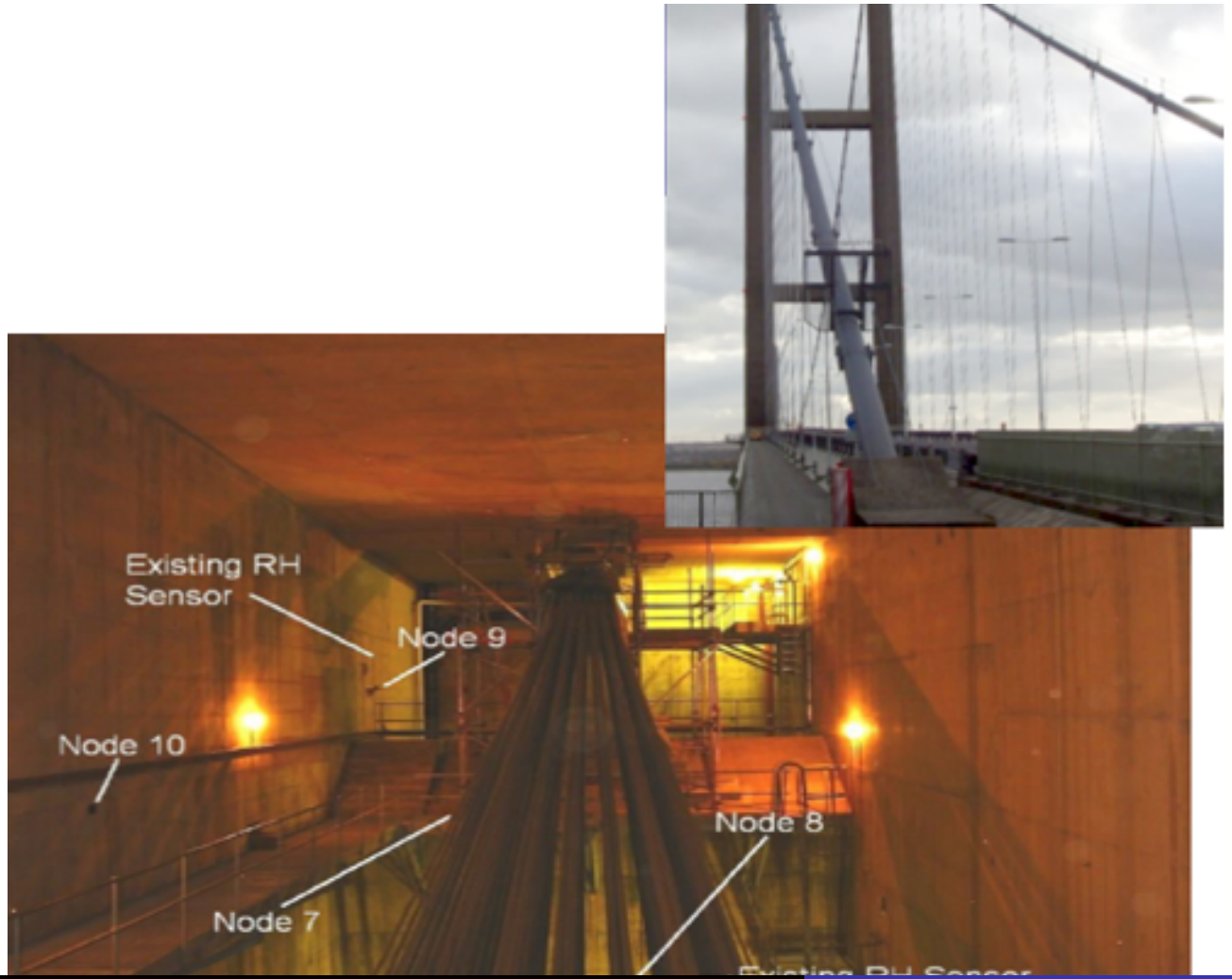
Combine any of the three.

Or, as the routing table size is 16,
just add as many new neighbours.

BUSLab

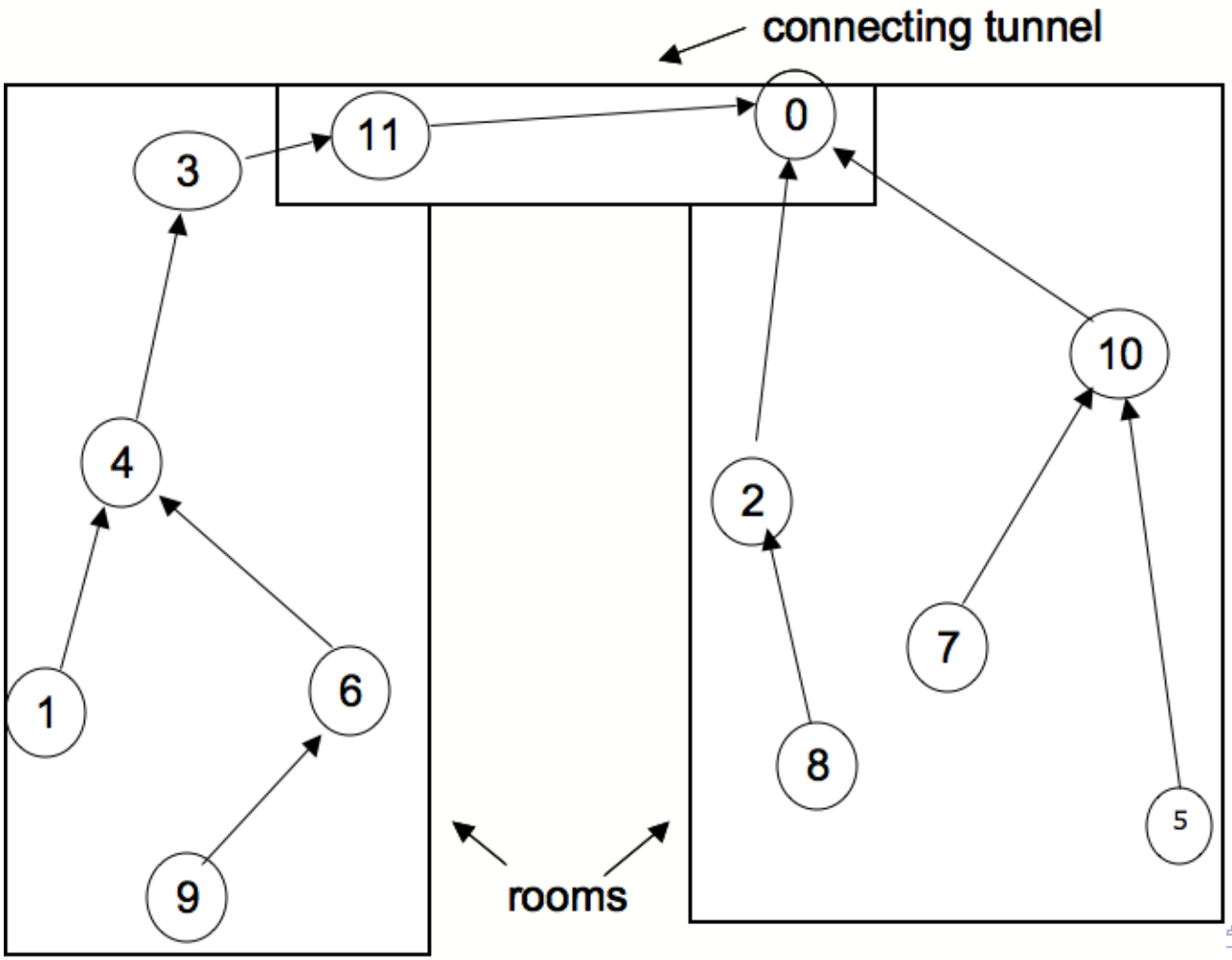
Traffic analysis

The exterior



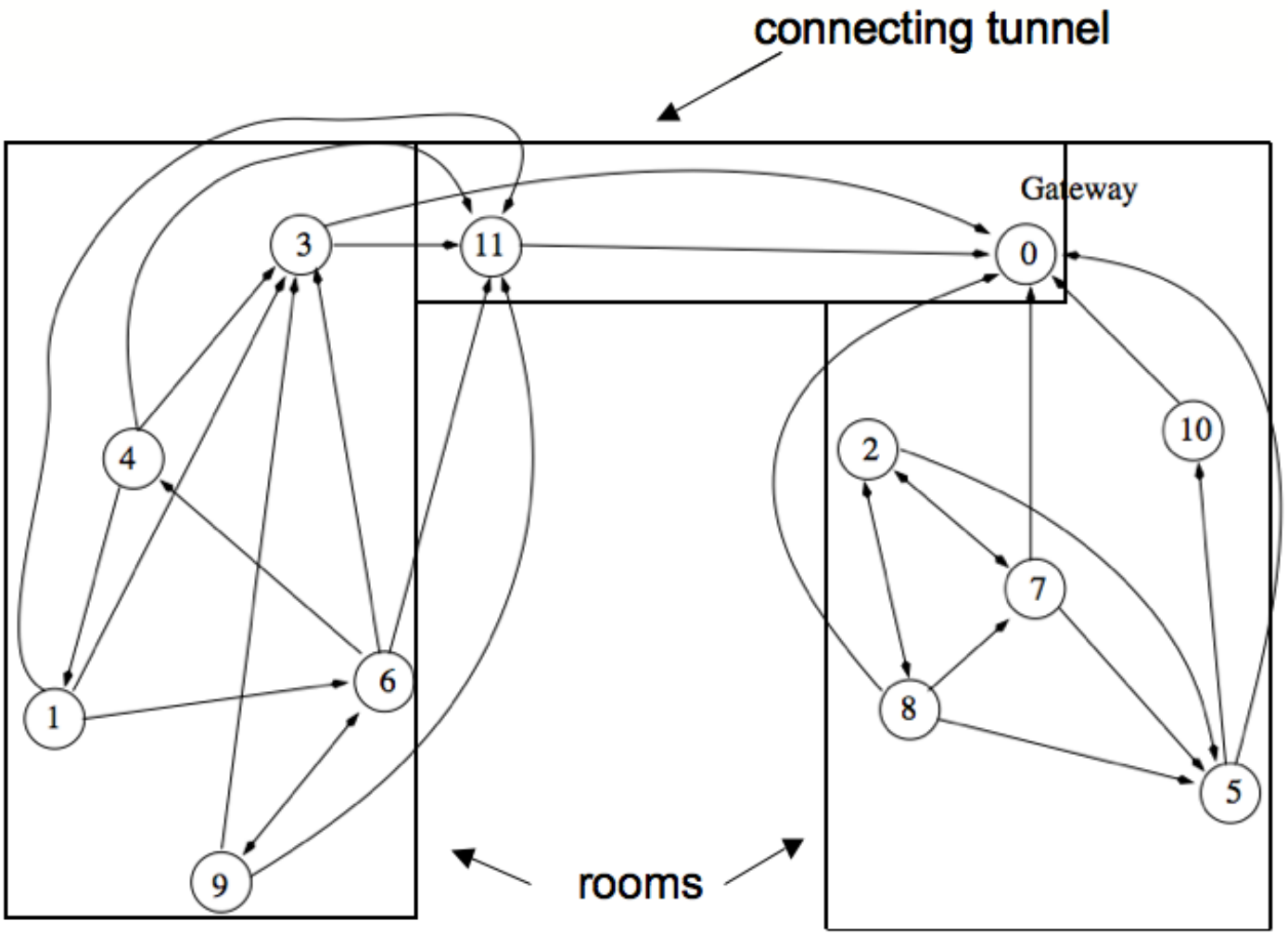
BUSLab

Network topology



BUSLab

Network topology over 24 hours



BUSLab

What can one play with

Traffic analysis allows

- Topology – never encrypted
- Routes to gateway – partially encrypted
- Data – encrypted

Cryptography

What is out there?

TinySec – crypto library allowing encryption and MACing

Only MICA2 motes – not ZigBee motes

We had to port TinySec for MICAz motes first!
RF chip $\Leftrightarrow$ [Tx/Rx $\leftrightarrow$ TinySec $\leftrightarrow$ Tx/Rx $\leftrightarrow$ Processing]
$\leftrightarrow$ - 1 byte for MICA2 (868/916 MHz RF)
$\leftrightarrow$ - 1 frame for MICAz (2.4GHz / 802.15.4 RF)

BUSLab

What is out there?

TinySec – crypto library allowing encryption and MACing

Only MICA2 motes – not ZigBee motes

We had to port TinySec for MICAz motes first!
RF chip $\Leftrightarrow$ [Tx/Rx $\leftrightarrow$ TinySec $\leftrightarrow$ Tx/Rx $\leftrightarrow$ Processing]
$\leftrightarrow$ - 1 byte for MICA2 (868/916 MHz RF)
$\leftrightarrow$ - 1 frame for MICAz (2.4GHz / 802.15.4 RF)

BUSLab

Types of cryptographic problems

- Correctness of implementation
- Computational limitations
- Usage limits / errors

Implementation errors

Processing of eight-bytes block: CBC-MAC

```
for (i = 0; i < msgLen; i++) {
   // unroll
   partial[pos++] ^= msg[i];
   if (pos == 7) {
      if (!call BlockCipher.encrypt
                (&context->cc, partial, partial)) {
         return FAIL;
      }
      pos = 0;
   }
```

BUSLab

Hardware limits for crypto

Personally, I don't think there are any.

TinyOS

- MAC length 4 bytes
- Power efficient symmetric ciphers
- Public key cryptography?

Power consumption of computations v communication

- MAC overhead, synchronisation, . . .

BUSLab

Hardware limits for crypto

Personally, I don't think there are any.

TinyOS

- MAC length 4 bytes
- Power efficient symmetric ciphers
- Public key cryptography?

Power consumption of computations v communication

- MAC overhead, synchronisation, . . .

Cryptographic boundary

TinySec is excellent because it's transparent for gateway

Different treatment of wireless and wired interfaces

A few words from Xbow

A few points from communication with Xbow

- Our customers are universities (i.e., they only want to play)
- We agree that security issues highlighted must be solved
- We are not interested in security until we have customers

Project webpage:

http://www.cl.cam.ac.uk/research/security/sensornets/

BUSLab

A few words from Xbow

A few points from communication with Xbow

- Our customers are universities (i.e., they only want to play)
- We agree that security issues highlighted must be solved
- We are not interested in security until we have customers

Project webpage:

http://www.cl.cam.ac.uk/research/security/sensornets/

BUSLab

A few words from Xbow

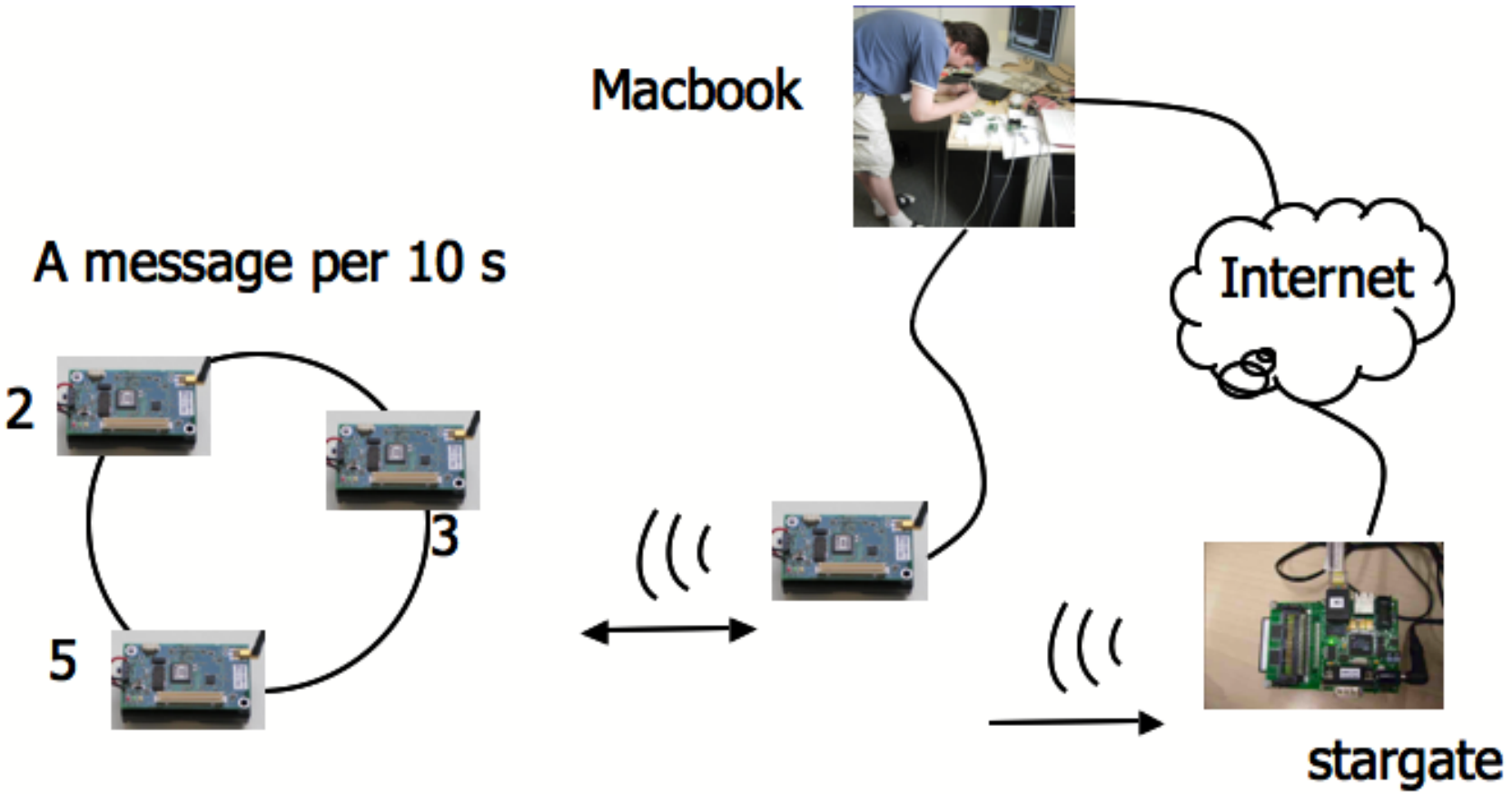
A few points from communication with Xbow

- Our customers are universities (i.e., they only want to play)
- We agree that security issues highlighted must be solved
- We are not interested in security until we have customers

Project webpage:

http://www.cl.cam.ac.uk/research/security/sensornets/

BUSLab



BUSLab