A Test-Bed for Mobile Ad-hoc Networks

How Much Can Watchdogs Really Do?

Sonja Buchegger, Cedric Tissieres, Jean-Yves Le Boudec
EPFL (Swiss Federal Institute of Technology Lausanne)

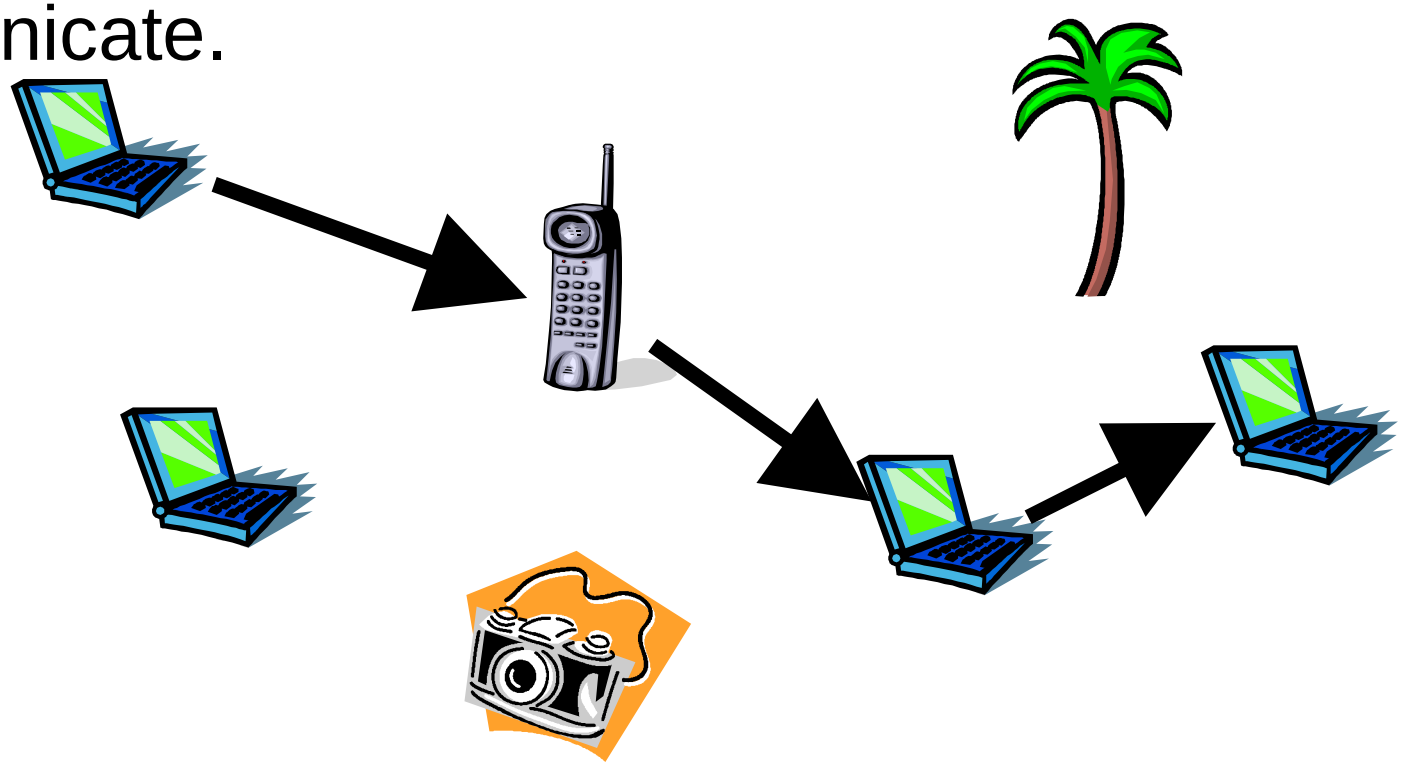
WMCSA, December 3, 2004

Presentation Outline

- Problem: Detecting Misbehavior in Mobile Ad-hoc Networks
  - Attacks on Dynamic Source Routing (DSR)
  - Detectability of Attacks
- Proposed Solution:
  - Enhanced Passive Acknowledgment
  - Test-Bed
- Performance Evaluation: Some Experimental Results
- Related Work
- Conclusions

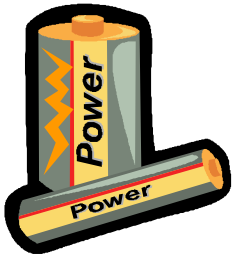
Mobile Ad-hoc Networks

- Network of devices, no infrastructure, nodes forward packets for others. Nodes cooperate to communicate.



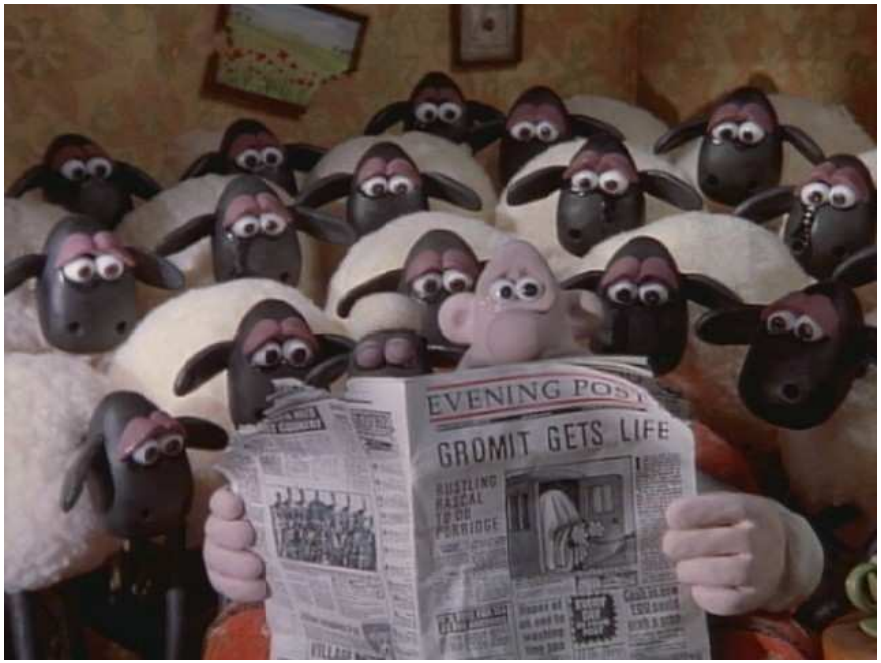
But Why Cooperate? Misbehavior Pays Off

- Selfish: to save power
  - Example: No or incorrect forwarding
- Malicious: to attack the net
  - Example: Route deviation
- Faulty: (no reason)
  - Example: Repeating packets



Here's the Dilemma!

- Tragedy of the Commons:
- Free ground for everyone to let sheep graze
- Individually: good to put many sheep
- Overall: too many sheep!

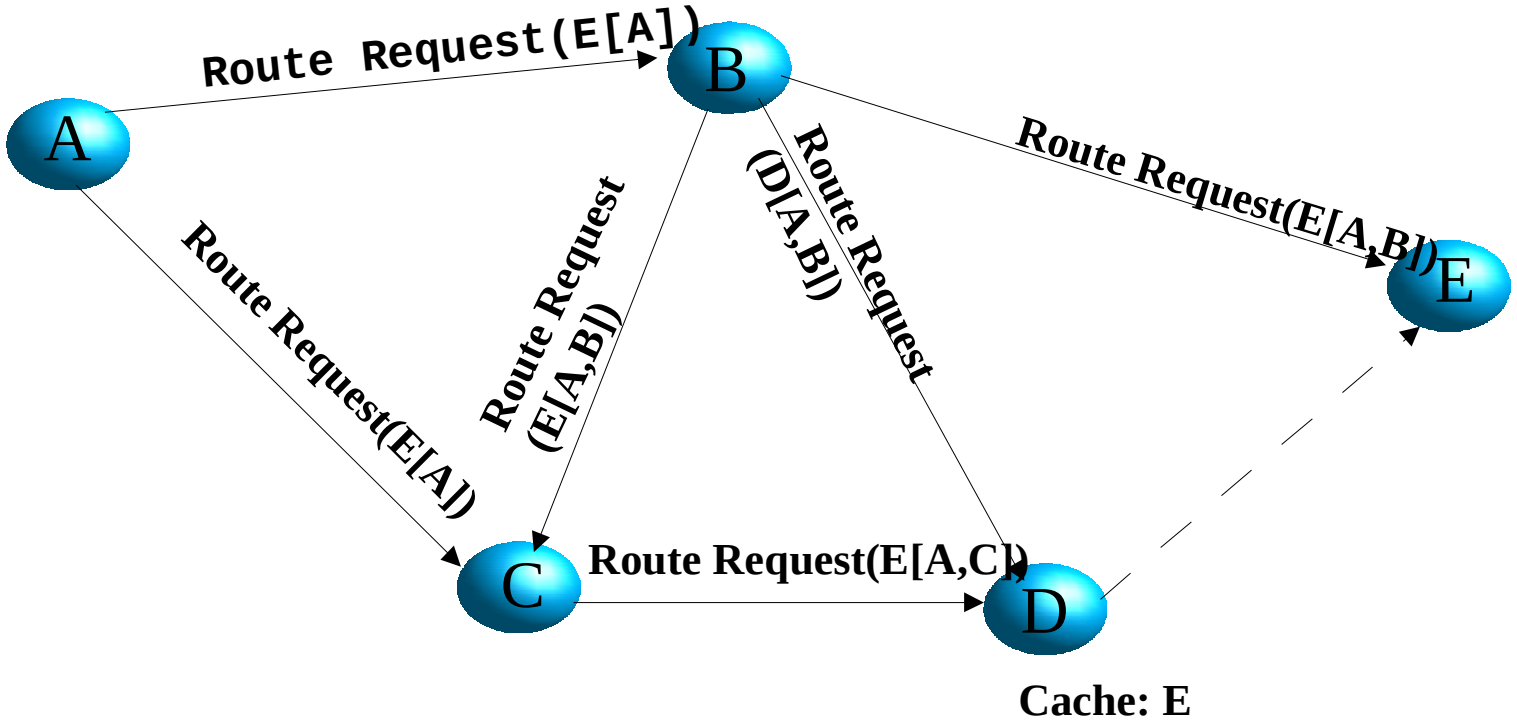


Problem Statement

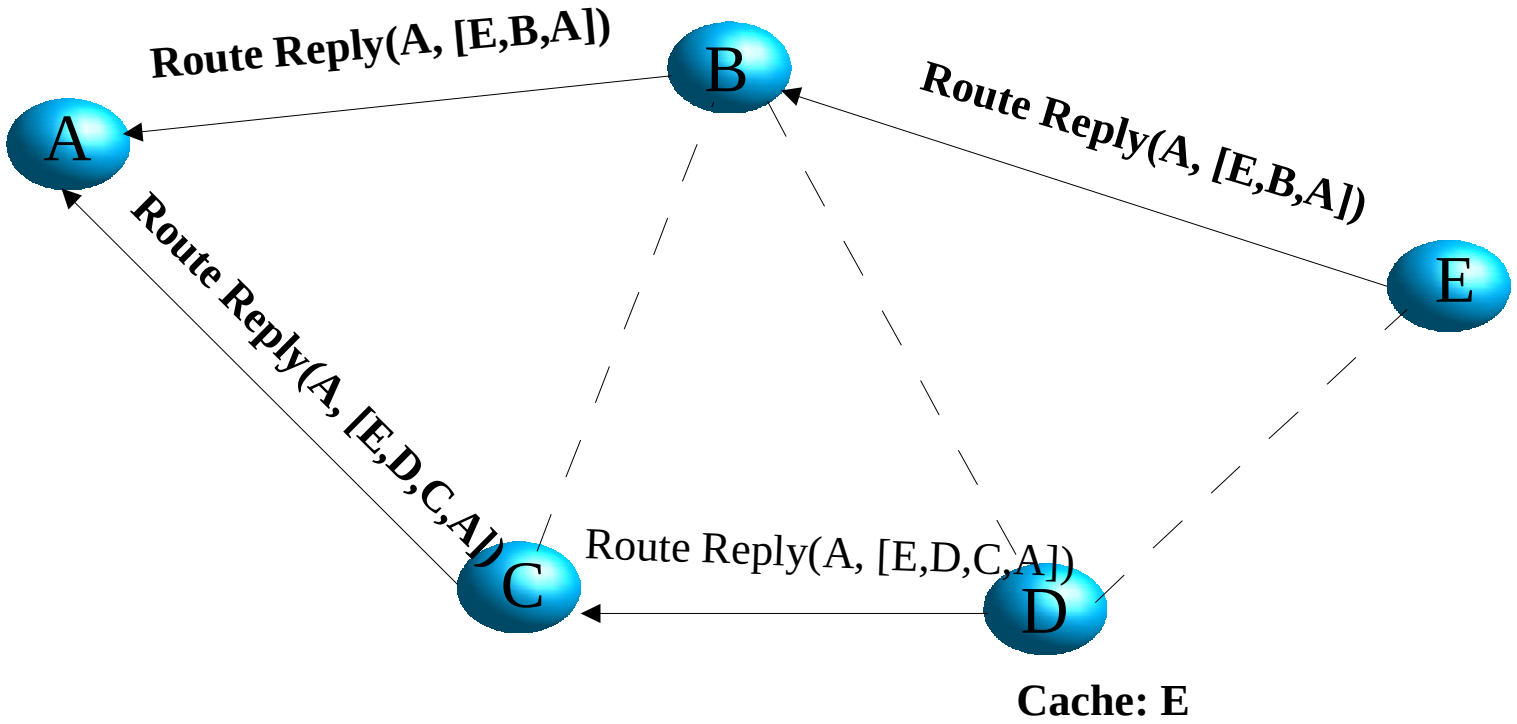
- How can we make a system work despite misbehavior?
- Which types of misbehavior are actually detectable and how?

Background: Dynamic Source Routing (DSR)

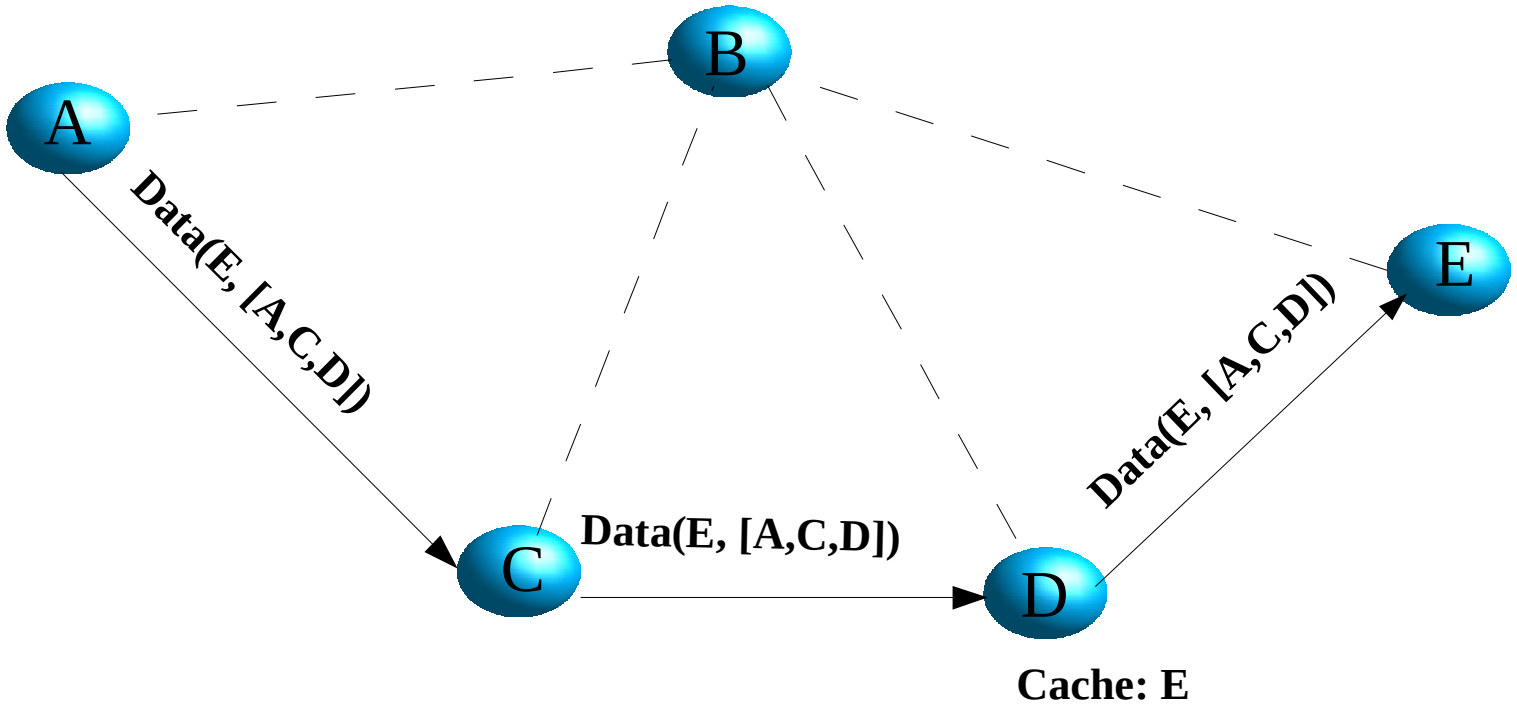
DSR - Route Request



DSR - Route Reply



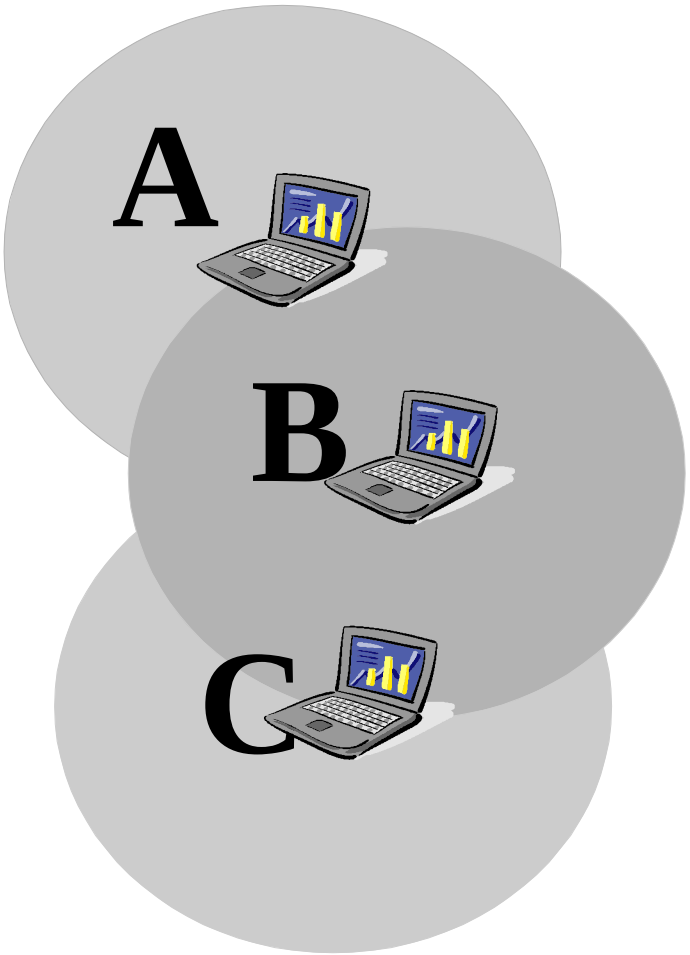
DSR – Data



Acknowledgments in DSR

- Explicit ACK
- Passive ACK
- Link-layer notification

Enhanced Passive Acknowledgment



- PACK: Overhearing of
  - Forwarding
  - Tampering
  - Fabrication
- In addition: Packet Reception

Attacks on DSR

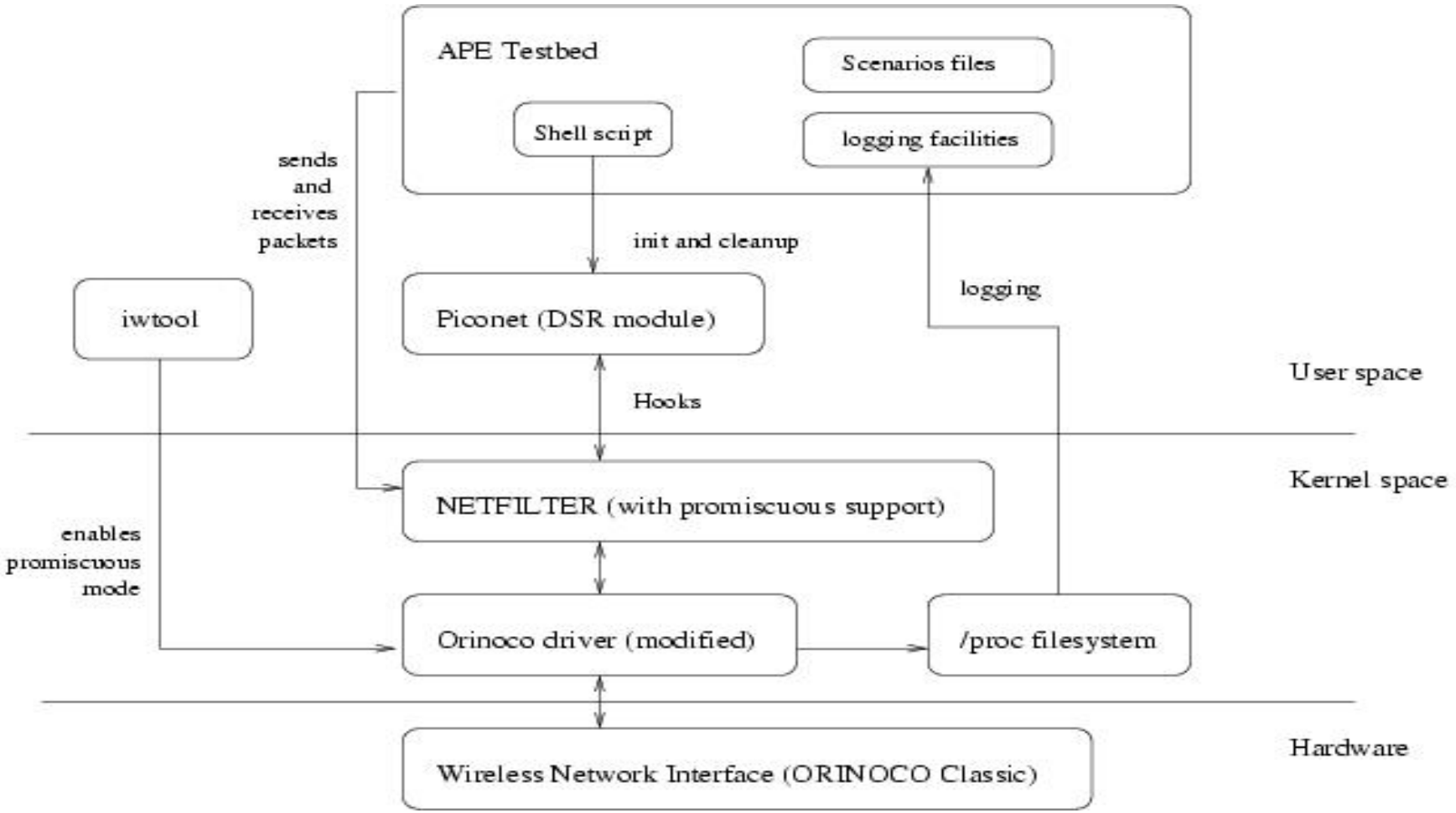
- Dropping Attacks
  - All or partial
  - Omit Route Error
- Modification Attacks
  - Forged routing packets
  - Added nodes
  - Last Hop External
  - Salvage intact routes
  - Loops
  - Tamper with RREQ, RREP
  - Decrease TTL
- Fabrication Attacks
  - Forged RERR
  - Spoofed RREQ
  - Forged RREP
  - Frequent RREQ
- Timing Attacks
  - RREP disproportionally fast

Test-Bed Components

- Piconet with PACK, enhanced PACK, and attacks
- APE
- Netfilter with promiscuous mode
- Pcmcia-cs with promiscuous mode

- Setup: Laptops with Linux kernels 2.4.19 and 2.4.20, Orinoco Classic Gold 802.11b cards

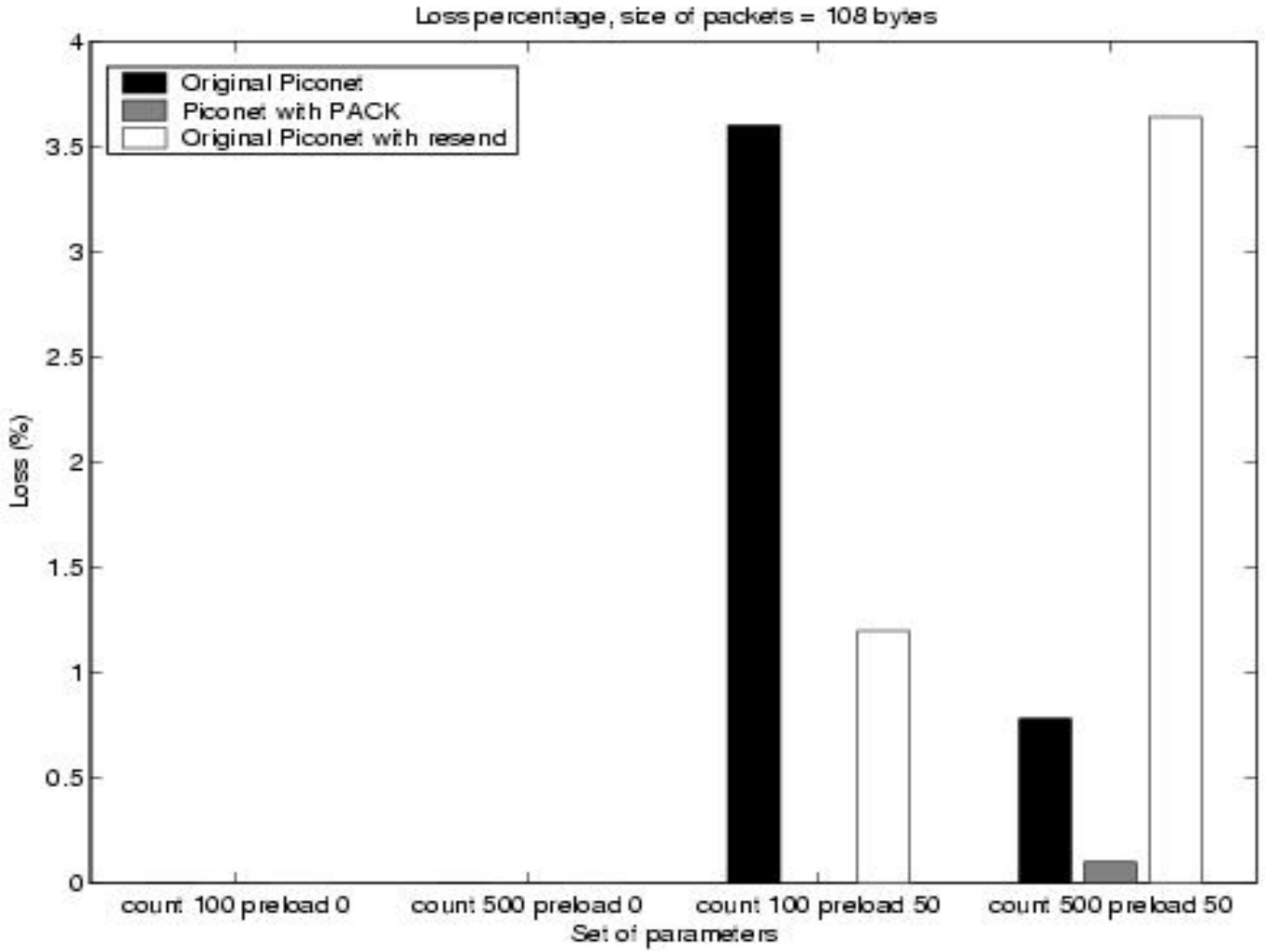
Test-Bed Architecture



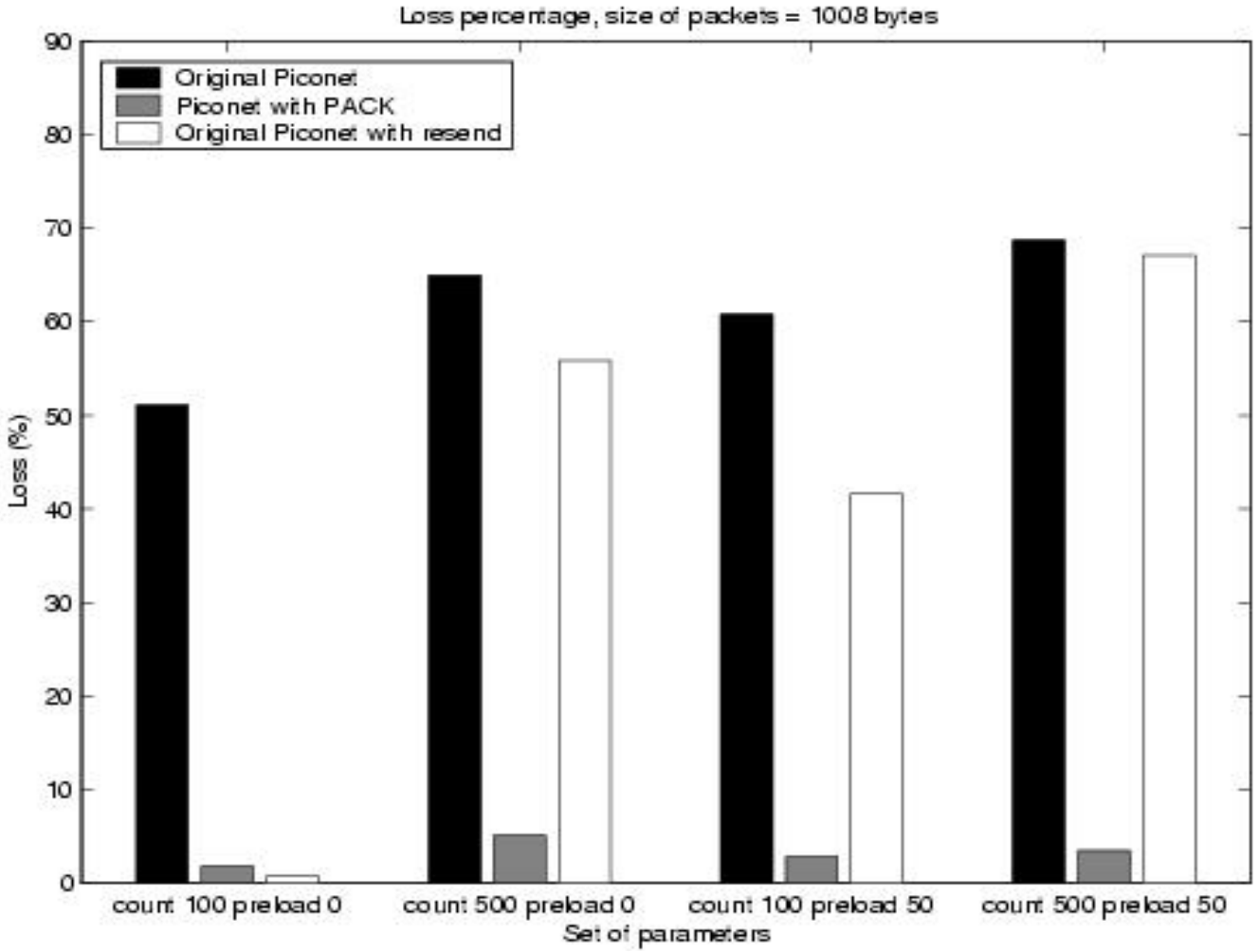
Implemented Example Attacks

- Header Modification
  - Selfish Attacks
    - Remove from RREP
    - RERR modification
    - Attacks work!
  - Malicious Attacks
    - Change Source Route
    - RERR destination
    - Attacks work!
- Partial Dropping
  - Attack works!
- RERR Fabrication
  - Attack works!

Experimental Results



Experimental Results II



Related Work: Economic Incentives

- Forwarding is rewarded.
- Target: selfish/rational nodes
- Examples: nuglets/counters, Crowcroft, Sprite

- Solution only for the non-forwarding type of misbehavior.

Related Work: Secure Routing

- Using Cryptography to secure route discovery
- Target: malicious nodes
- Examples: Ariadne, SRP, S-AODV, BISS

- Solution only for route discovery. Nodes can still deviate traffic or drop packets.

Related Work: Reputation Systems 1

- In MANET or P2P:
  - Keep track of misbehaving nodes, exclude them
  - Target: misbehaving nodes regardless of reason
  - Examples: Watchdog, CORE, Context, OCEAN, ID, Aberer, SECURE
- Either
  - Use only first-hand information, so only detect neighbors, or
  - are vulnerable to spurious ratings, or
  - assume trust transitivity, or
  - only consider negative (positive) information

Related Work: Reputation Systems 2

- E-Commerce
  - History of transactions for future choice of partners
  - Target: human decision makers, agents
  - Examples: E-Bay
- Centralized

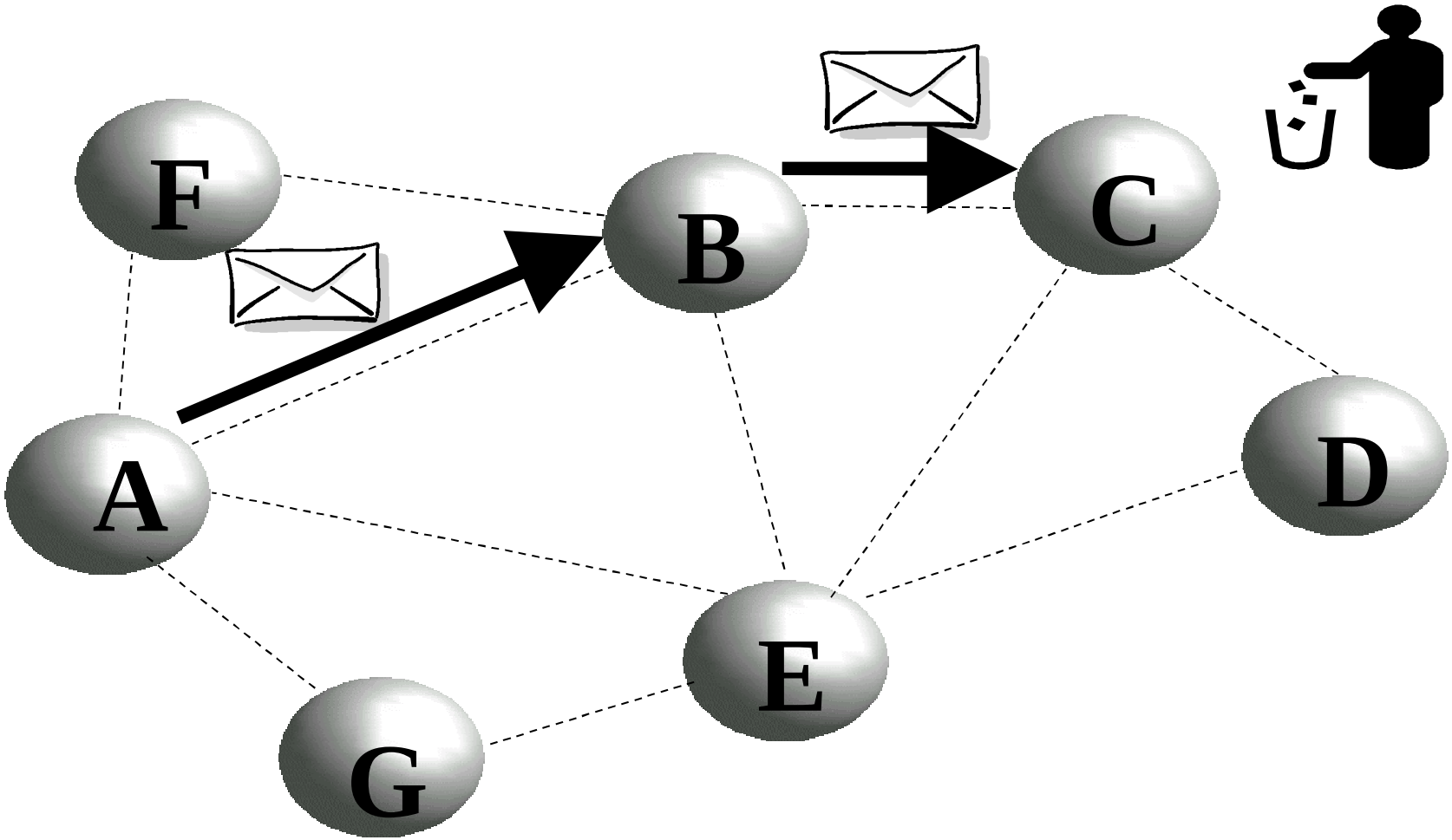
Solution Proposal: CONFIDANT

- Target both routing and forwarding misbehavior
- Regardless whether selfish, faulty, or malicious
- Be able to detect misbehavior before meeting (use second-hand information)
- Cope with spurious ratings
- Fully distributed

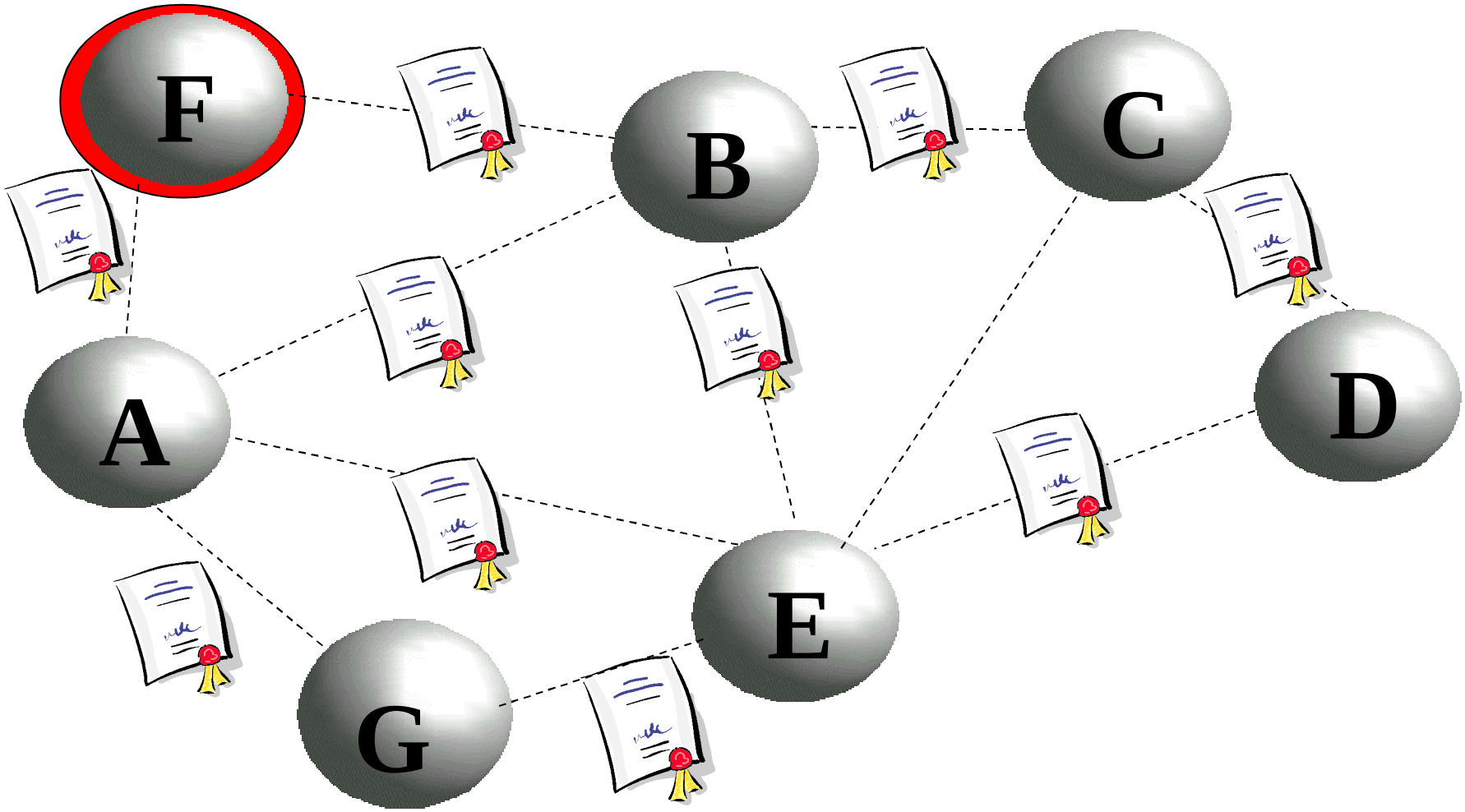
Purpose of CONFIDANT

- CONFIDANT detects misbehaving nodes
  - by means of observation or reports about several types of attacks
- and thus allows nodes
  - to route around misbehaved nodes and
  - to isolate misbehaved nodes from the network, so that misbehavior
    - does not pay off,
    - cannot continue, and
    - routes are functional.

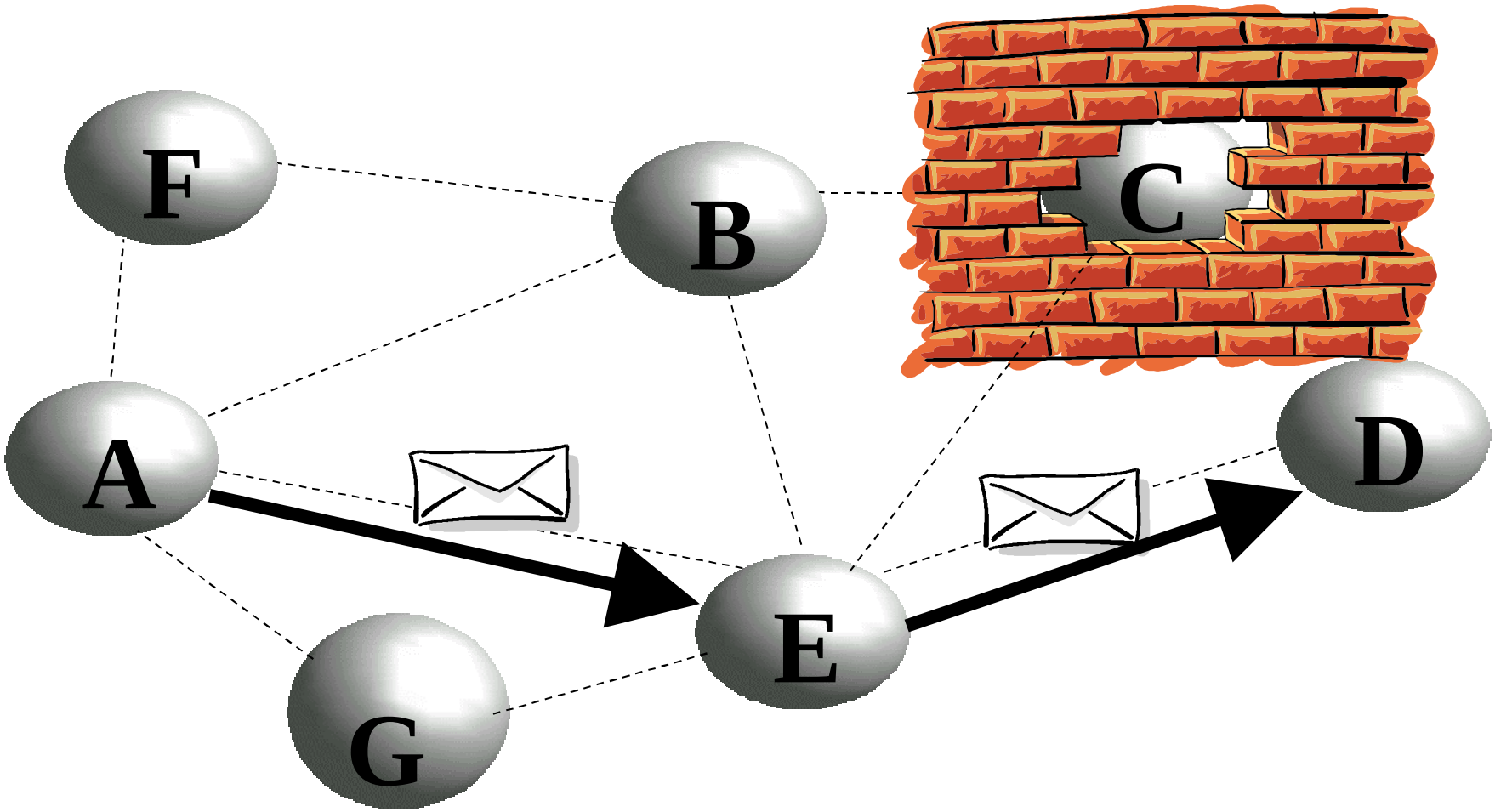
Misbehavior



Publication



Isolation and Rerouting



Conclusions

- Watchdogs can work well
- Enhanced Passive ACK can detect quite a lot
- Watchdogs with enhanced PACK can give useful input to misbehavior detection and reputation systems
- Need to do larger test-bed experiments to find limitations, false positives
- Make code and documentation freely available

Watch This Space!

- Code and Documentation will be available from
- http://icapeople.epfl.ch/sbuchegg
- Soon.