Mobile Adventure

Reconfiguration in ad-hoc networks.

Christian Prehofer
DoCoMo Euro-Labs
Munich, Germany
Overview

- DoCoMo Euro Labs

- Example Problem:
  - The synchronized reconfiguration problem
  - A time-out based algorithm
    - joint work with B. Souville

- Some thoughts & questions
Mobile Adventure

NTT DoCoMo’s R&D Centers

DoCoMo Europe
Aug. 1998
(Standardization)

DoCoMo USA Labs
Nov. 1999

YRP R&D Center
Mar. 1998

DoCoMo Euro-Labs
Nov. 2000

© by DoCoMo Communications Laboratories Europe GmbH

YRP: Yokosuka Research Park
Synchronized Reconfiguration Problem

Simultaneous reconfiguration of mobile nodes
- Software upgrade, reconfiguration
  - may include all communication layers
  - may fail and fallback is needed
- No central control
- All connected nodes shall have same configuration
- Dynamic nature of ad-hoc networks
Application scenarios

- Change ad hoc Routing protocols
  - AODV and DSR
    - Change of routing protocols according to network load and mobility characteristics
    - Fallback signal
      - IEEE 802.11 broadcast message on the MAC layer

- Software radio reconfiguration
  - IEEE 802.11 and Hiperlan/2
    - Main differences on the MAC layer
    - Small differences on the Physical layer
    - No fallback signal

- Change frequency, e.g. in IEEE 802.11
  - No fallback signal
Assumptions and Requirements

- **Assumptions**
  - Mobility of the nodes
    - Ad hoc network may split into several groups
  - No communication possible during reconfiguration
  - Reconfiguration failures possible

- **Requirements**
  - Consistency properties
  - Fallback signals in the case of reconfiguration failures (optional)
Goal of the Algorithm

- **Goal:** Connected groups have same configuration
  - Connected refers to a specific decision time point (agreed time out)

- **Problem**
  - Cannot distinguish two cases
    - node moves away
      - don’t care
    - node fails with reconfiguration
      - May be detected with fallback signal
      - Do fallback

- **Variations**
  - Do fallback if node disappeard, but no fallback signal
Time-out based algorithm: Setup phase

One node retrieves the software

Exchange capabilities

Distribution of SW

Negotiation of maximum time \( T \) for reconfiguration

Start reconfiguration \( t=t_0 \)

Cellular Network

N1

N2

N3
Time-out based algorithm
Reconfiguration phase

Start reconfig. 
\[ t_0 = t \]

\[ t < t_0 + T? \]

yes

Successful Reconfiguration?

yes

Send positive indication

wait for signals from other nodes and distribute them to other reachable nodes

no

Send fallback signal

Variation: Fallback Signal available

No signal or at least one fallback signal from one currently reachable node?

yes

No positive indication?

yes

Fall back to the old software

no

Commit to the new software
Some thoughts & questions

• "Coordinated operations" are not possible in asynchronous systems with failures (in theory)
  – e.g. distributed consensus, transactions, etc
  – theoretical results assume "infinite wait"
    • no time out
    • need approximate solutions
  – Relevance for intermitted communication?
    • e.g. due to reconfiguration
    • sensor sleep mode

• What kind of ad-hoc coordination is possible/needed?