Multi-hop ad-hoc network (MANET) autoconfiguration

Comparative Analysis

Hanga Boros

Overview

- MANET autoconfiguration requirements
- Two recent proposals
  - Jeong model (2004)
  - PACMAN model (2004)
- Comparative analysis of the two models
    - Compatibility with both pro- and re-active routing protocols
    - Required modifications to the existing protocols
    - Overhead
    - Solutions to network merging
  - Re-active AODV
    - Jeong model
    - PACMAN model
  - Pro-active OLSR
    - Jeong model
    - PACMAN model
- Conclusions

MANET autoconfiguration requirements

- Multi-hop (!)
- Self-configuration
- On-going conflict detection (DAD)
  - Network merging, partitioning
- Minimal overhead
- Connection transparency

Model comparison

- PACMAN (2004, Welinger)
  - Initialisation
    Random IP
    No explicit DAD
  - On-going DAD
    Passive DAD (PDAD): Extracting and analysing info from routing
    Algorithms: comparison of sequence numbers, source addresses
    Routing-specific algorithms: LP, NH (OLSR); RwR, 2RoR (AODV)
  - Good performance: Pro-active routing (OLSR)

- Jeong model (2004, Jeong et al)
  - Initialisation
    Random IP
    Query DAD, timeout: AREQ, AREP
  - On-going DAD
    Weak DAD: Extra info (Key and IP tuple) is added
    Uses routing protocol for distribution only
  - Good performance: Re-active routing (AODV)

Research Motivation

- Which of the two is best suited to become the standard?
  - Compatibility with routing protocols
  - Impact on existing protocols
  - Overhead
  - Network merging performance

Re-active AODV (Jeong model)

- Required modifications
  - ICMP types (8-bit) for AREQ, AREP and AERR
  - AODV protocol stack (Strong and Weak flags)
  - AODV routing table
    - Key field (128-bit) to support Weak DAD

- Overhead
  - AREQ, AREP once for each node (ICMP packet size)
  - AERR constantly (ICMP packet size)
    - Also Address Conflict Notification
  - Interface - Key Extension Format constantly
  - Address Change Notification message
  - Store Key field in routing table

- Merging
  - Good performance
  - Except: problem when same key and same IP address occurs

Re-active AODV (PACMAN model)

- Required modifications
  - Eliminate: Intermediate nodes sending RREP messages (Solution: flag)
  - Eliminate: Intermediate nodes can repair connection, no regular RREQs needed
    Modify AODV to synchronize the Sequence Numbers
  - Eliminate: Hello messages are RREPs (Solution: flag)
  - Major problem: Expanding ring-search technique at route setup

- Overhead
  - Address Conflict Notification message
  - Address Change Notification message
  - Store routing info table for calculating max delay of packets

- Merging
  - "Hidden" conflicts are possible
  - Inefficient

Pro-active OLSR (Jeong model)

- (References: Vaidya, 2002 and INRIA, 2005)
- Required modifications
  - No explicit Query DAD is needed
  - For Weak DAD

    Tagging the "hello" messages with Key

    OR Tagging the link state (TC) messages with Key

    OR Send separate packet only with Key
- Overhead
  - Key attached to routing packets or alone (variable size: 32 - 128 bits)
  - Extra beacon to prevent identical tuples of 2 Keys and 2 IP addresses
  - Store Key field in the routing table
  - Store Neighborhood History Table
- Merging
  - Good performance
  - Except for: MPR problem ("hidden" conflicts)

Pro-active OLSR (PACMAN model)

- Required modifications
  - None...
  - For optimisations only: Modify "hello" messages (shorter DAD latency)
- Overhead
  - Address Conflict/Change Notification message
  - Special message to avoid "hidden" conflicts (PDAD-MPR)
  - Neighborhood History Table
  - Store routing info table for calculating max delay of packets
- Merging
  - Sufficient performance
  - Except for: MPR problem ("hidden" conflicts)

Conclusions (1)

- PACMAN
  - + minimal overhead
  - + minimal modification for OLSR
  - - much modification for AODV
  - - unreliable merging detection with AODV (maybe with OLSR too)
  - - far too complex
  - - needs many adjustments to circumstances
  - - dependent on routing protocols and network settings
- Jeong model
  - + reliable merging detection
  - + does not need many adjustments (same DAD is used)
  - + less dependent on routing protocols
  - - somewhat much overhead (especially with OLSR)
  - - overhead depends on Key length

Conclusion (2)

- Which model to choose?
  - Jeong model
    - Reliable conflict detection
    - Works with both pro- and re-active protocols
    - Needs optimisations (especially with OLSR)
  - PACMAN
    - Somewhat unreliable conflict detection
    - Does not reliably work with AODV (if at all)
    - Try DSR instead of AODV
  - Compromise 1: Jeong for re-active and PACMAN for pro-active protocols
  - Compromise 2: Jeong in small networks and PACMAN in larger networks
  - The "wide-spread" routing protocols will dictate the development
  - Other models to come... (E.g. INRIA 2005, DART)

Questions ?