Routing Protocols in Mobile Ad-hoc Networks

Michael Westergaard
mw@daimi.au.dk
Department of Computer Science
University of Aarhus, Denmark
Overview

- A project on routing in mobile ad-hoc networks
- Modules in coloured Petri nets
- Routing Interoperability Protocol
A project on routing in mobile ad-hoc networks
The Project

Participants: Ericsson Denmark A/S, Telebit and CPN Group at University of Aarhus

Project duration: July 2003-December 2005

Project web-page: http://www.daimi.au.dk/CPnets/IPv6/

Executive summary summary: This project deals with the design and validation of routing protocols and other protocols in ad-hoc and mobile networks

The goal was to explore the use of IPv6 in the context of ad-hoc networks using CP-nets
Wireless Communication

W-LAN (e.g. 802.11a/b/g)  Cellular networks

Key characteristics:

- Communication is based on pre-existing (fixed) infrastructure
- No direct communication between mobile nodes
Mobile Ad-hoc Networks

Application areas
- Sensor networks
- Search-and-rescue operations
- Home networking
- Traffic Safety

Challenges
- Mobility and bandwidth
- Power consumption
- Security
- Fully distributed operation

No pre-existing infrastructure and multi-hop communication
A main topic of the project was protocols for integration of fixed core networks and mobile ad-hoc networks.
Sub-projects

1) Specification of mobility and communication scenarios in an Internet-MANET network architecture

2) Specification of an Edge Router Discovery Protocol for mobile ad-hoc networks

3) Model-based prototyping of protocols for Internet-MANET routing with redundant gateways
Modules in coloured Petri nets
Sub-project 3

~ 54 places
~ 40 transitions
~ = big!
~ = incomprehensible?
~ No, because we use modules
A CPN Model

\[
\begin{align*}
\text{if } n=k \\
\text{then } \text{str}^d \\
\text{else } \text{str}
\end{align*}
\]

Transmit

Ack

Receive

Packet

Send

Packet

Send

NextSend

Receive

Ack

Transmit

Packet

Receive

Packet

NextRec

if success
then 1`n
else empty
if n=k
then k+1
else k
if success
then 1`n
else empty

\[
\begin{align*}
\text{INT} \times \text{DATA} \\
\text{DATA} \times \text{str} \\
\text{INT} \times \text{DATA} \\
\text{INT} \times \text{DATA} \\
\text{INT} \times \text{DATA} \\
\end{align*}
\]
A Simpler CPN model

Sender

Receiver

Main

if n=k then k+1 else k

if n=k then str^d else str

if success then 1`(n,d) else empty

if success then 1`(n,d) else empty

if n=k then k+1 else k
Main Module

if success
then 1`n
else empty

if success
then 1`(n,d)
else empty(n,d)

Receiver

Sender

Transmit

Packet

C

INT

B

INTxDATA

A

INTxDATA

(n,d)

Transmit

Ack

D

INT

n

Receiver

Sender
Routing
Interoperability
Protocol
Network Architecture

Possible solutions
- Mobile IP
- Mobile host routes injected by gateways into the core network
- Dynamic DNS and renumbering
Figure 2 shows the approach taken to use CPN models to develop a prototype of the interoperability protocol. A CPN model (lower left of Fig. 2) has been developed by modelling the natural language protocol specification [22] (lower right) of the interoperability protocol. The modelling activity transforms the natural language specification into a formal executable specification represented by the CPN model. The CPN model captures the network architecture depicted in Fig. 1 and the protocol mechanisms of the interoperability protocol, e.g., the periodic transmission of advertisements, the dynamic updates of the DNS database, and traffic flows between hosts in the core network and nodes in the ad-hoc network.
Figure 2 shows the approach taken to use CPN models to develop a prototype of the interoperability protocol. A CPN model (lower left of Fig. 2) has been developed by modelling the natural language protocol specification [22] (lower right) of the interoperability protocol. The modelling activity transforms the natural language specification into a formal executable specification represented by the CPN model. The CPN model captures the network architecture depicted in Fig. 1 and the protocol mechanisms of the interoperability protocol, e.g., the periodic transmission of advertisements, the dynamic updates of the DNS database, and traffic flows between hosts in the core network and nodes in the ad-hoc network.
Scenario
Router Advertisements

AHN(3) -> 3ffe:100:3:405::3
AHN(4) -> 3ffe:100:3:405::4
AHN(5) -> 3ffe:100:3:406::5

Gateway 1
3ffe:100:3:401::3
3ffe:100:3:405::1

Gateway 2
3ffe:100:3:401::4
3ffe:100:3:406::1

DNS Server
3ffe:100:3:401::1

Ad-hoc Node 3
3ffe:100:3:405::3

Ad-hoc Node 4
3ffe:100:3:405::4

Ad-hoc Node 5
3ffe:100:3:406::5
Sending Data
Mobility & DNS Update

- AHN (3) -> 3ffe:100:3:405::3
- AHN (4) -> 3ffe:100:3:405::4
- AHN (5) -> 3ffe:100:3:406::5

- DNS request/reply/update
- Gateway advertisement
- Data packet

- DNS Server
  - 3ffe:100:3:401::1

- Host 1
  - 3ffe:100:3:401::2

- Gateway 1
  - 3ffe:100:3:401::3
  - 3ffe:100:3:405::1

- Gateway 2
  - 3ffe:100:3:401::4
  - 3ffe:100:3:406::1

- Ad-hoc Node 3
  - 3ffe:100:3:405::3
  - 3ffe:100:3:406::3

- Ad-hoc Node 4
  - 3ffe:100:3:405::4
  - 3ffe:100:3:406::4

- Ad-hoc Node 5
  - 3ffe:100:3:405::5
  - 3ffe:100:3:406::5
Basic Operation
1 `("3ffe:100:3:401::3","3ffe:100:3:405::1","3ffe:100:3:406::")

1 `(ROUTING,{src="3ffe:100:3:401::2",dest="3ffe:100:3:401::1",cont=DNS_REQ("AHN(3)"))}

1 `(RECEIVE("AHN(4)"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::1")))++

1 `(FLOODING("3ffe:100:3:405::1"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::1")))`
Ad-hoc Node

1 `(RECEIVE("AHN(3)"),{src="3ffe:100:3:405::3","3ffe:100:3:405::1","3ffe:100:3:405::")})++
1 `(AHN(4),[{"3ffe:100:3:405::4","3ffe:100:3:405::1","3ffe:100:3:405::")})++
1 `(AHN(5),[{"3ffe:100:3:406::5","3ffe:100:3:406::1","3ffe:100:3:406::"},{"3ffe:100:3:405::5","3ffe:100:3:405::1","3ffe:100:3:405::")}])

1 `(RECEIVE("AHN(4)"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV("3ffe:100:3:401::1","3ffe:100:3:405::")})++
1 `(FLOODING("3ffe:100:3:405::1"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV("3ffe:100:3:401::1","3ffe:100:3:405::")})++

1 `(AHN(3),[("3ffe:100:3:405::3","3ffe:100:3:405::1","3ffe:100:3:405::")])++
1 `(AHN(4),[("3ffe:100:3:405::4","3ffe:100:3:405::1","3ffe:100:3:405::")])++
1 `(AHN(5),[("3ffe:100:3:406::5","3ffe:100:3:406::1","3ffe:100:3:406::"),("3ffe:100:3:405::5","3ffe:100:3:405::1","3ffe:100:3:405::")])
Advantages of Model-based Prototypes

- Easier to control and reproduce scenarios
- Implementation details can be abstracted away
- Setup of physical network equipment is not required
- Larger scenarios can be investigated
Advantages of Integration of CP-nets with Animation

- Behaviour is as defined by the formal model
- Knowledge of the formal modelling language is not required
- Presentation for military leaders is possible
- Validation that the implemented prototype corresponds to the specification