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

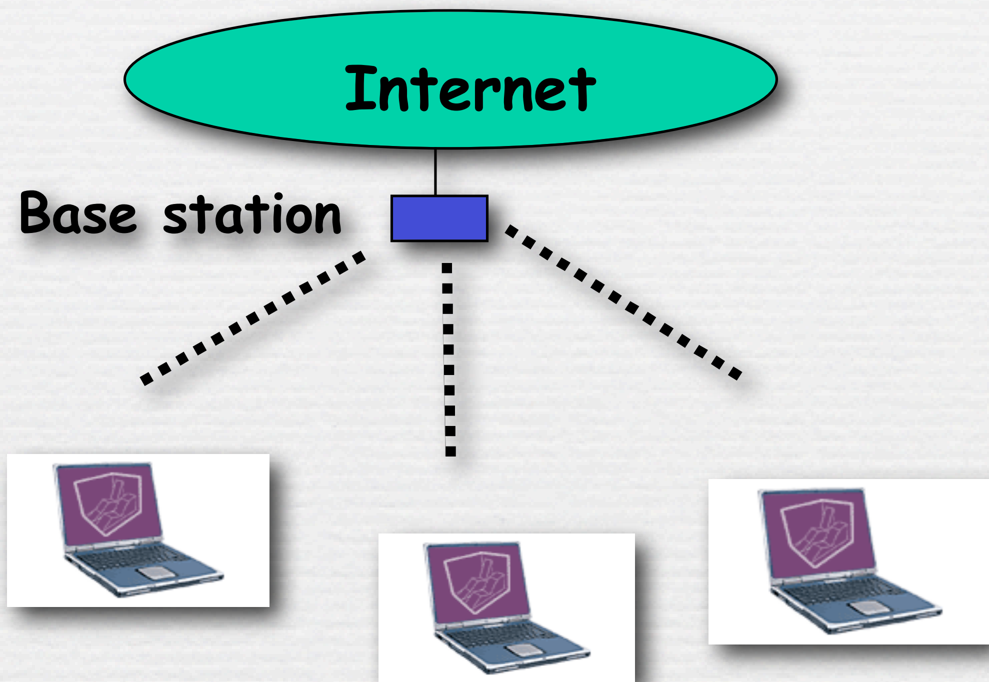
A project on routing in mobile ad-hoc networks

The Project

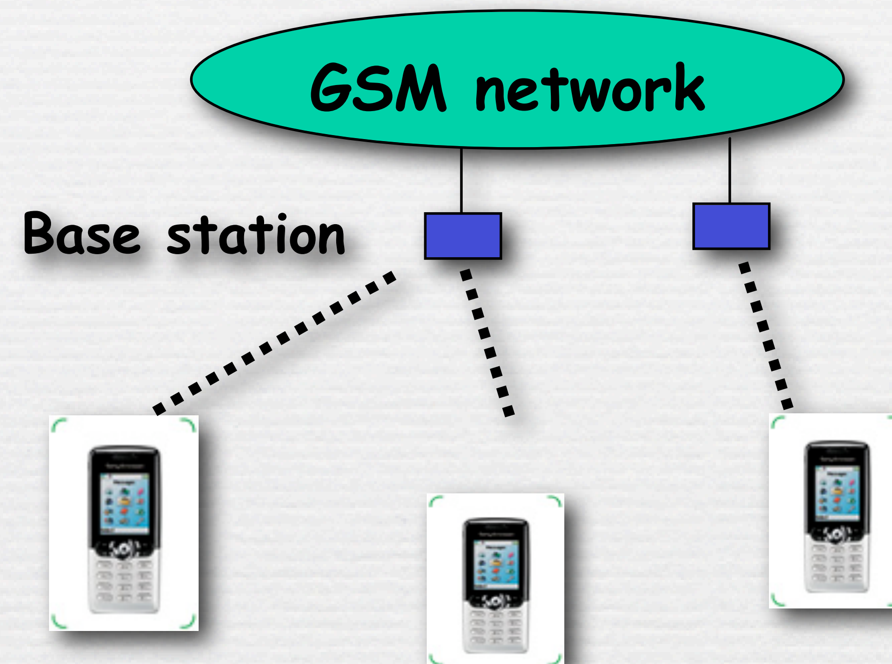
- Participants: Ericsson Denmark A/S, Telebit and CPN Group at University of Aarhus
- Project duration: July 2003-December 2005
- Project web-page: http://www.daimi.au.dk/CPnets/IPv6/
- Executive summary summary: *This project deals with the design and validation of routing protocols and other protocols 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)



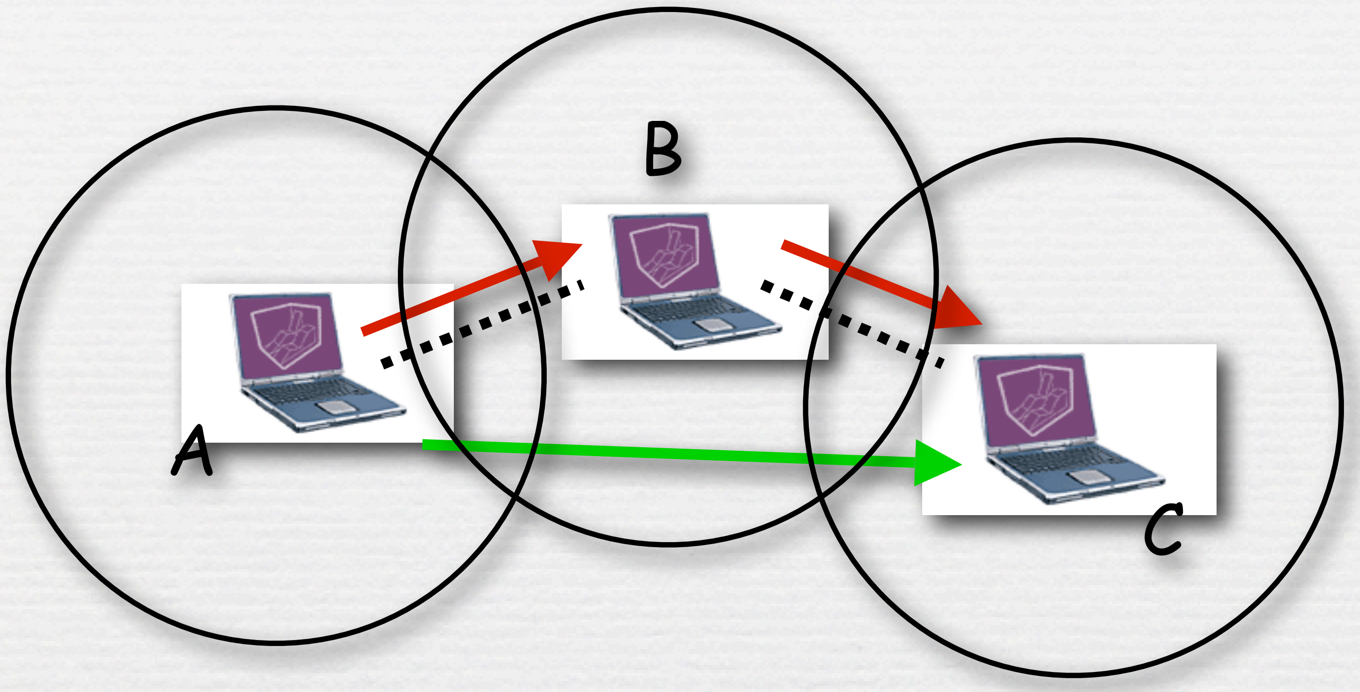
Cellular networks



Key characteristics:

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

Mobile Ad-hoc Networks



No pre-existing infrastructure and multi-hop communication

Application areas

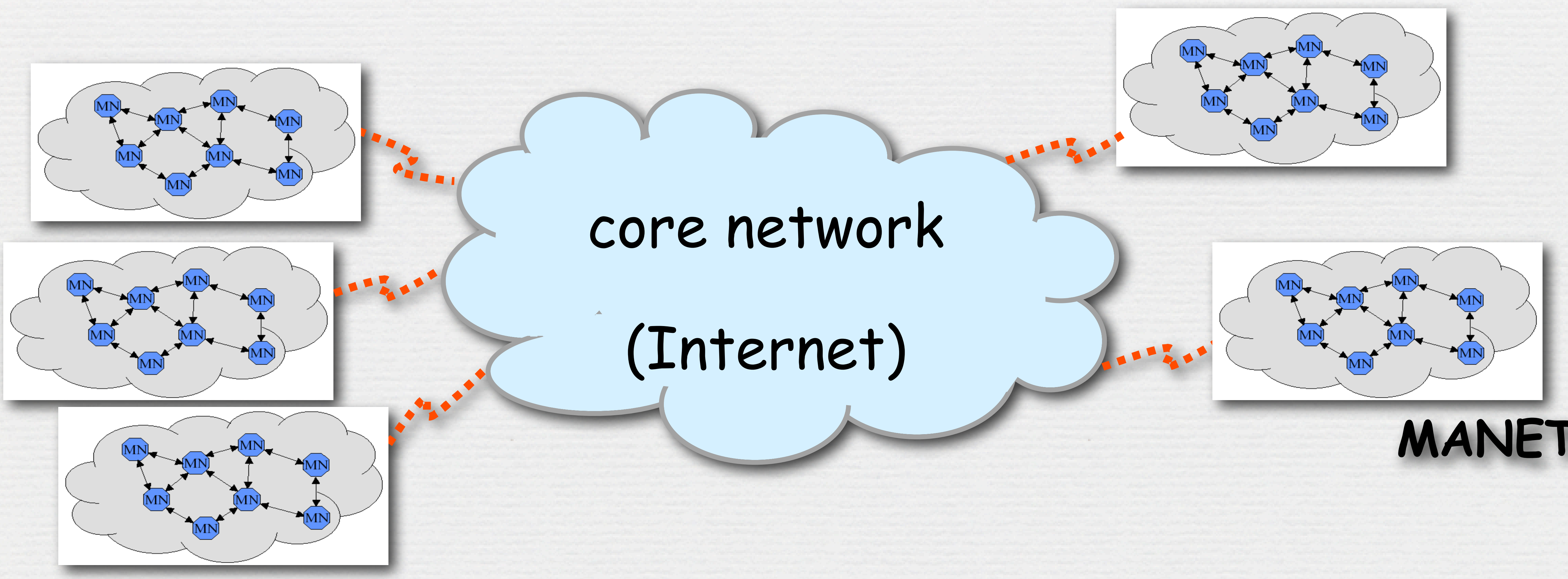
- Sensor networks
- Search-and-rescue operations
- Home networking
- Traffic Safety

Challenges

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

Hybrid Network Architecture

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

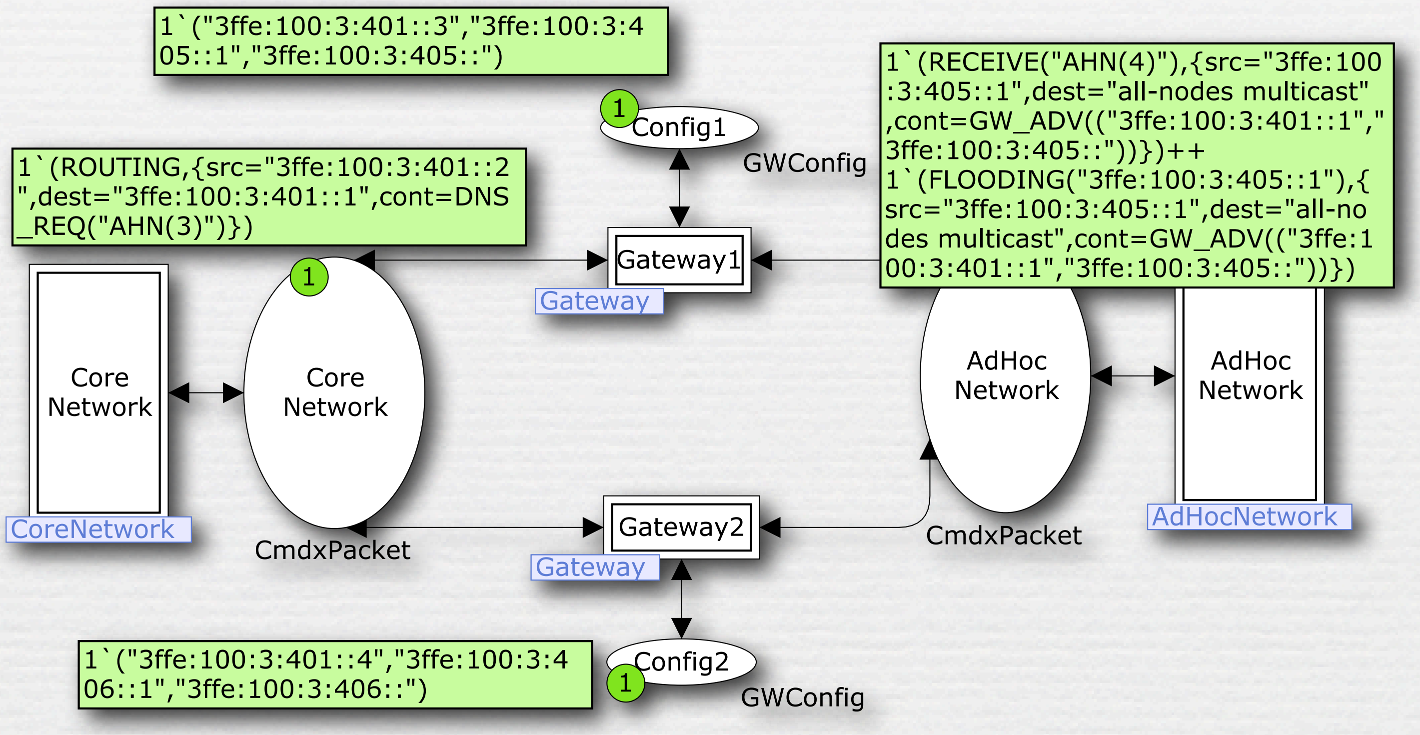
2) Specification of an Edge Router Discovery Protocol for mobile ad-hoc networks

3) Model-based prototyping of protocols for Internet-MANET routing with redundant gateways

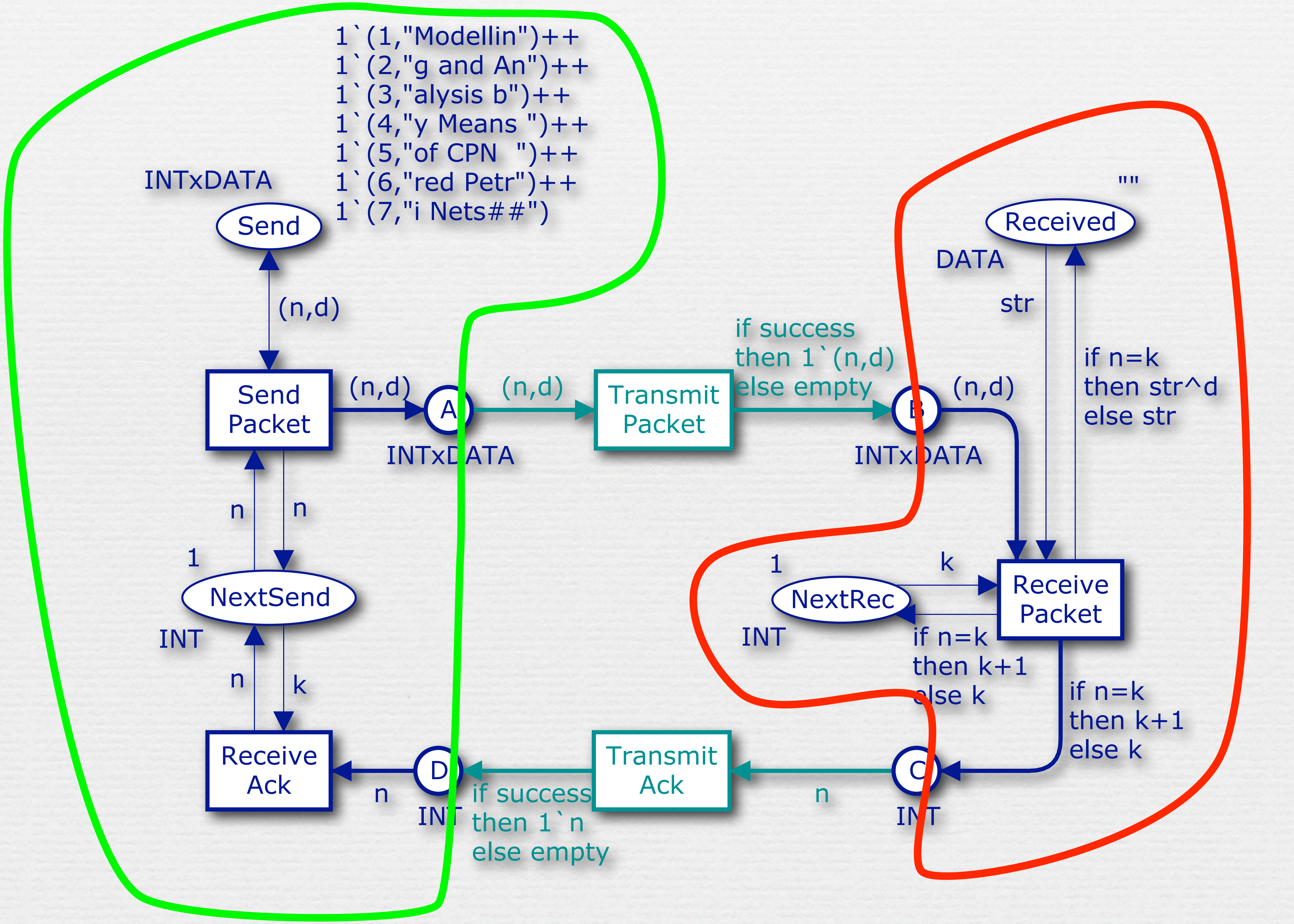
Modules in coloured Petri nets

Sub-project 3

- 54 places
- 40 transitions
- = big!
- = incomprehensible?
  - No, because we use modules

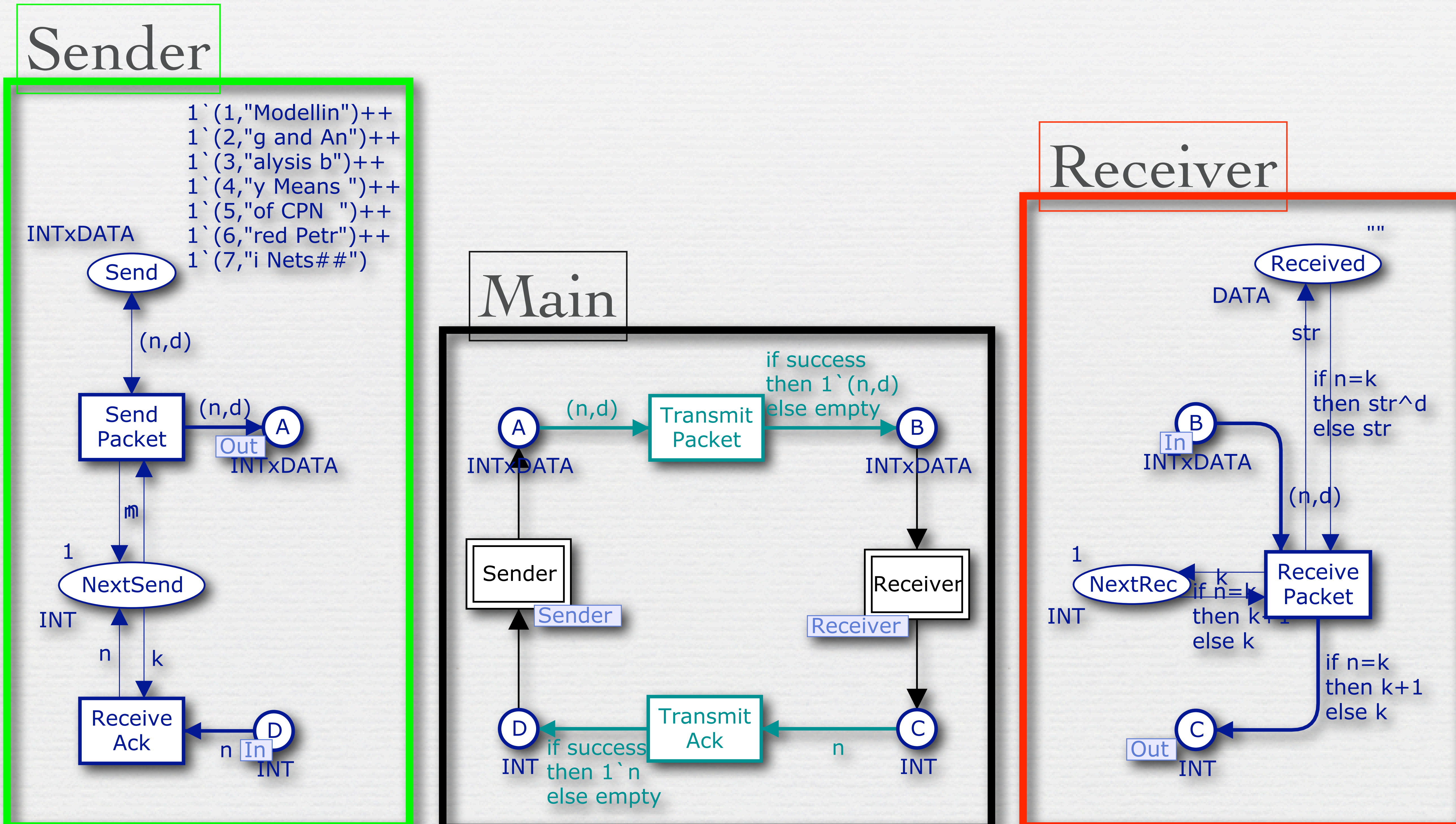


A CPN Model

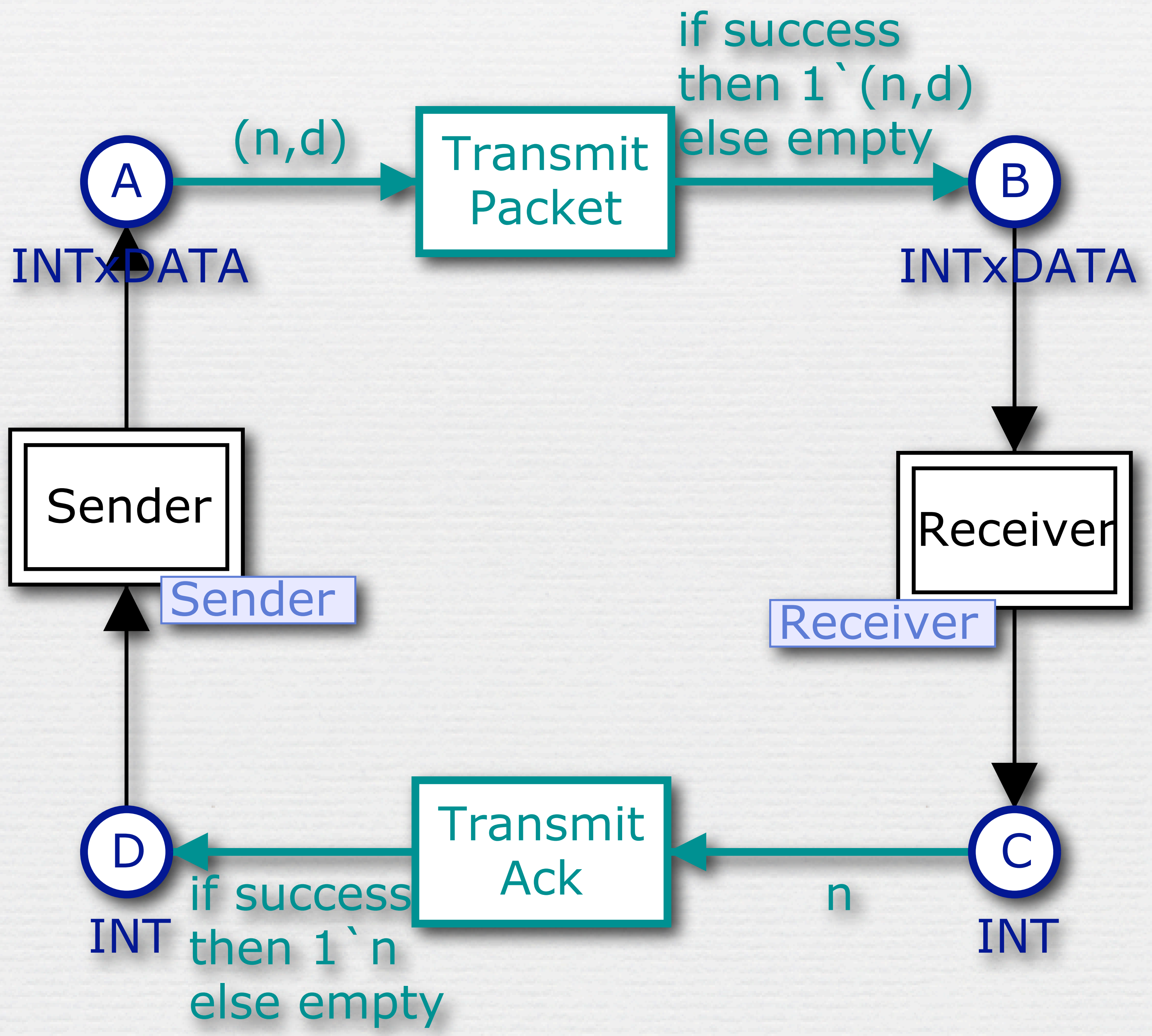


1`(1,"Modellin")++
1`(2,"g and An")++
1`(3,"alysis b")++
1`(4,"y Means ")++
1`(5,"of CPN ")++
1`(6,"red Petr")++
1`(7,"i Nets##")
INTxDATA
Send
(n,d)
Send Packet
(n,d)
A
(n,d)
Transmit Packet
if success then 1`(n,d) else empty
B
(n,d)
INTxDATA
INTxDATA
n
n
1
NextSend
INT
n
k
Receive Ack
n
D
INT
if success then 1`n else empty
Transmit Ack
n
C
INT
""
Received
DATA
str
if n=k then str^d else str
1
k
NextRec
INT
Receive Packet
if n=k then k+1 else k
if n=k then k+1 else k

A Simpler CPN model



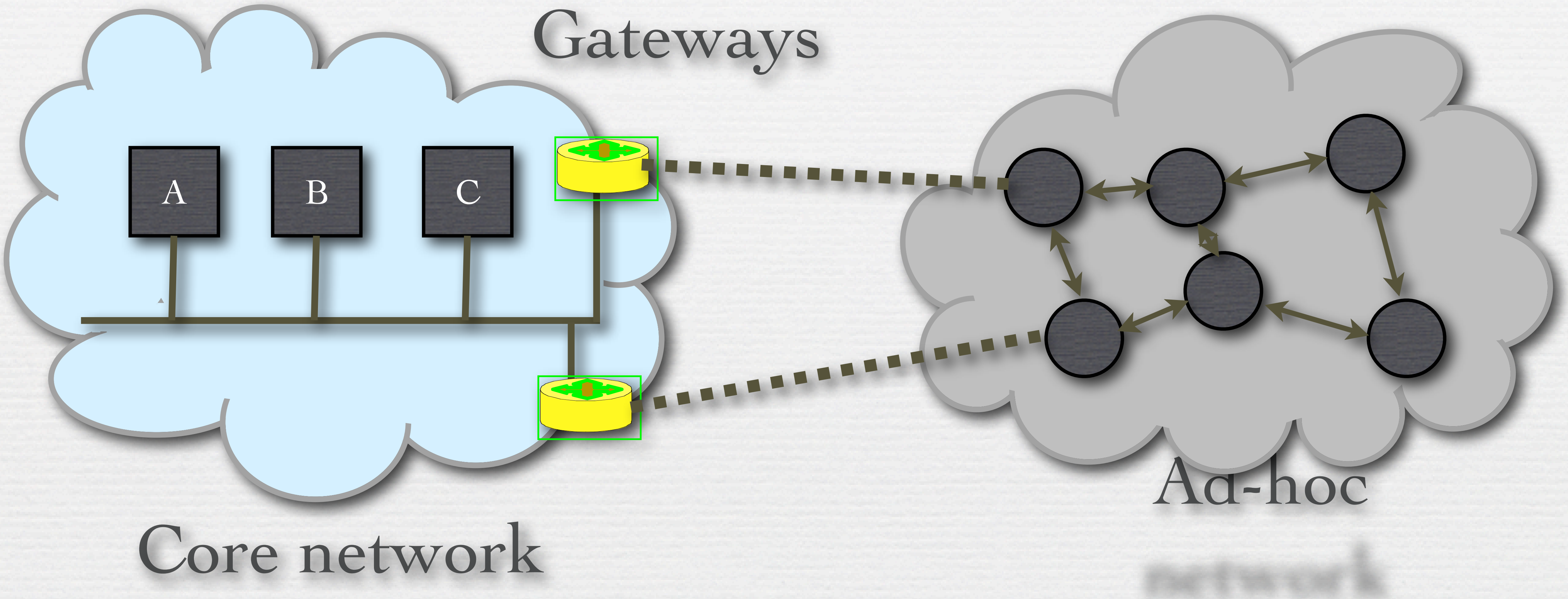
Main Module



if success
then 1`(n,d)
else empty
(n,d)
Transmit
Packet
A
B
INTxDATA
INTxDATA
Sender
Sender
Receiver
Receiver
Transmit
Ack
D
C
if success
then 1`n
else empty
n
INT
INT

Routing Interoperability Protocol

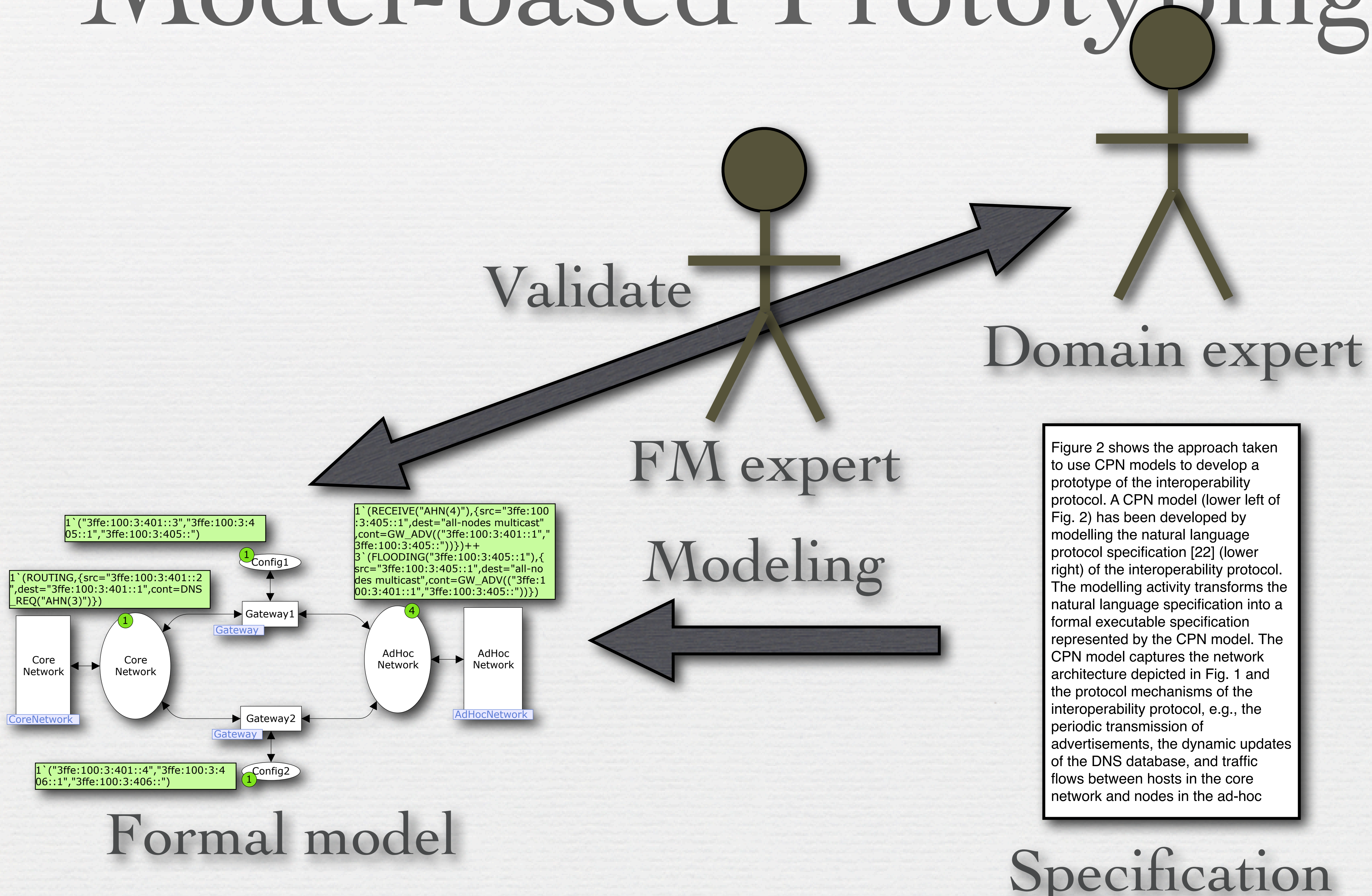
Network Architecture



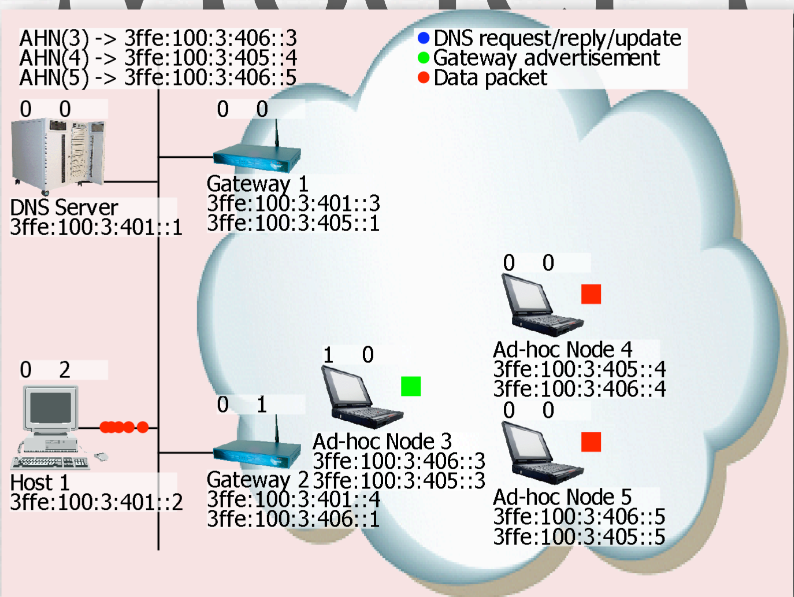
Possible solutions

- Mobile IP
- Mobile host routes injected by gateways into the core network
- Dynamic DNS and renumbering

Model-based Prototyping



Model-based Prototyping



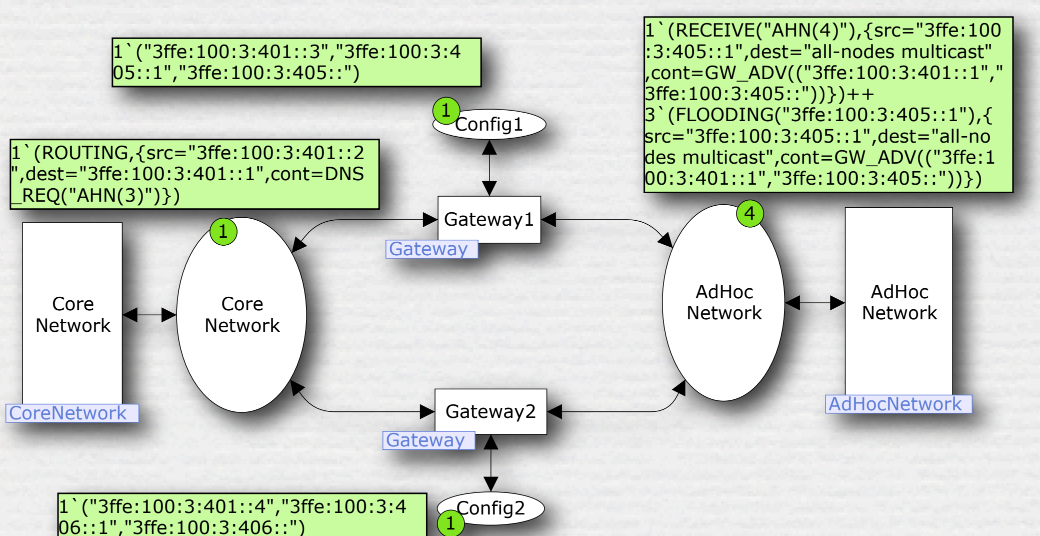
Explore and interact

Domain expert

Animation

FM expert

Modeling

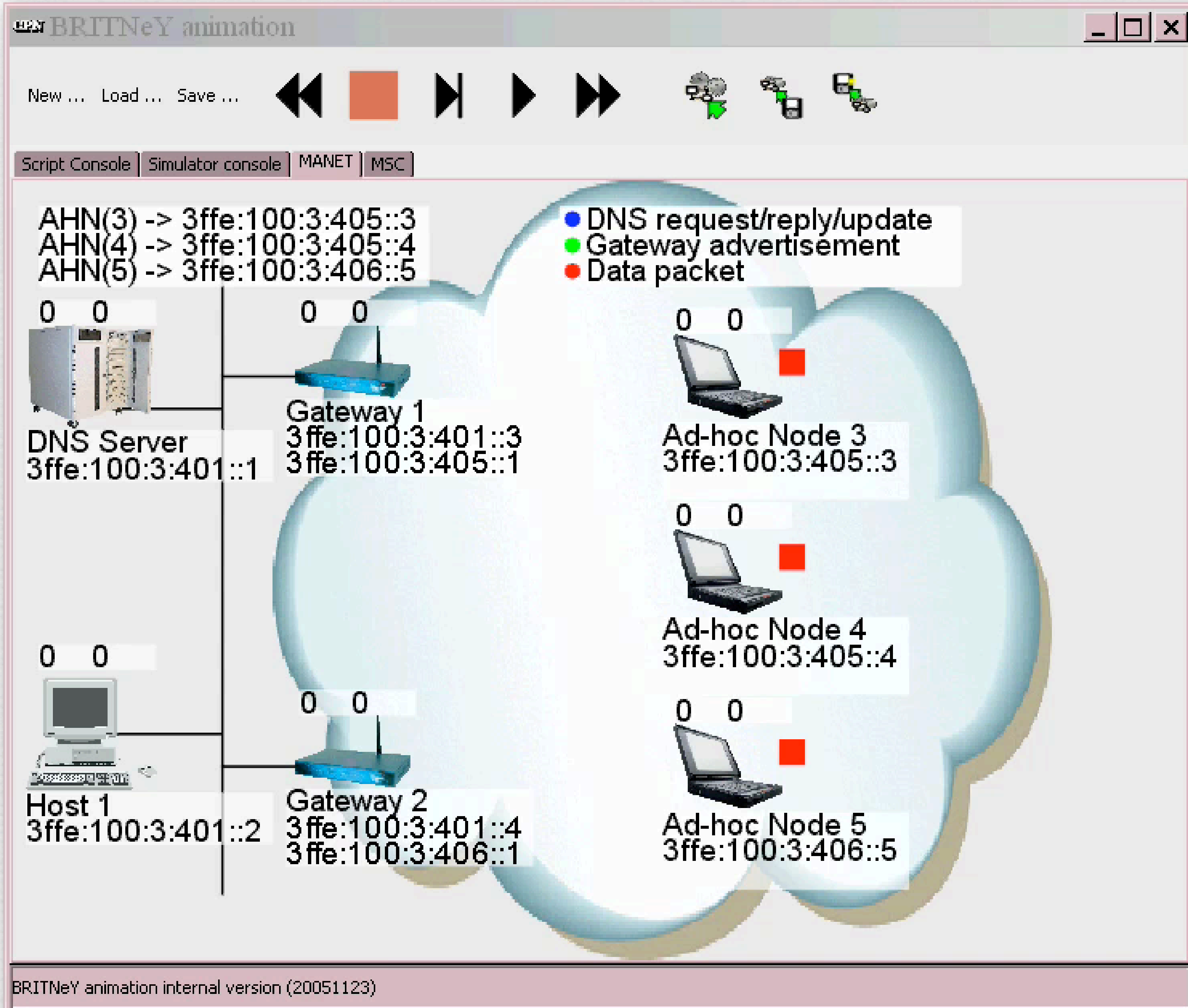


Formal model

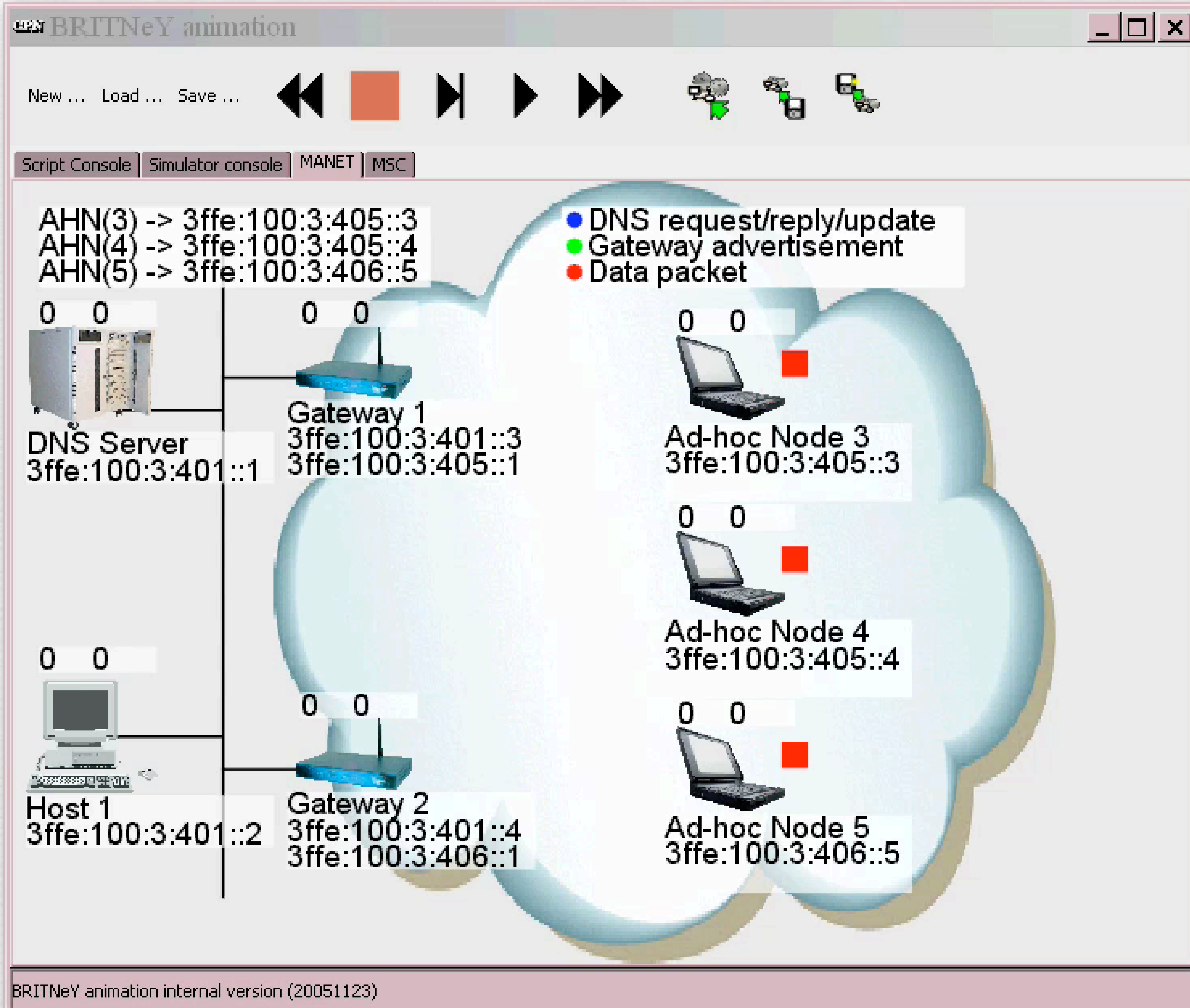
Figure 2 shows the approach taken to use CPN models to develop a prototype of the interoperability protocol. A CPN model (lower left of Fig. 2) has been developed by modelling the natural language protocol specification [22] (lower right) of the interoperability protocol. The modelling activity transforms 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

Specification

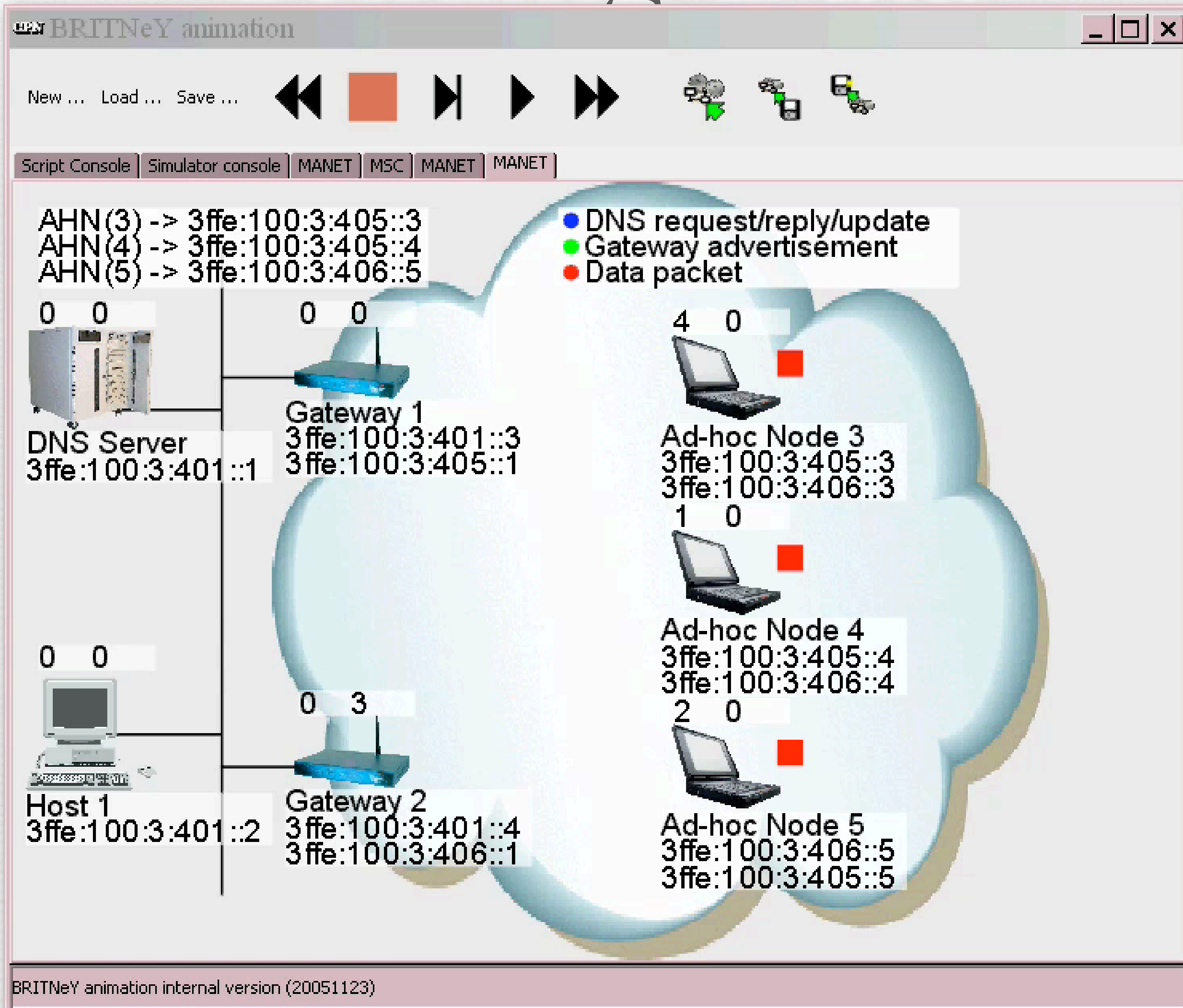
Scenario


BRITNeY animation
New ...
Load ...
Save ...
Script Console
Simulator console
MANET
MSC
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
0 0
DNS Server
3ffe:100:3:401::1
0 0
Gateway 1
3ffe:100:3:401::3
3ffe:100:3:405::1
0 0
Ad-hoc Node 3
3ffe:100:3:405::3
0 0
Ad-hoc Node 4
3ffe:100:3:405::4
0 0
Host 1
3ffe:100:3:401::2
0 0
Gateway 2
3ffe:100:3:401::4
3ffe:100:3:406::1
0 0
Ad-hoc Node 5
3ffe:100:3:406::5
BRITNeY animation internal version (20051123)

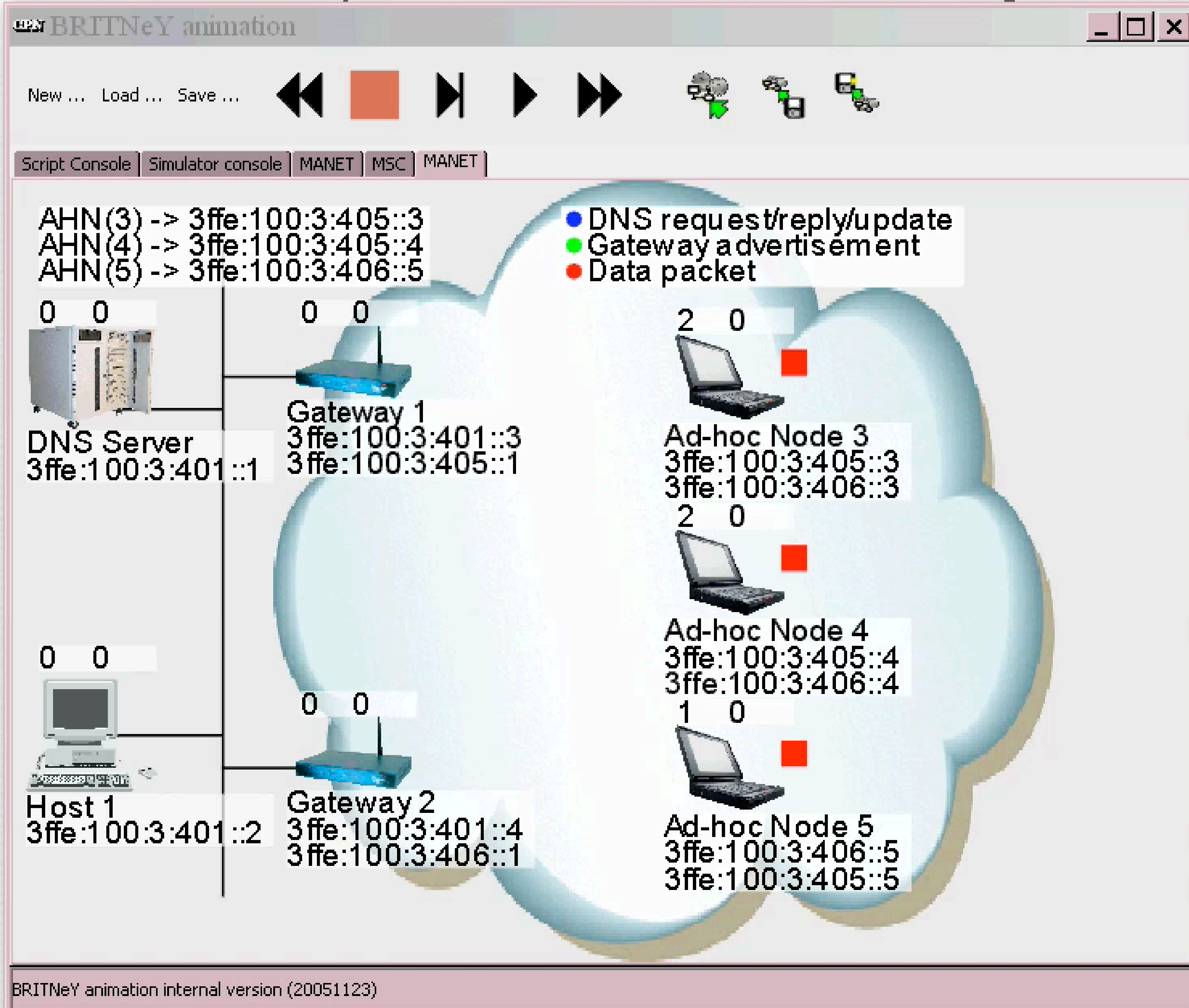
Router Advertisements



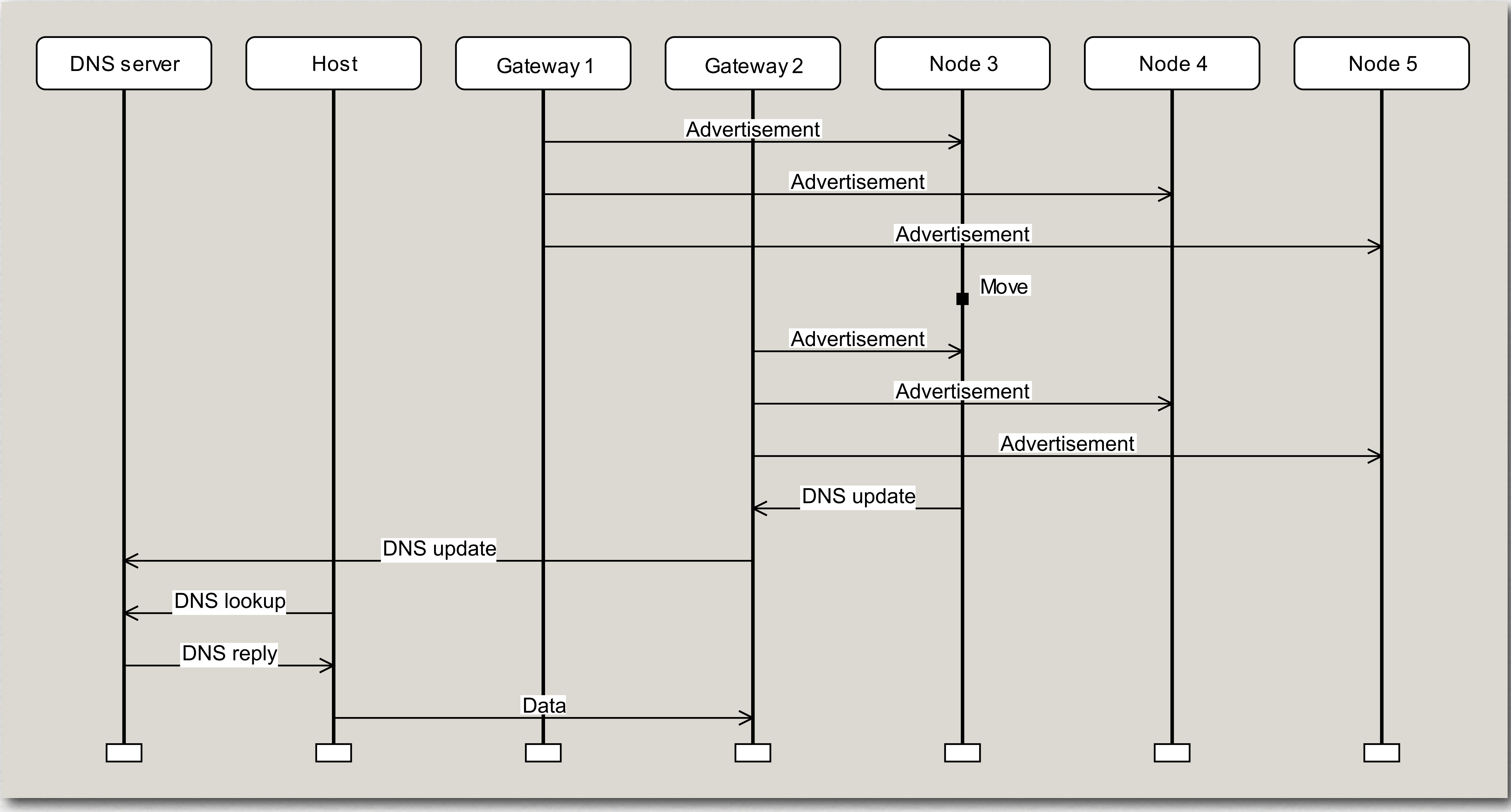
Sending Data



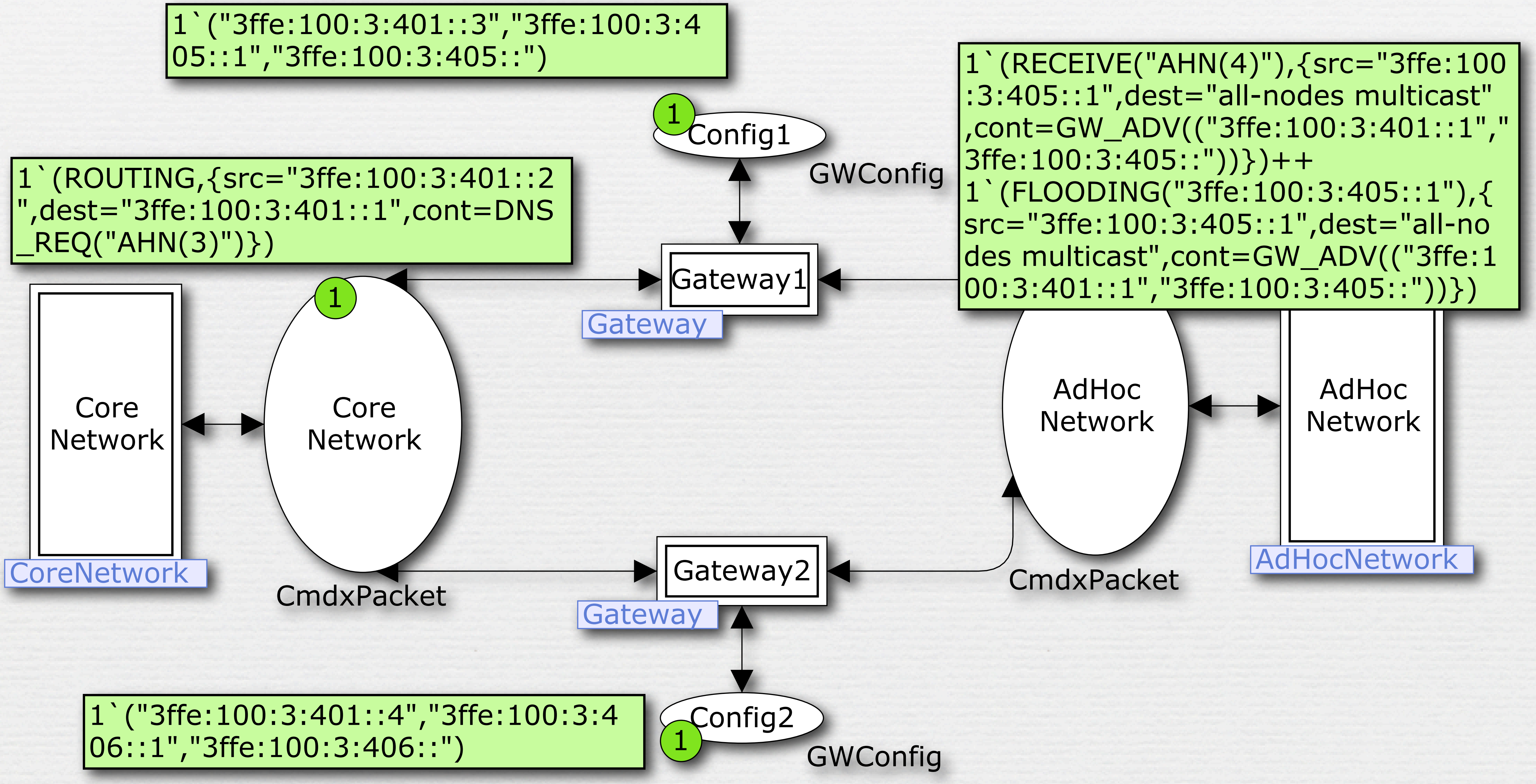
Mobility & DNS Update



Basic Operation

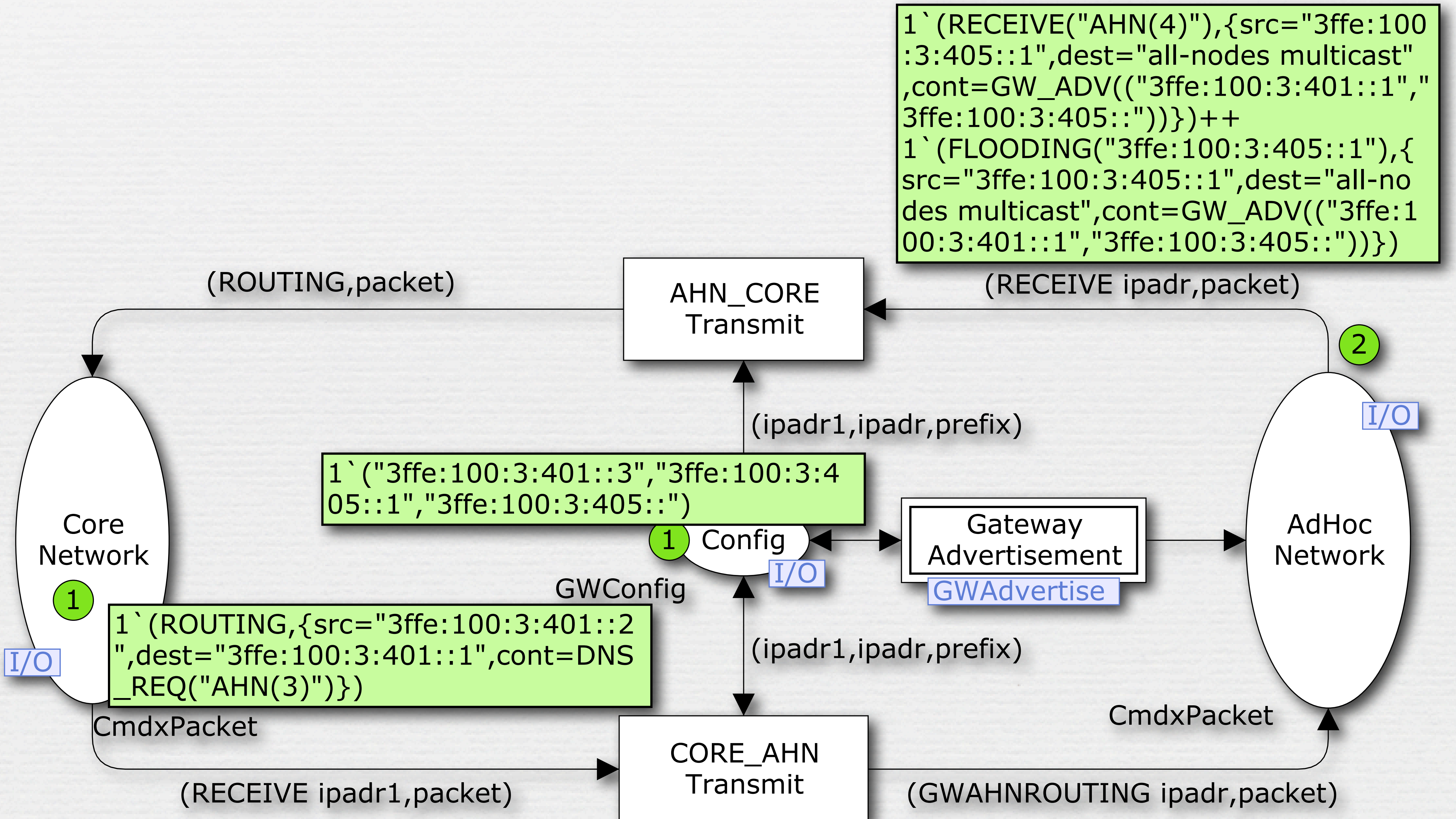


Model



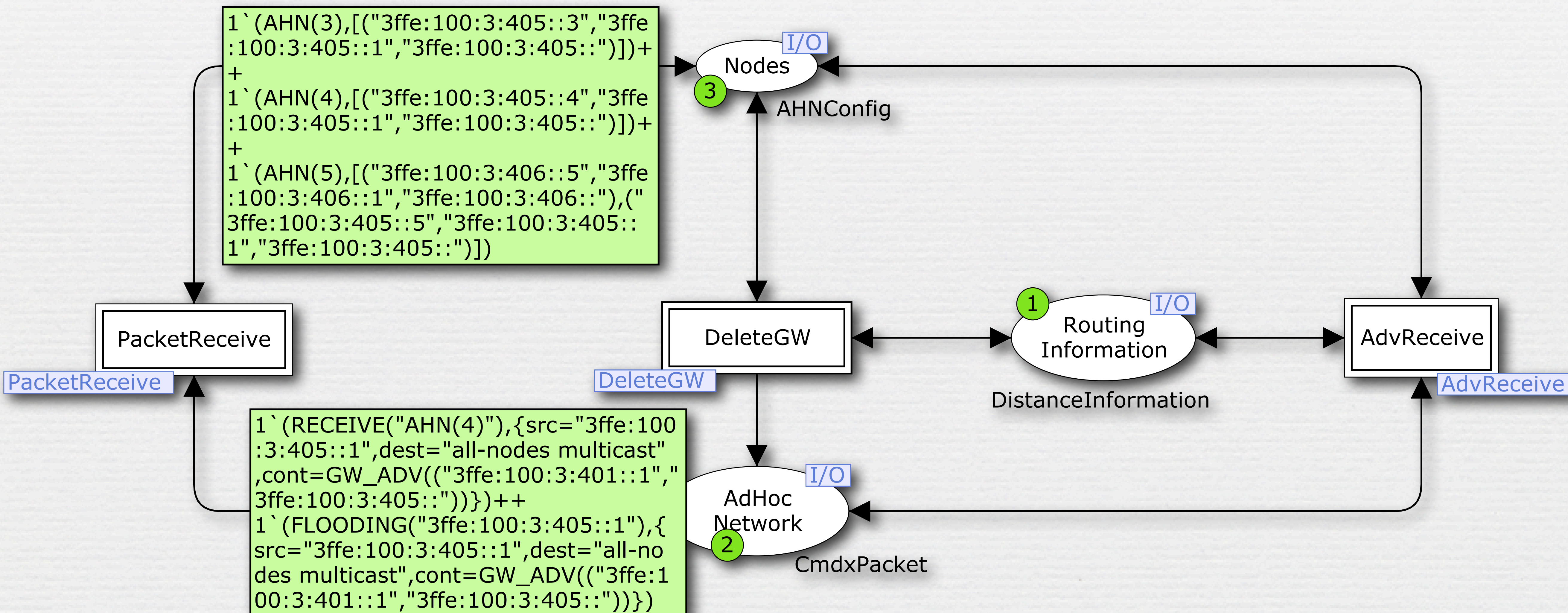
1`("3ffe:100:3:401::3","3ffe:100:3:405::1","3ffe:100:3:405::")
1
Config1
GWConfig
1`(ROUTING,{src="3ffe:100:3:401::2",dest="3ffe:100:3:401::1",cont=DNS_REQ("AHN(3)")})
1`(RECEIVE("AHN(4)"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})++
1`(FLOODING("3ffe:100:3:405::1"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})
Gateway1
Gateway
1
Core Network
CoreNetwork
Core Network
CmdxPacket
AdHoc Network
CmdxPacket
AdHoc Network
AdHocNetwork
Gateway2
Gateway
1`("3ffe:100:3:401::4","3ffe:100:3:406::1","3ffe:100:3:406::")
Config2
1
GWConfig

Gateway

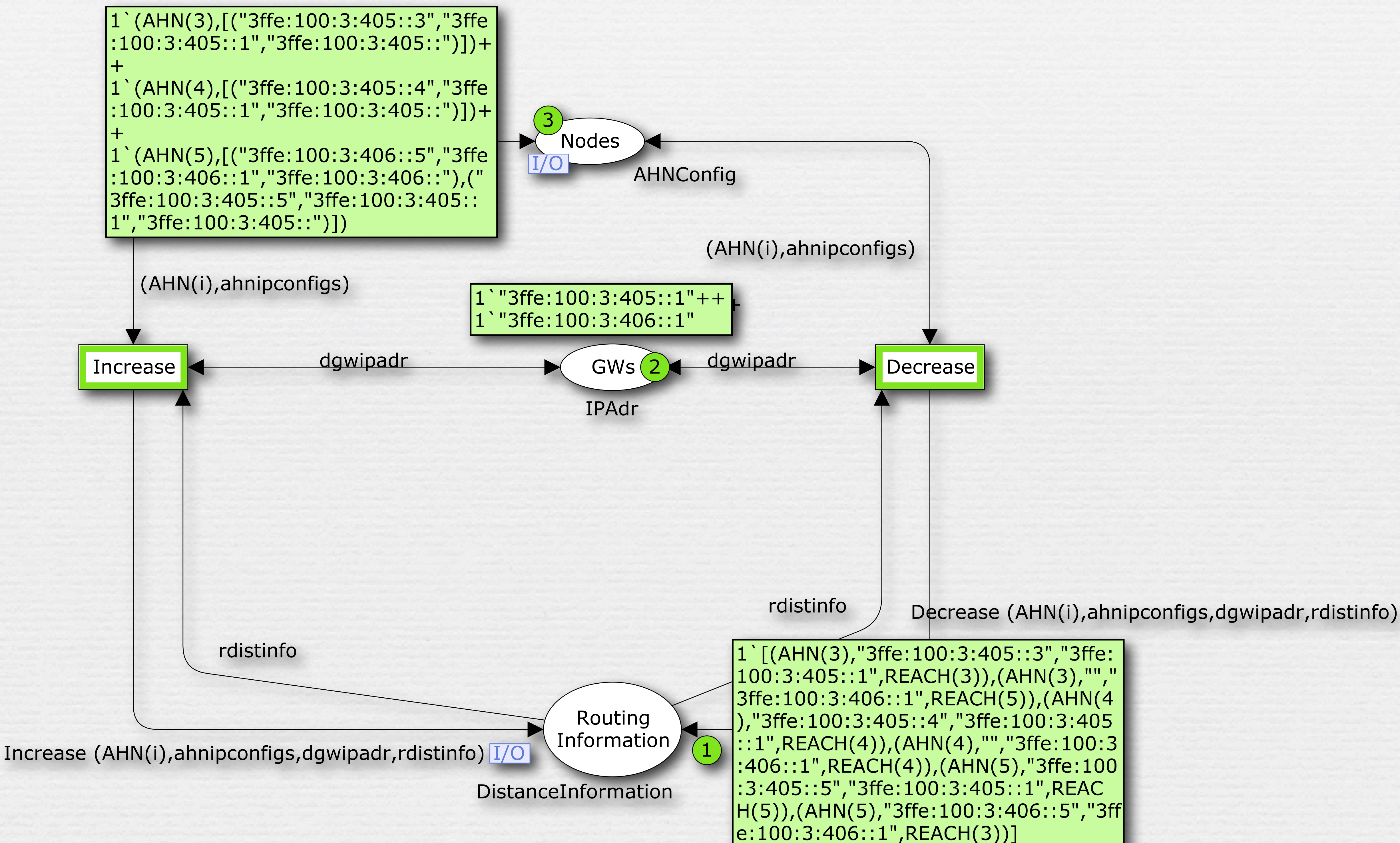


1`(RECEIVE("AHN(4)"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})++
1`(FLOODING("3ffe:100:3:405::1"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})
(ROUTING,packet)
AHN_CORE Transmit
(RECEIVE ipadr,packet)
2
I/O
(ipadr1,ipadr,prefix)
1`("3ffe:100:3:401::3","3ffe:100:3:405::1","3ffe:100:3:405::")
Core Network
1
Config
I/O
GWConfig
Gateway Advertisement
GWAdvertise
AdHoc Network
1
1`(ROUTING,{src="3ffe:100:3:401::2",dest="3ffe:100:3:401::1",cont=DNS_REQ("AHN(3)")})
I/O
(ipadr1,ipadr,prefix)
CmdxPacket
CmdxPacket
CORE_AHN Transmit
(RECEIVE ipadr1,packet)
(GWAHNROUTING ipadr,packet)

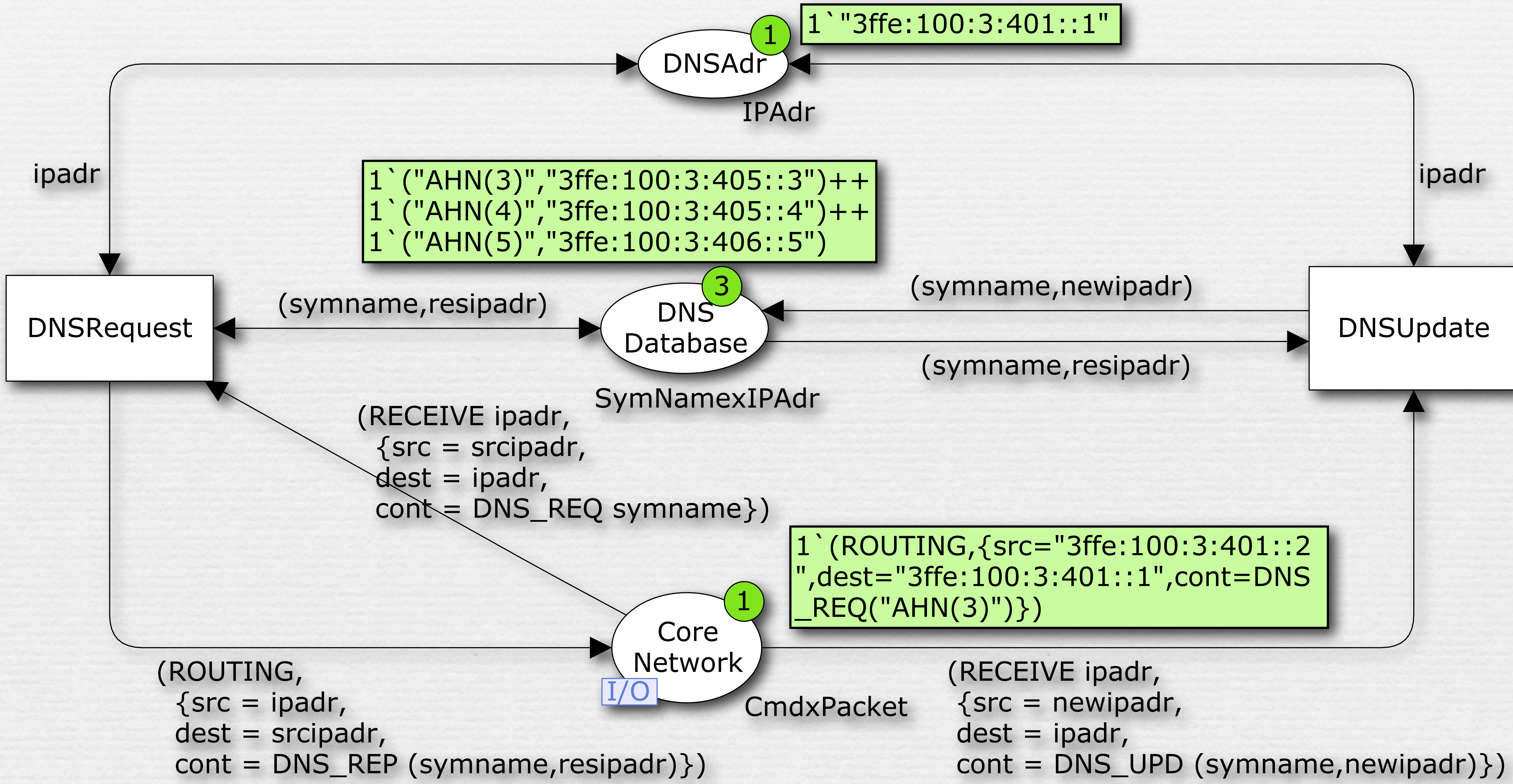
Ad-hoc Node


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::")])
I/O
Nodes
3
AHNConfig
PacketReceive
PacketReceive
DeleteGW
DeleteGW
1
I/O
Routing Information
DistanceInformation
AdvReceive
AdvReceive
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::"))})
I/O
AdHoc Network
2
CmdxPacket

Mobility


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::")])
3
Nodes
I/O
AHNConfig
(AHN(i),ahnipconfigs)
(AHN(i),ahnipconfigs)
1`"3ffe:100:3:405::1"++
1`"3ffe:100:3:406::1"
Increase
dgwipadr
GWs
2
dgwipadr
Decrease
IPAdr
rdistinfo
Decrease (AHN(i),ahnipconfigs,dgwipadr,rdistinfo)
rdistinfo
1`[(AHN(3),"3ffe:100:3:405::3","3ffe:100:3:405::1",REACH(3)),(AHN(3),"","3ffe:100:3:406::1",REACH(5)),(AHN(4),"3ffe:100:3:405::4","3ffe:100:3:405::1",REACH(4)),(AHN(4),"","3ffe:100:3:406::1",REACH(4)),(AHN(5),"3ffe:100:3:405::5","3ffe:100:3:405::1",REACH(5)),(AHN(5),"3ffe:100:3:406::5","3ffe:100:3:406::1",REACH(3))]
Routing Information
1
Increase (AHN(i),ahnipconfigs,dgwipadr,rdistinfo)
I/O
DistanceInformation

DNS Server


1`"3ffe:100:3:401::1"
1
DNSAdr
IPAdr
ipadr
ipadr
1`("AHN(3)","3ffe:100:3:405::3")++
1`("AHN(4)","3ffe:100:3:405::4")++
1`("AHN(5)","3ffe:100:3:406::5")
3
DNSRequest
(symname,resipadr)
DNS Database
(symname,newipadr)
(symname,resipadr)
DNSUpdate
SymNamexIPAdr
(RECEIVE ipadr,
{src = srcipadr,
dest = ipadr,
cont = DNS_REQ symname})
1`(ROUTING,{src="3ffe:100:3:401::2
",dest="3ffe:100:3:401::1",cont=DNS
_REQ("AHN(3)")})
1
Core Network
I/O
CmdxPacket
(ROUTING,
{src = ipadr,
dest = srcipadr,
cont = DNS_REP (symname,resipadr)})
(RECEIVE ipadr,
{src = newipadr,
dest = ipadr,
cont = DNS_UPD (symname,newipadr)})

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