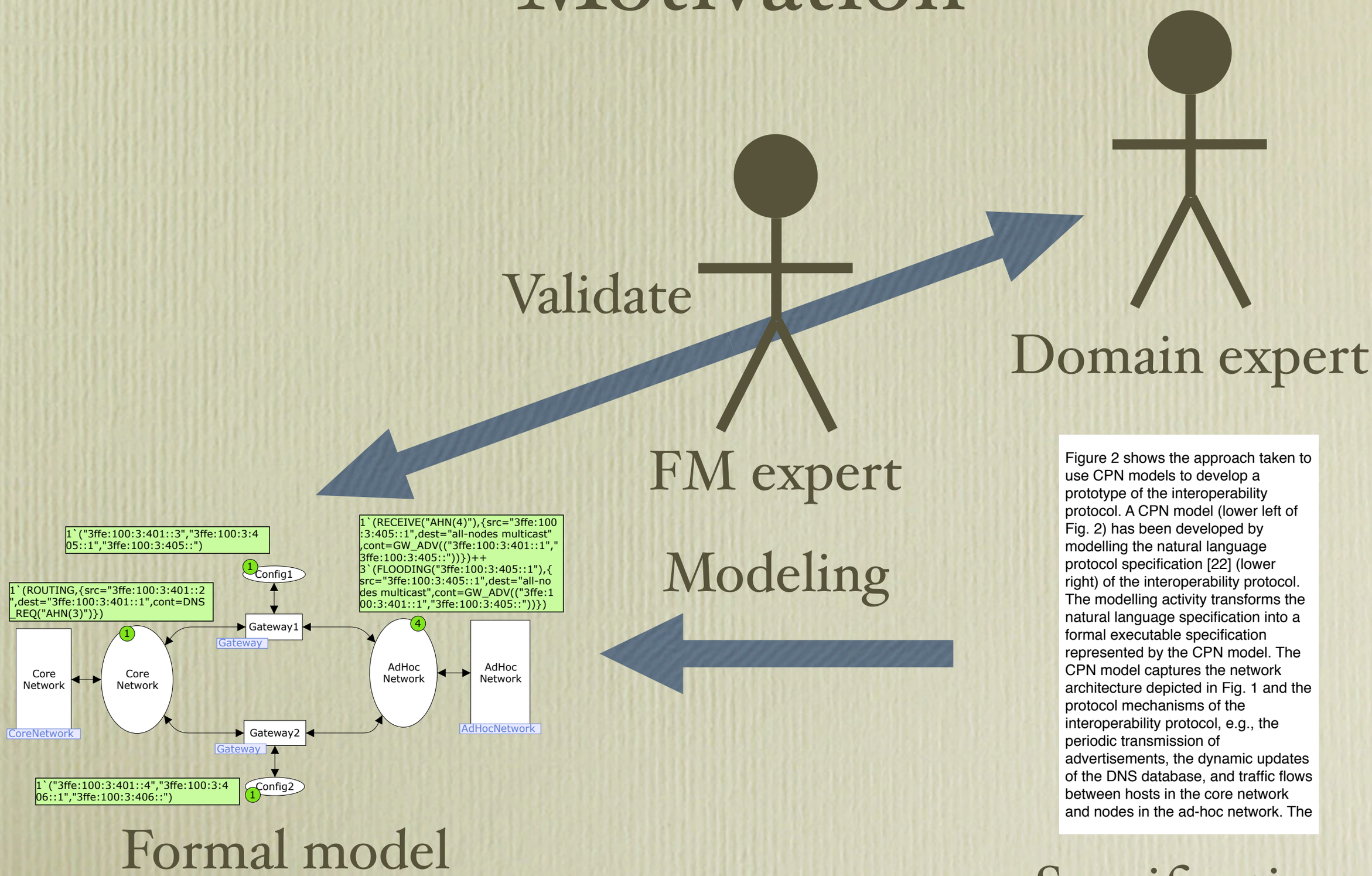
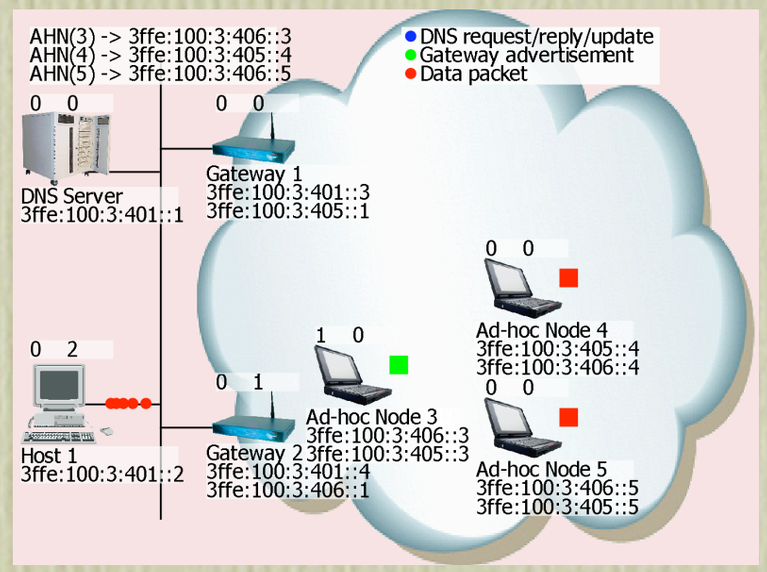
# Model-based Prototyping of an Interoperability Protocol for Mobile Ad-hoc Networks

L.M. Kristensen[1], M. Westergaard[1], and P.C. Nørgaard[2]
[1]Department of Computer Science,
University of Aarhus, Denmark
[2]Ericsson Danmark A/S, Telebit, Denmark

# Motivation

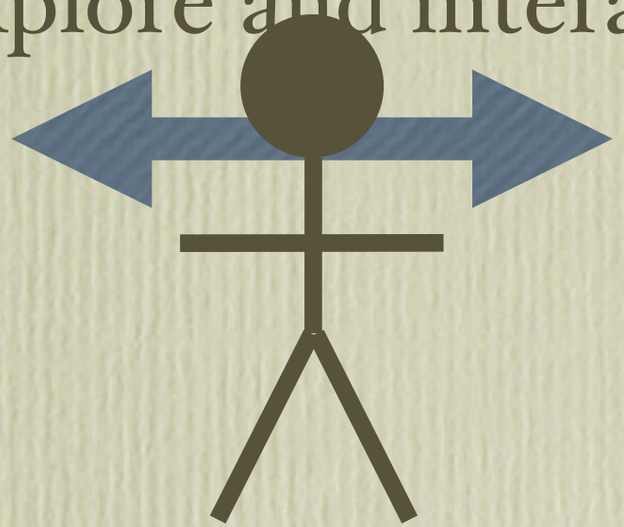

Validate
Domain expert
FM expert
Modeling
1`("3ffe:100:3:401::3","3ffe:100:3:405::1","3ffe:100:3:405::")
1`(RECEIVE("AHN(4)"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})++ 3`(FLOODING("3ffe:100:3:405::1"),{src="3ffe:100:3:405::1",dest="all-nodes multicast",cont=GW_ADV(("3ffe:100:3:401::1","3ffe:100:3:405::"))})
1`(ROUTING,{src="3ffe:100:3:401::2",dest="3ffe:100:3:401::1",cont=DNS_REQ("AHN(3)")})
Config1
Gateway1
Gateway
Core Network
Core Network
CoreNetwork
AdHoc Network
AdHoc Network
AdHocNetwork
Gateway2
Gateway
Config2
1`("3ffe:100:3:401::4","3ffe:100:3:406::1","3ffe:100:3:406::")
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 network. The
Specification

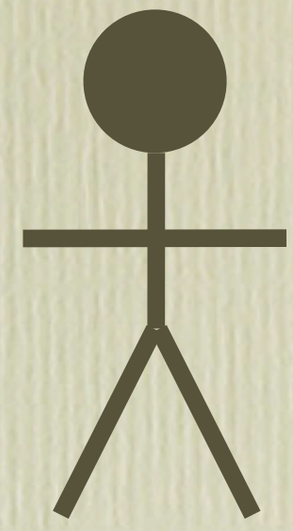
# Methodology




Animation

Explore and interact

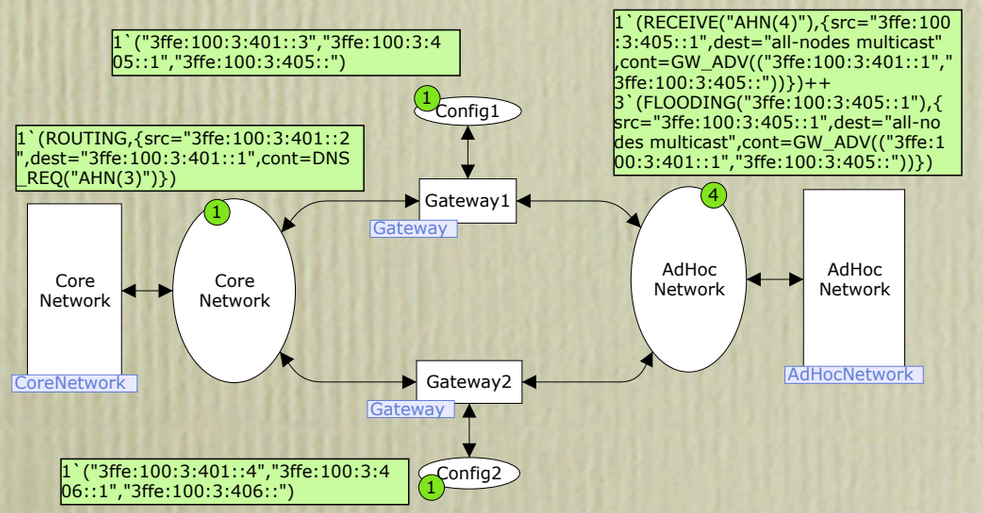




Domain expert

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 network. The

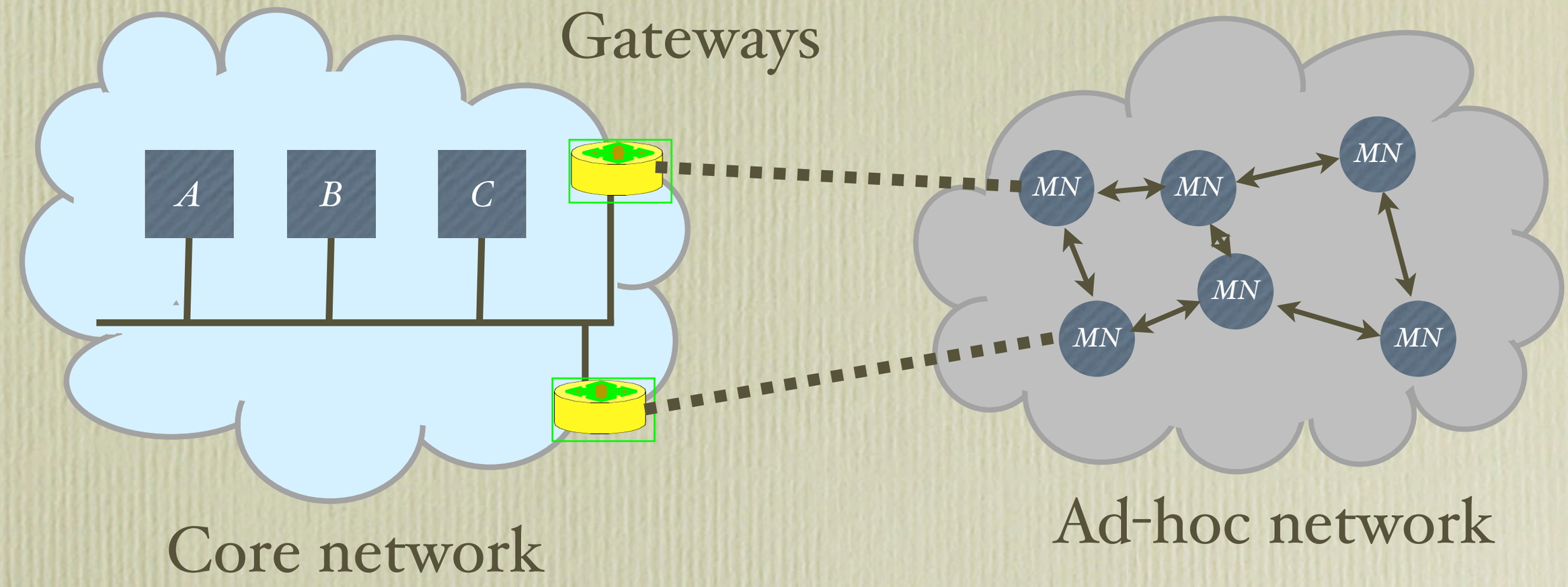
Specification

# Overview

- The project
- A solution: dynamic DNS and renumbering
- A formal model prototype
- Visualizing the formal model prototype
- Lessons learned

# The Project

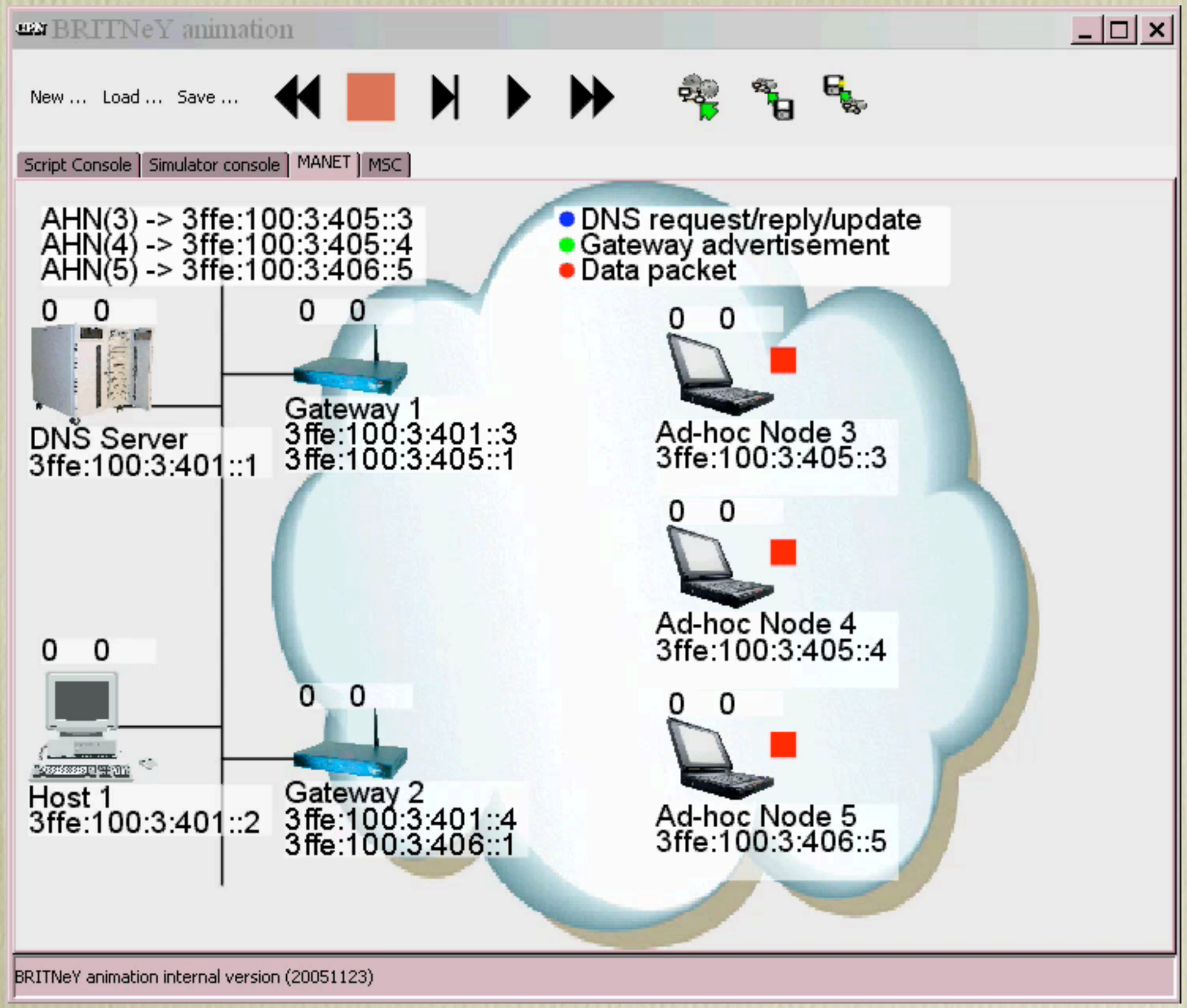
## Gateways



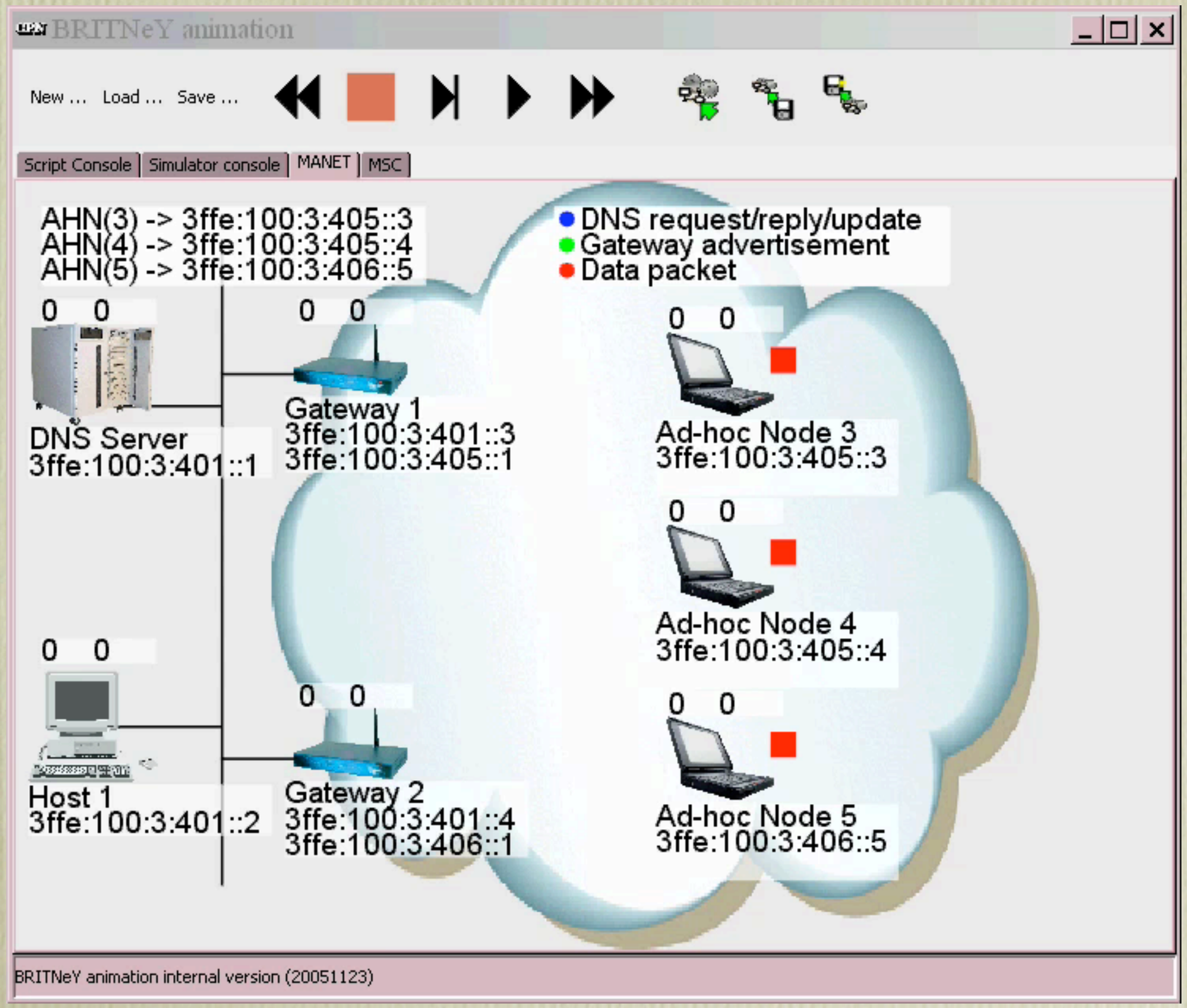

Possible solutions

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

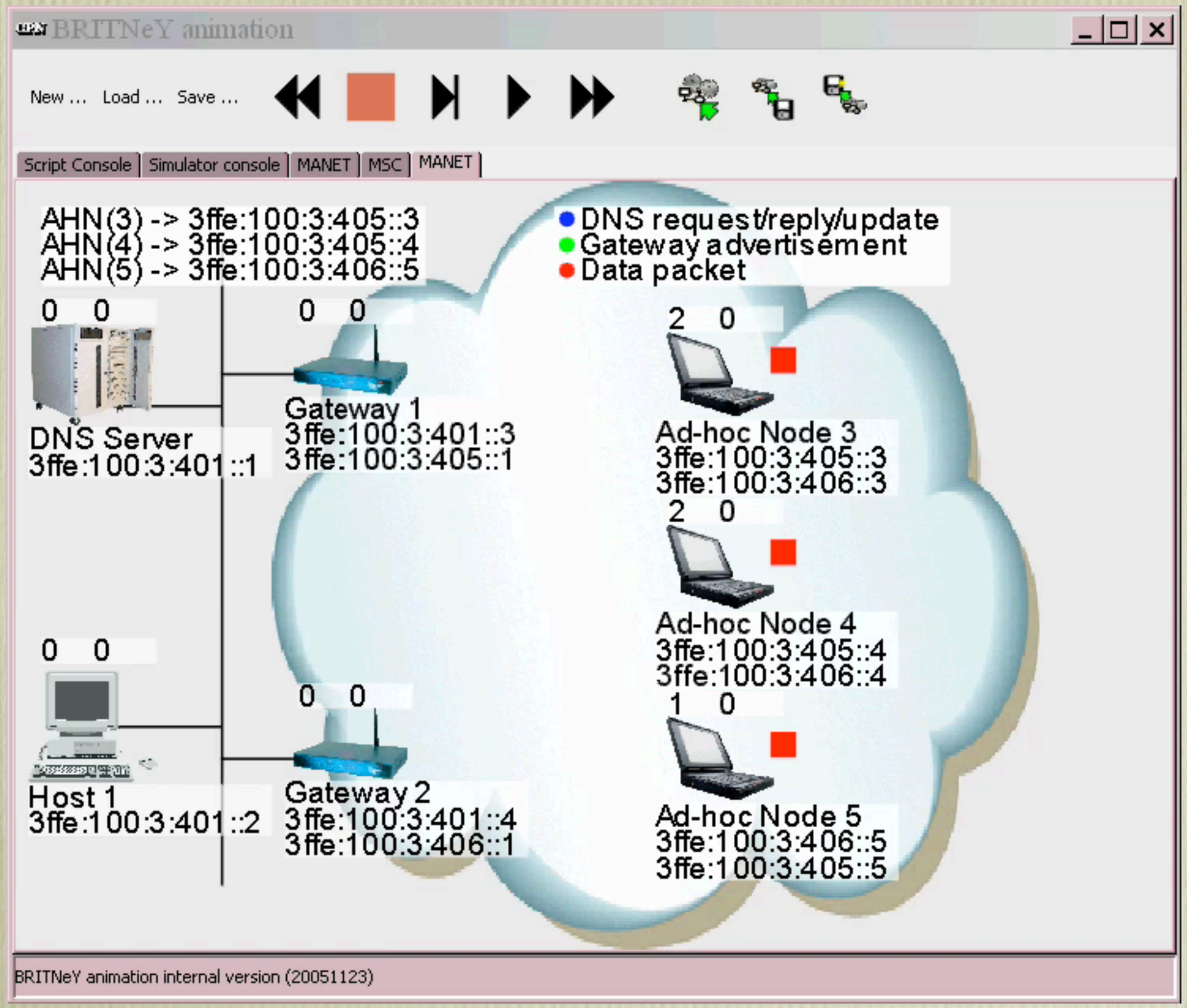
# Scenario


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
DNS Server
3ffe:100:3:401::1
Gateway 1
3ffe:100:3:401::3
3ffe:100:3:405::1
Ad-hoc Node 3
3ffe:100:3:405::3
Ad-hoc Node 4
3ffe:100:3:405::4
Host 1
3ffe:100:3:401::2
Gateway 2
3ffe:100:3:401::4
3ffe:100:3:406::1
Ad-hoc Node 5
3ffe:100:3:406::5
BRITNeY animation internal version (20051123)

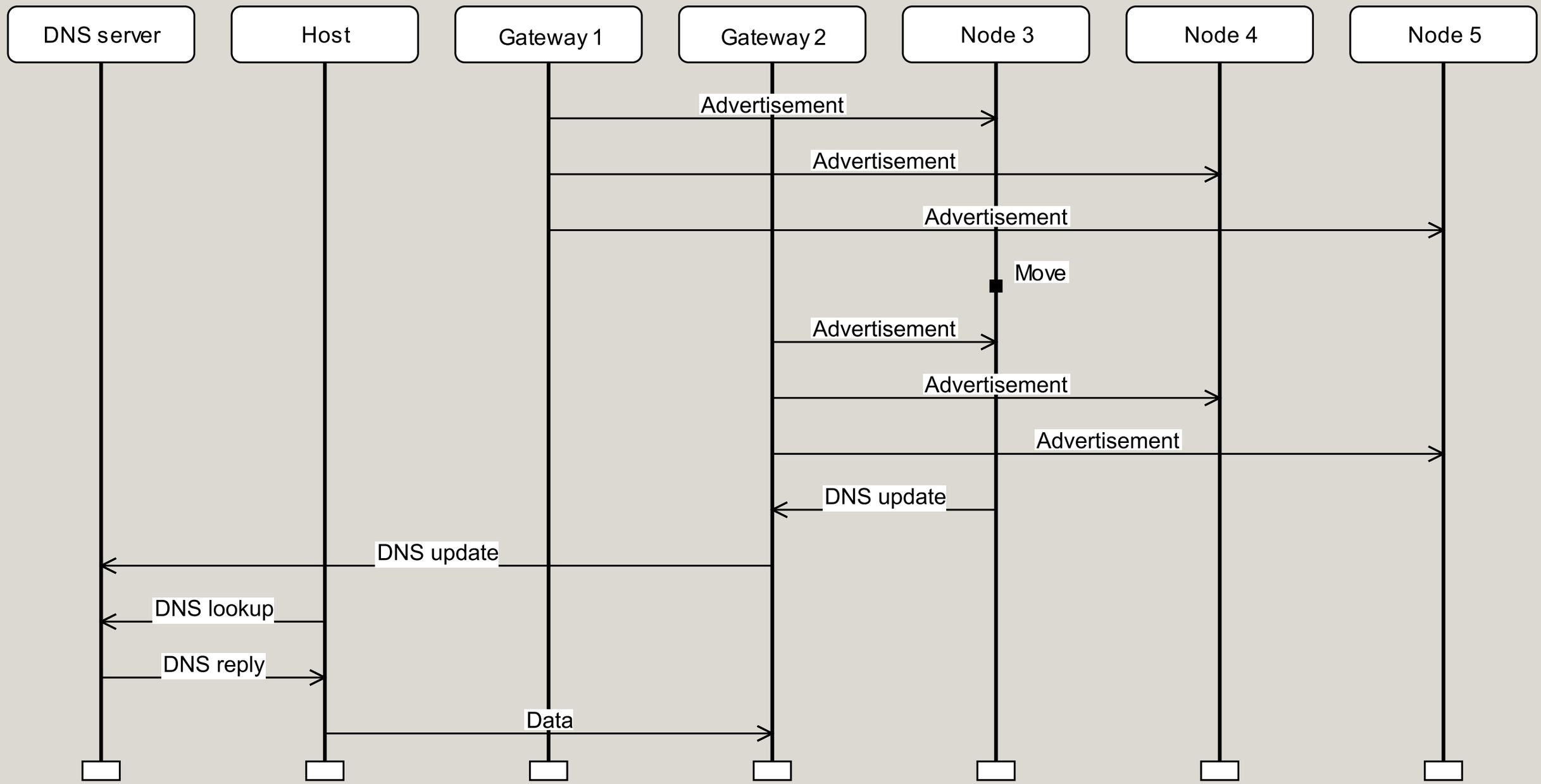
# Router Advertisements



# Mobility & DNS Update



# Mobility & DNS Update



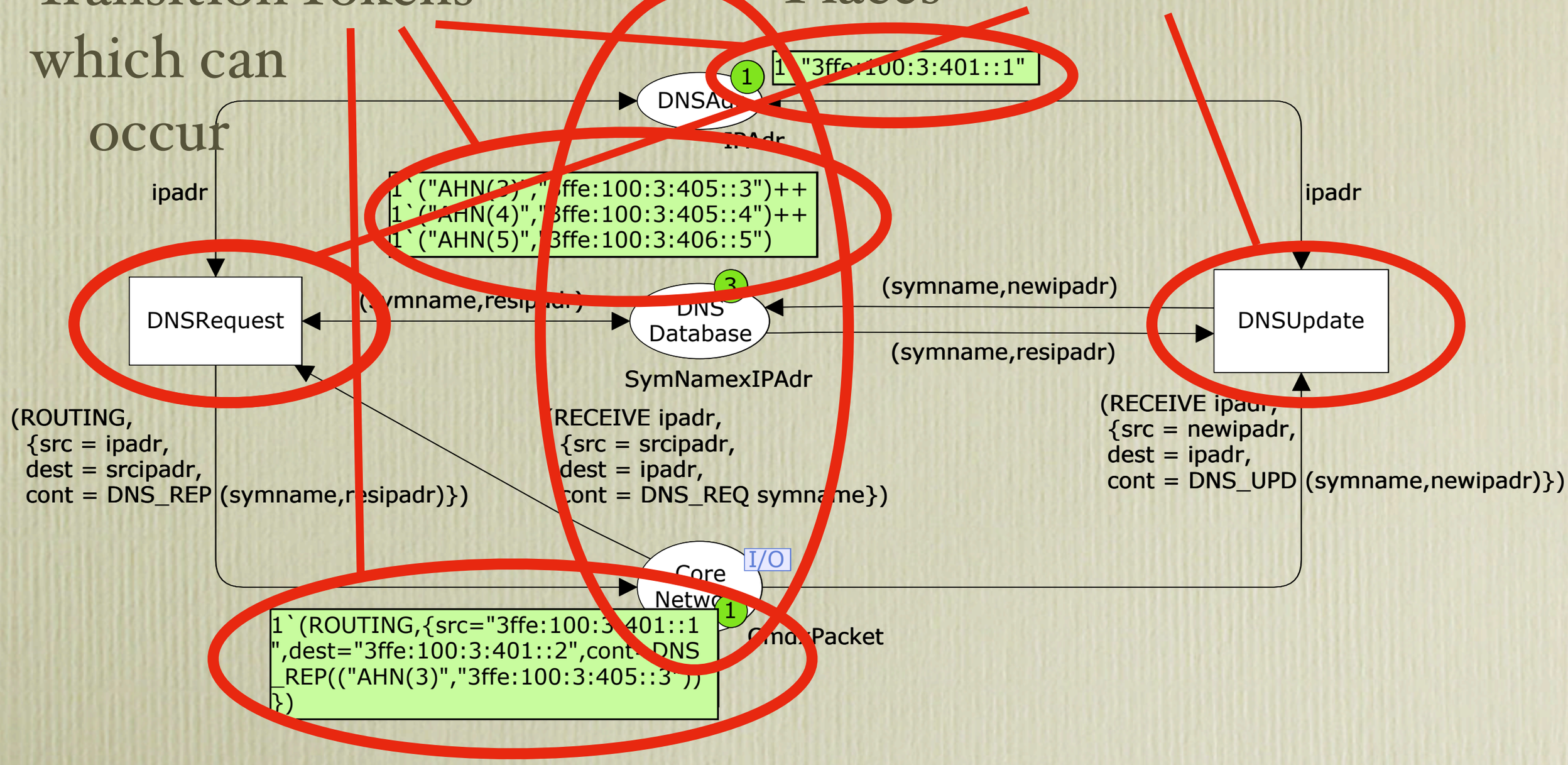
# Coloured Petri Nets

- Bi-partite directed labeled graph (places and transitions)
- Places can contain *tokens* (e.g 2, “token”, or 10.0.123.57)
- Transitions can *occur*, thereby moving tokens from *input-places* to *output-places*

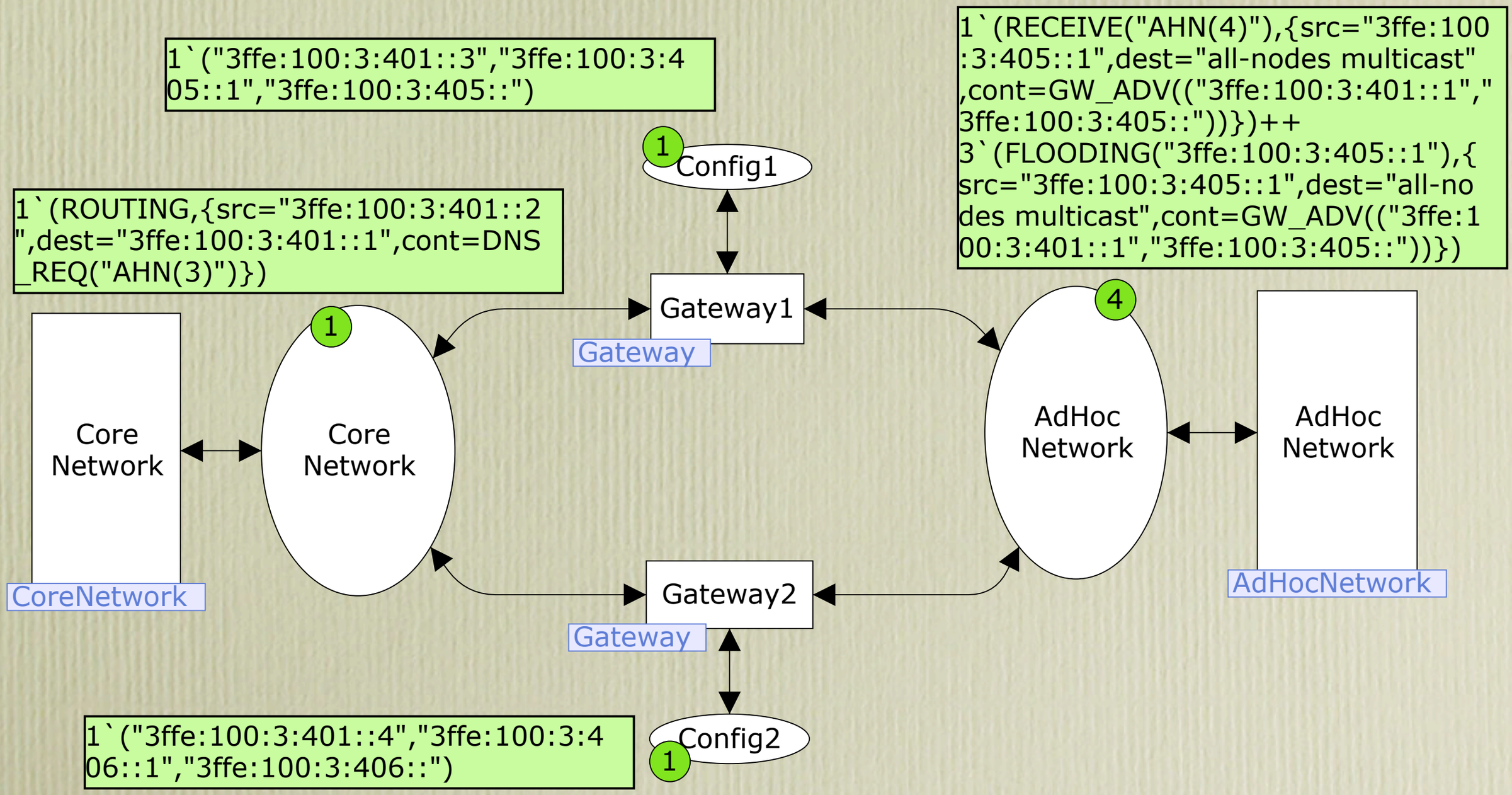
# Coloured Petri Nets Example

Transition Tokens which can occur

Places Transitions



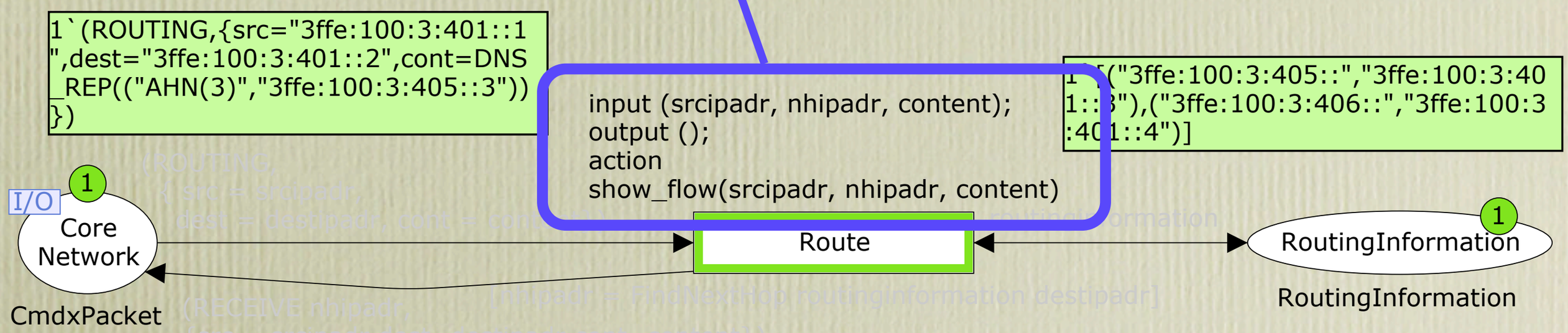
# Model


1`("3ffe:100:3:401::3","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::"))})++ 3`(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`(ROUTING,{src="3ffe:100:3:401::2",dest="3ffe:100:3:401::1",cont=DNS_REQ("AHN(3)")})
1
Config1
Gateway1
Gateway
1
Core Network
Core Network
CoreNetwork
4
AdHoc Network
AdHoc Network
AdHocNetwork
Gateway2
Gateway
Config2
1
1`("3ffe:100:3:401::4","3ffe:100:3:406::1","3ffe:100:3:406::")

# Tying the Model to the Animation

ns with code-
nenever the

```
input (srcipadr, nhipadr, content);
output ();
action
show_flow(srcipadr, nhipadr, content)
```



# 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

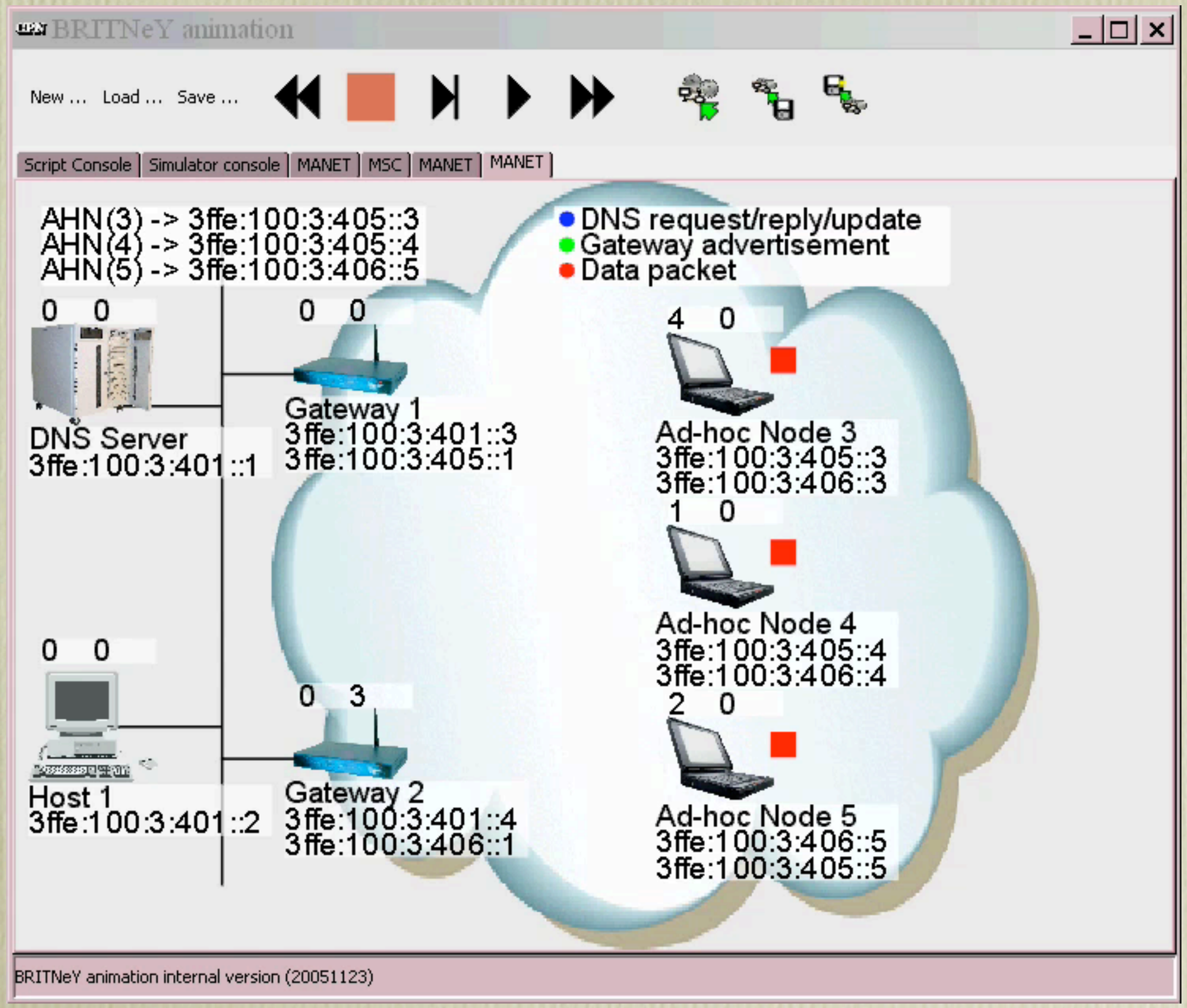
# Advantages of Integration of Formal Models with Informal Animation

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

# Future Work

- Enhancements of the modeling language to automatically trigger events in the visualization
  - Synchronous events: the current presentation is quite verbose
  - Asynchronous events: not really possible - in this project we use polling, which clutters the formal model

# Sending Data



# Sending Data and Mobility