

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

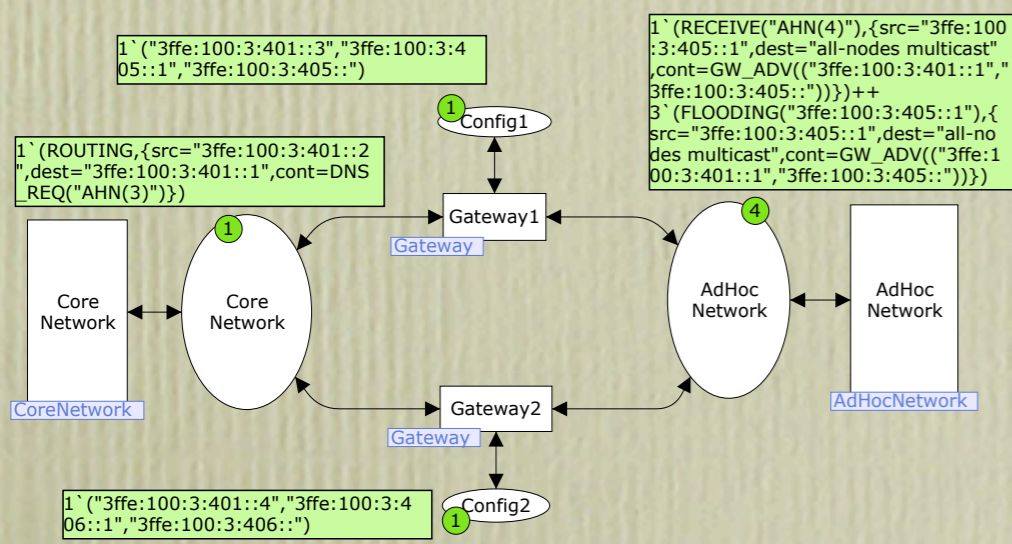
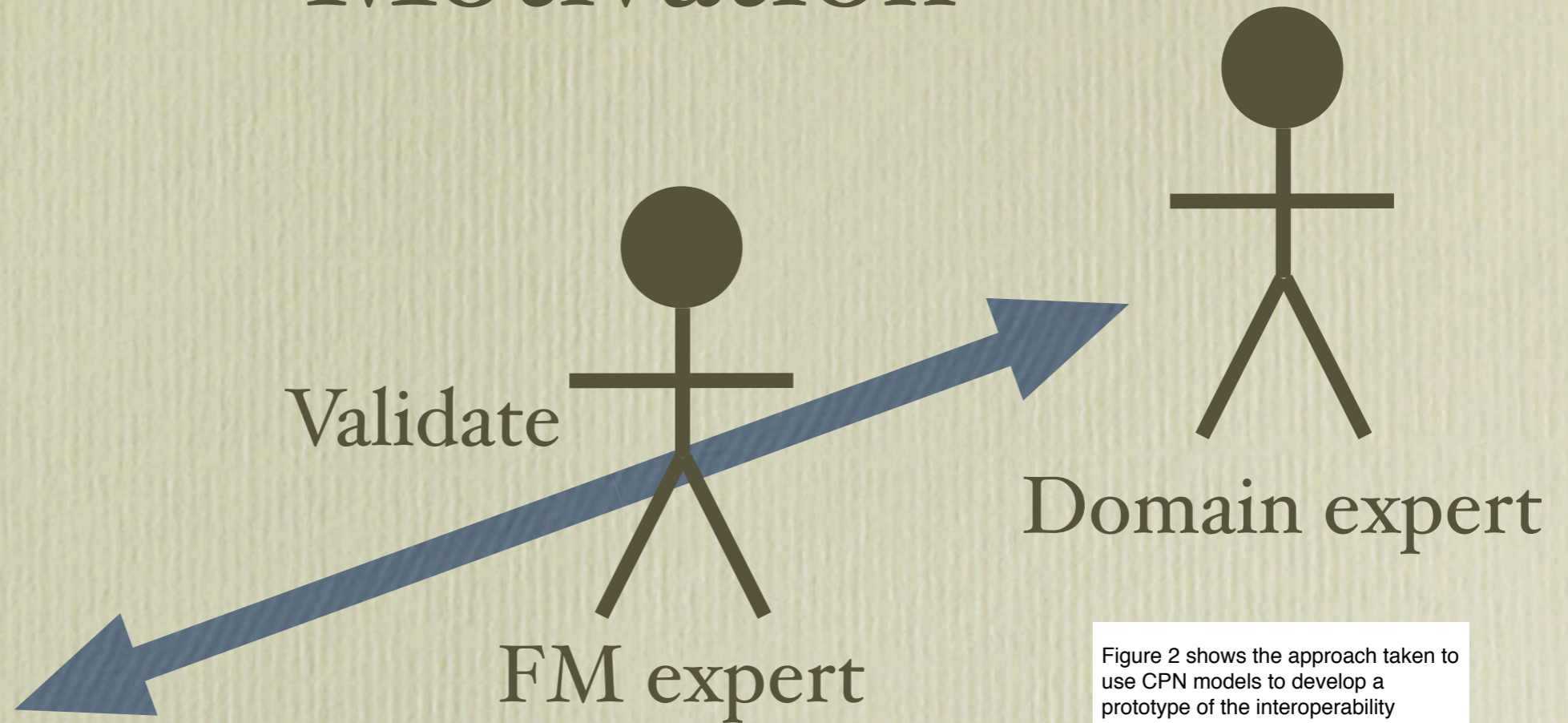
---

L.M. Kristensen<sup>1</sup>, [M. Westergaard<sup>1</sup>](#), and P.C. Nørgaard<sup>2</sup>

<sup>1</sup>Department of Computer Science,  
University of Aarhus, Denmark

<sup>2</sup>Ericsson Danmark A/S, Telebit, Denmark

# Motivation

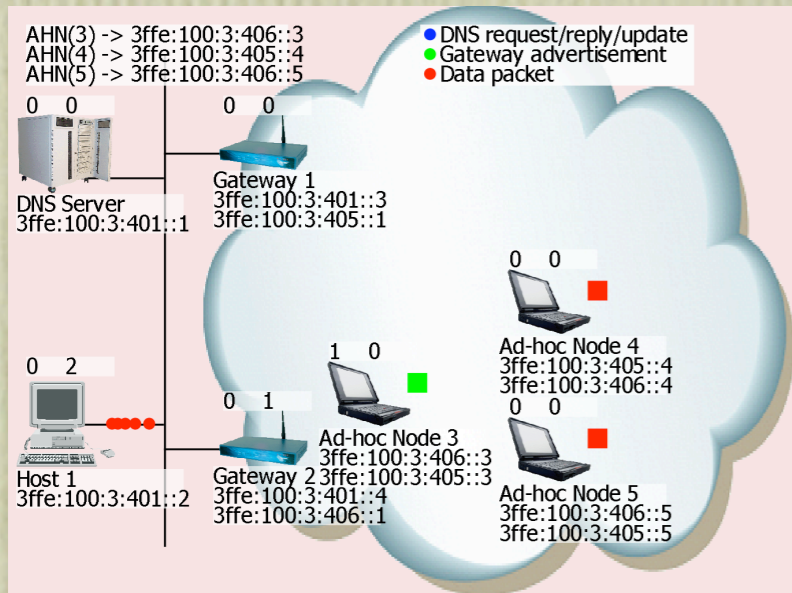


Formal model

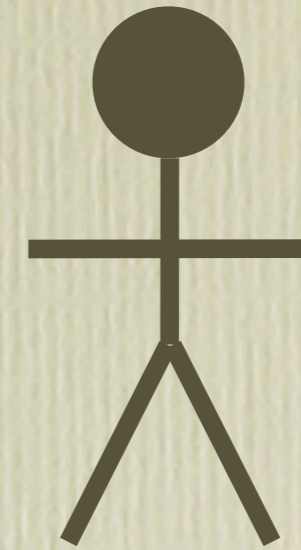
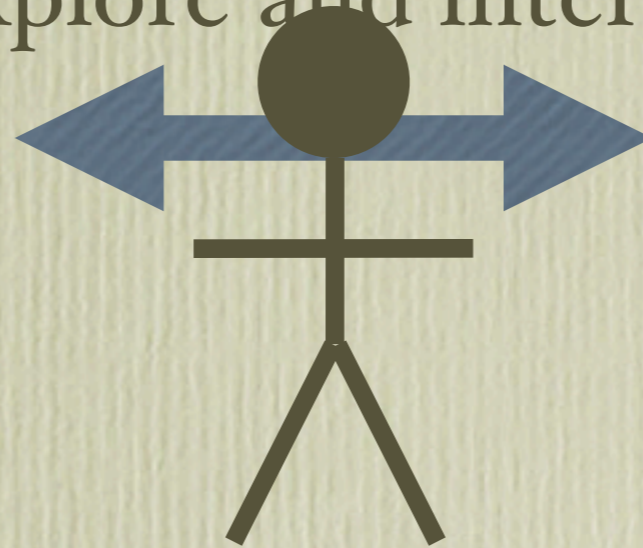
Figure 2 shows the approach taken to use CPN models to develop a prototype of the interoperability protocol. A CPN model (lower left of Fig. 2) has been developed by modelling the natural language protocol specification [22] (lower right) of the interoperability protocol. The modelling activity transforms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



Explore and interact

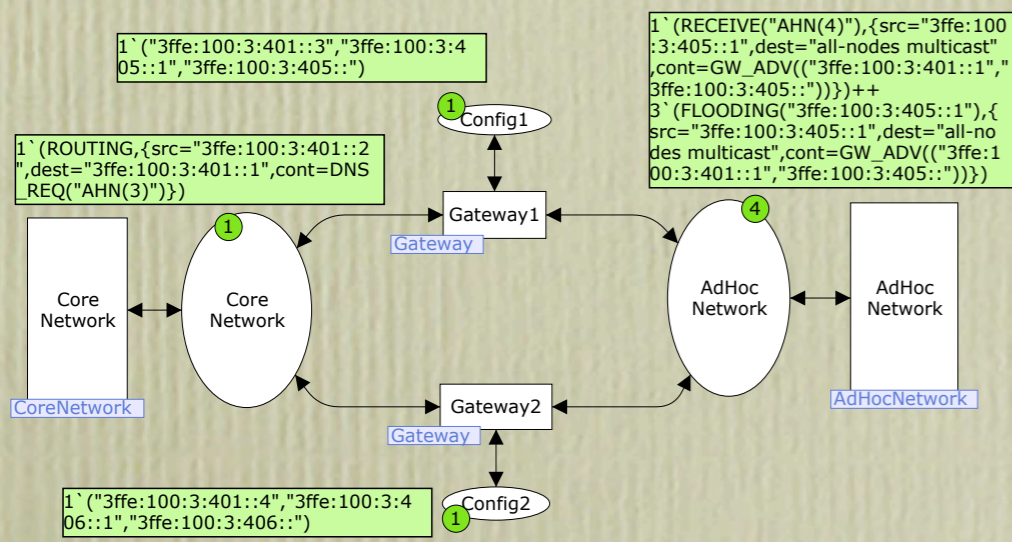


Domain expert

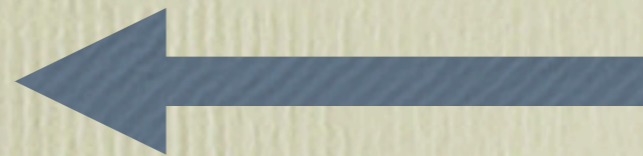
Animation

FM expert

Modeling



Formal model



Specification

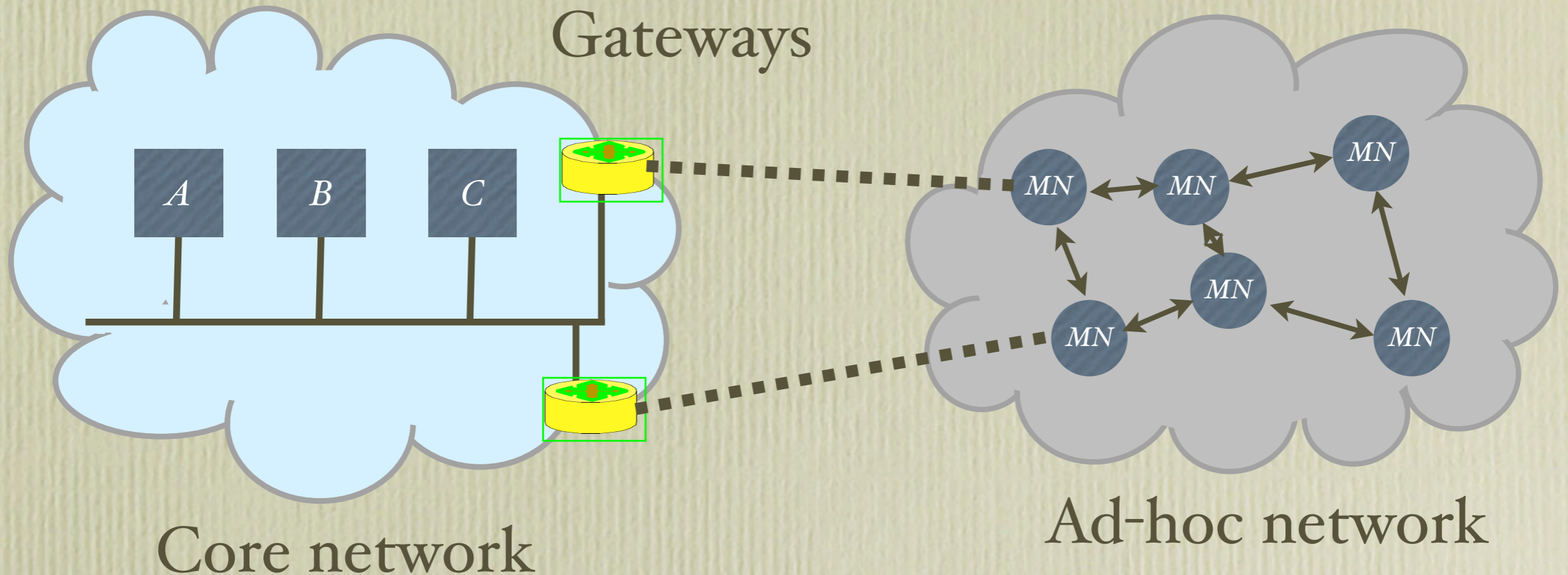
Figure 2 shows the approach taken to use CPN models to develop a prototype of the interoperability protocol. A CPN model (lower left of Fig. 2) has been developed by modelling the natural language protocol specification [22] (lower right) of the interoperability protocol. The modelling activity transforms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

# Overview

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

# The Project

Gateways

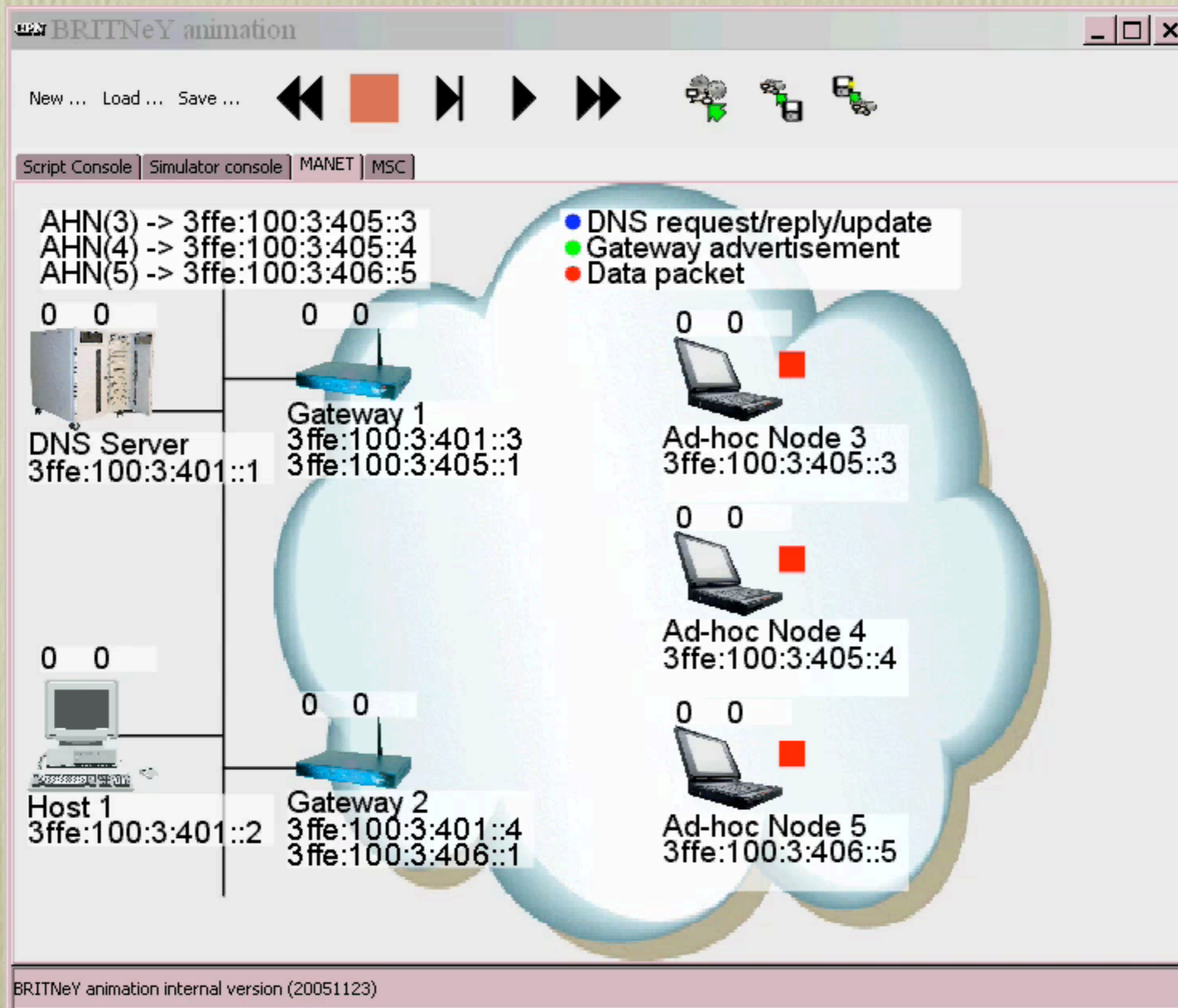


## Possible solutions

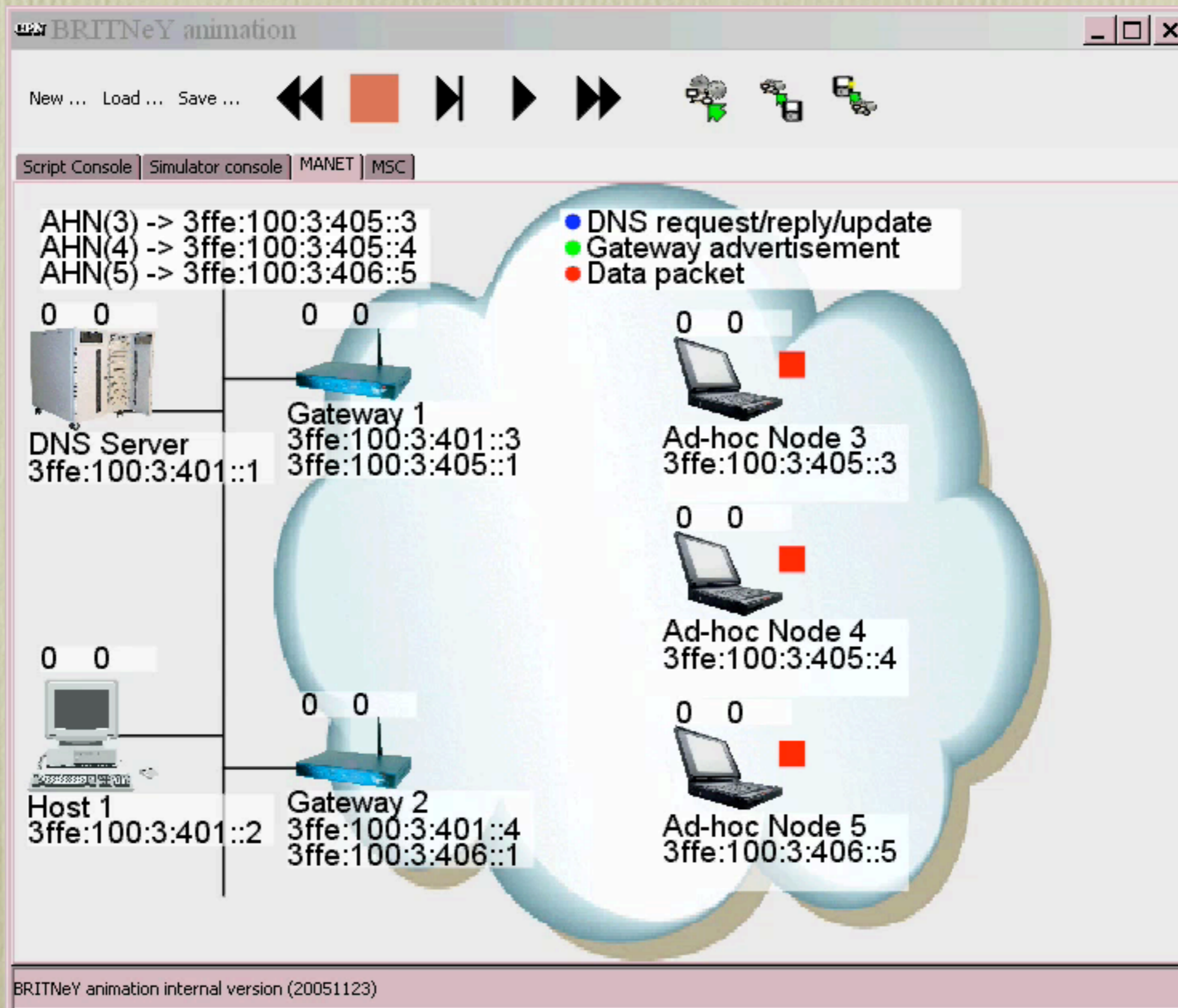
- ➔ Mobile IP
- ➔ Mobile host routes injected by gateways into the core network
- ➔ Dynamic DNS and renumbering



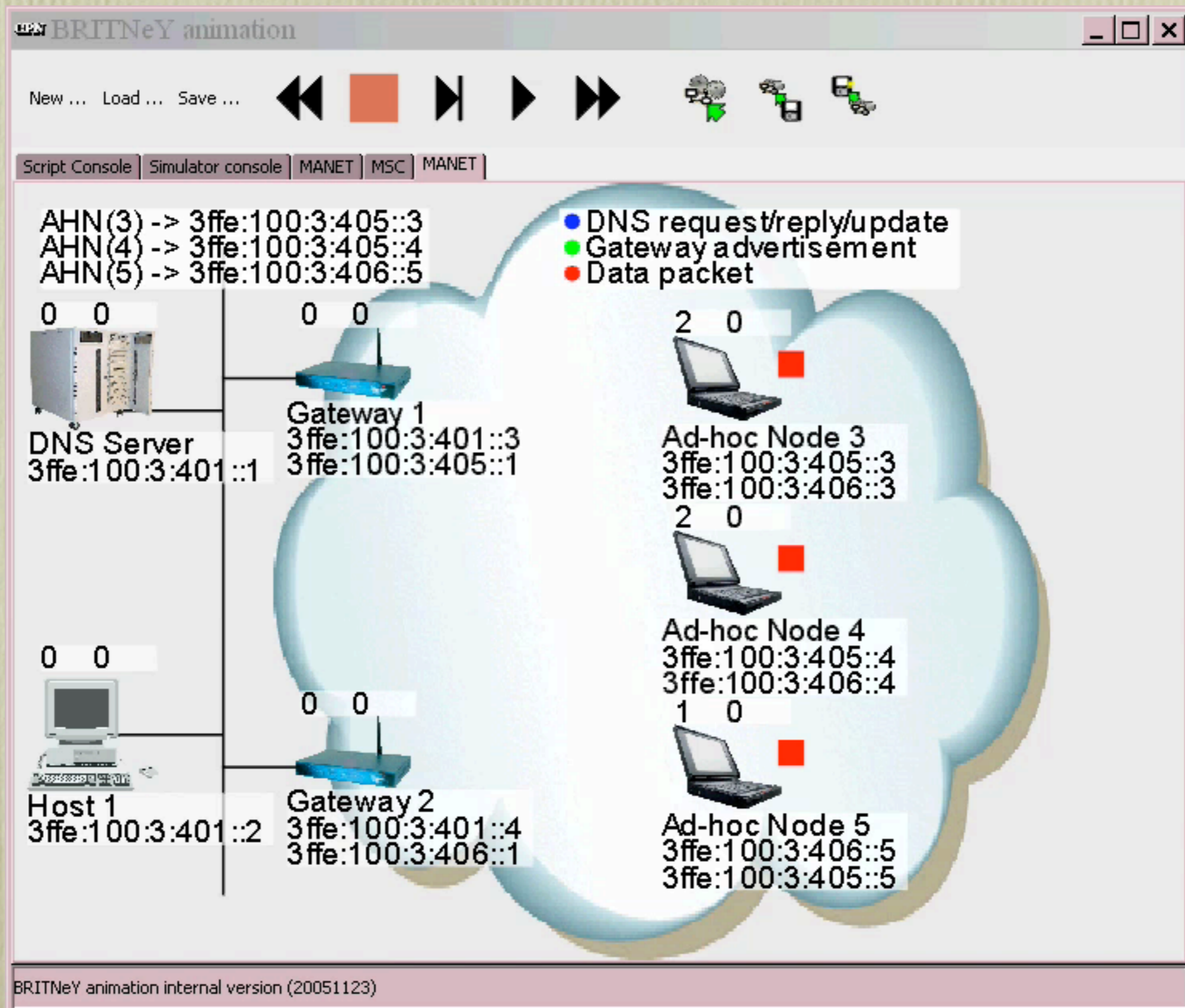
# Scenario



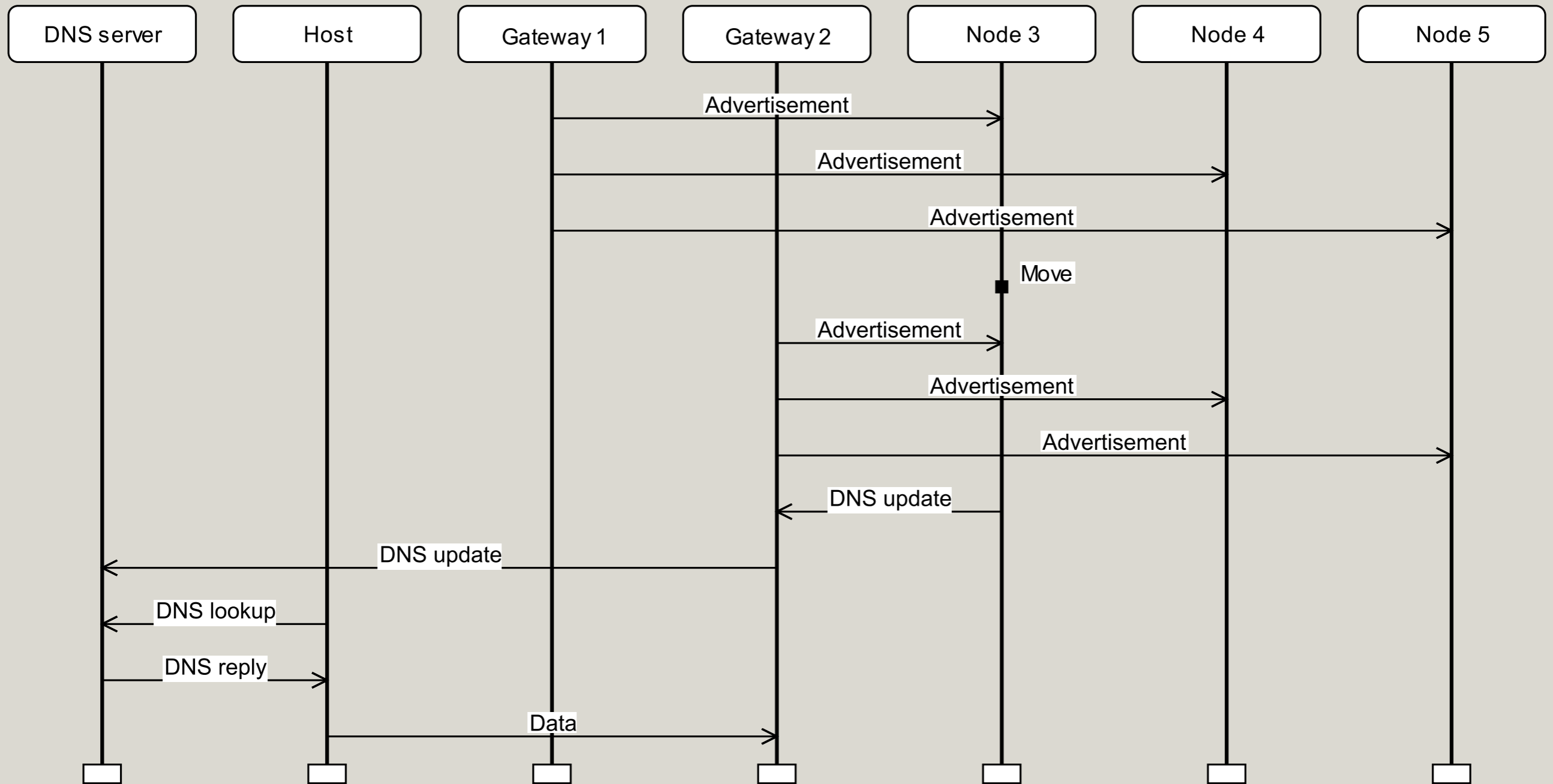
# Router Advertisements



# Mobility & DNS Update



# Mobility & DNS Update



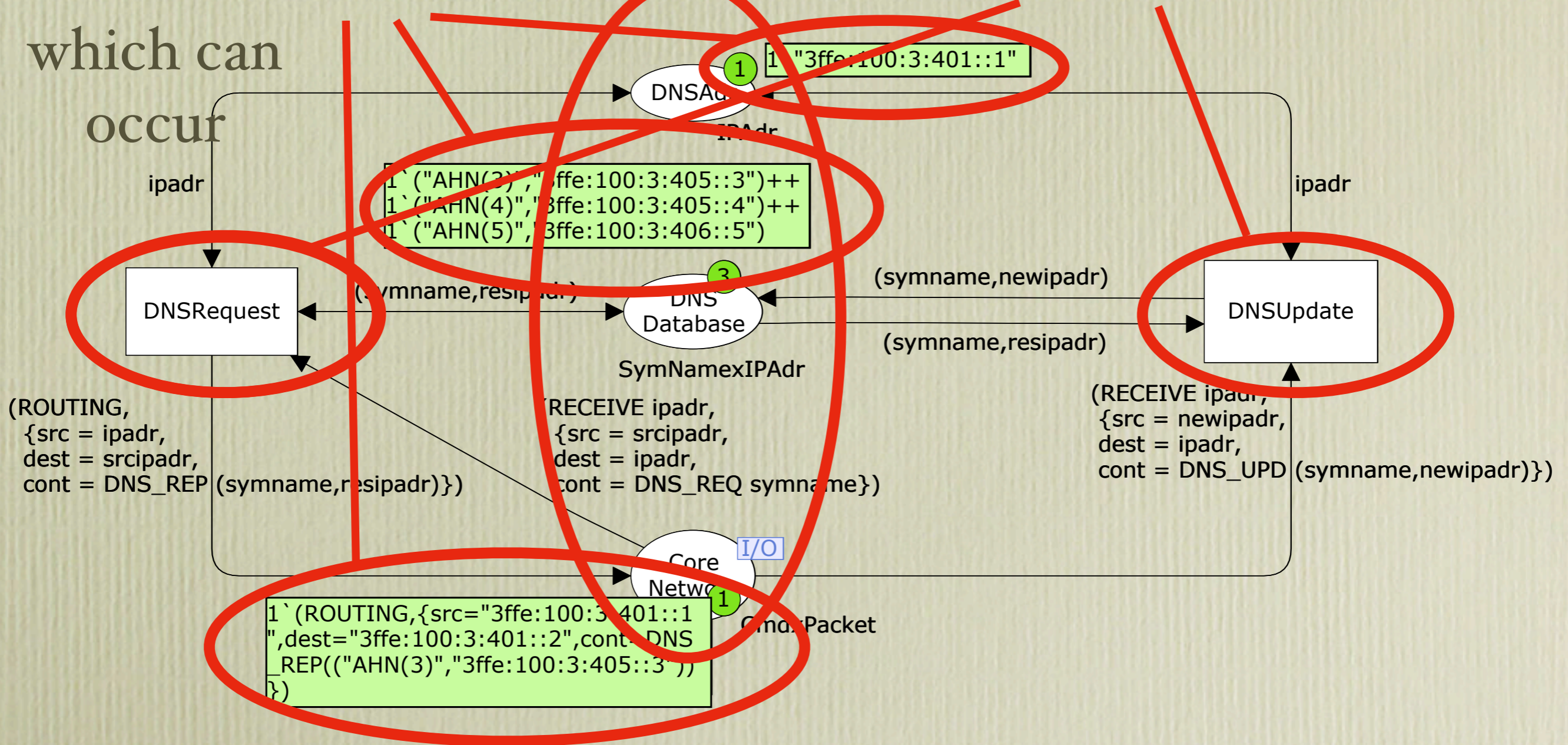
# Coloured Petri Nets

- 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*

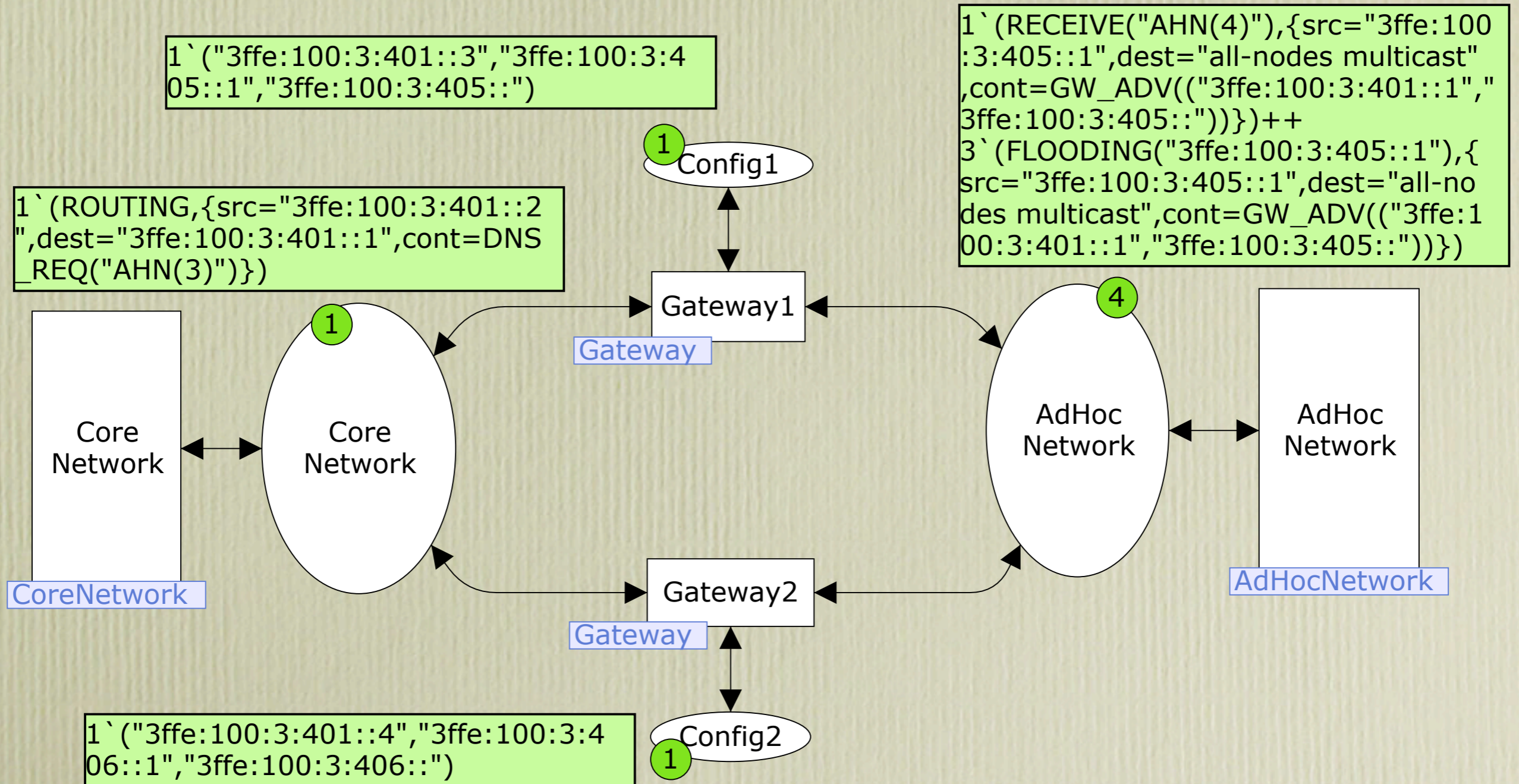
# Coloured Petri Nets Example

Transition Tokens  
which can

Places Transitions



# Model



# Tying the Model to the Animation

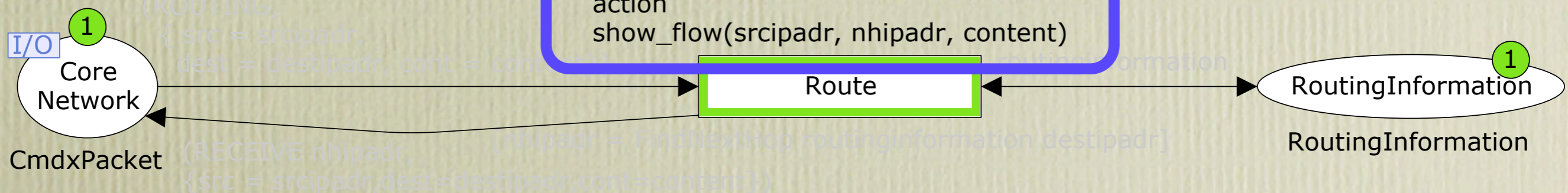
```
input (srcipadr, nhipadr, content);
output ();
action
show_flow(srcipadr, nhipadr, content)
```

ns with code-  
enever the

```
1` (ROUTING, {src="3ffe:100:3:401::1",
dest="3ffe:100:3:401::2", cont=DNS
REP(("AHN(3)", "3ffe:100:3:405::3")})
```

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

```
1` ("3ffe:100:3:405::", "3ffe:100:3:401::3"), ("3ffe:100:3:406::", "3ffe:100:3:401::4")]
```



# 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

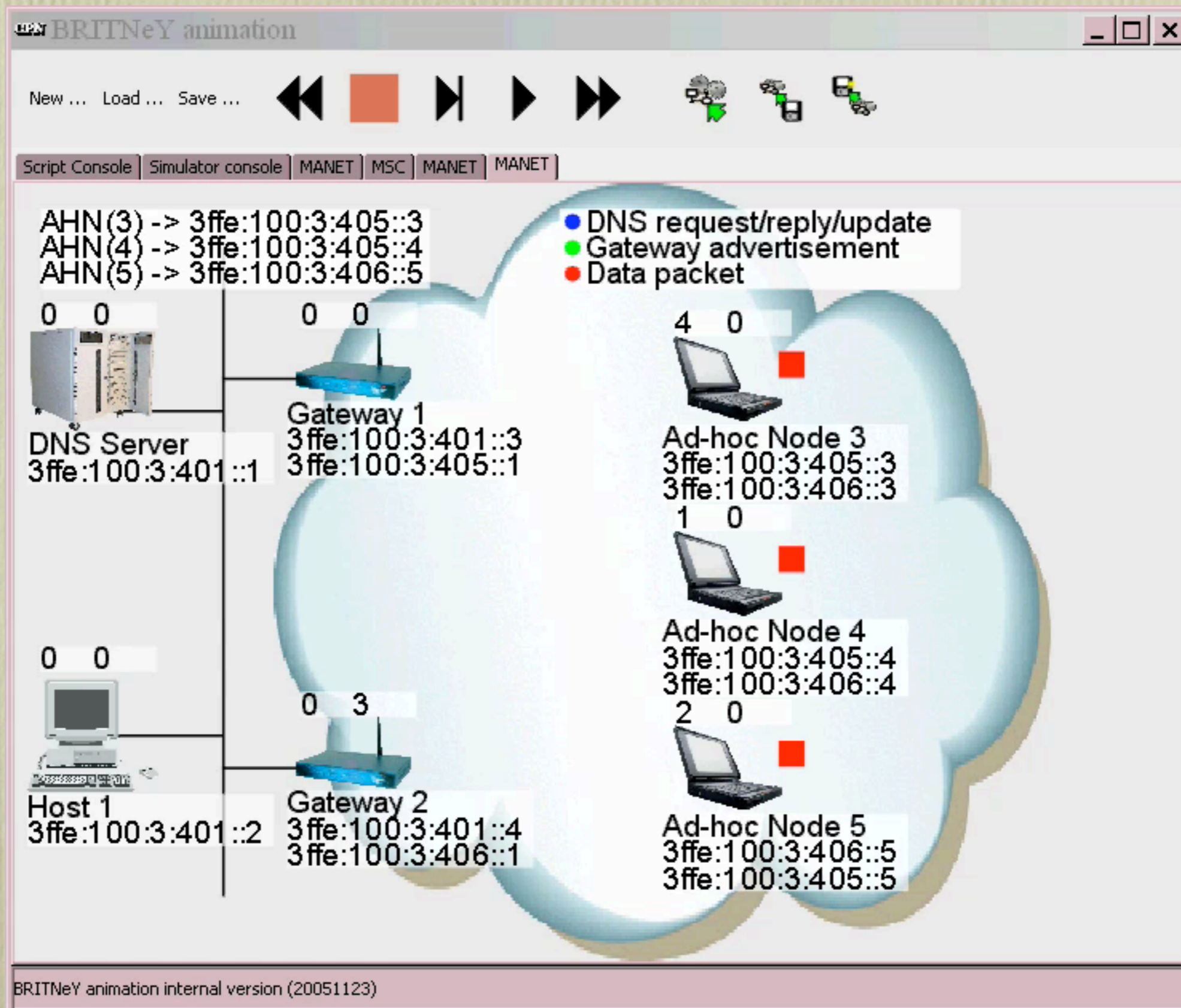
# Advantages of Integration of Formal Models with Informal Animation

- 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

