Testing and Verifying an IPv6 Based Multicast Network

Vilmos Bilicki

University of Szeged
Institute of Informatics
Department of Software Engineering

IPv6TD, 2006

Outline

Outline

Current trends

- There is an evolution of the network infrastructure and of the services provided
  - Broadband access is becoming more common { 70% of total US homes}
  - This will push the providers towards Triple Play services
    - Streaming content (TV,...)
    - On-line communication (VoIP, Video Conferencing,...)
    - Data access (Web, Email, ...)
  - IPTV is the most promising new service
    - Digital high quality interactive content (e-voting, scene selection, ...)
    - Mobile access (if there is no DVB-H,...)
    - On-line video stores

Outline

Issues with the streaming content

- Problems to be solved
  - Very high number of users
  - New user expectations / traffic types (delay, jitter, error rate)
- Current solutions:
  - Unicast communication -> inefficient bandwidth usage
  - Multicast communication -> new technology (the same age as the Internet :))
    - IPv4 small address space
    - Lack of working experience in this field
    - Lack of testing and monitoring solutions in this field
- Questions:
  - Is my network ready for IPv6 multicast?
  - What kind of QoS properties can I guarantee for my customers?
  - How can I deal with special situations?

Outline

Free tools for IPv6 multicast monitoring/testing

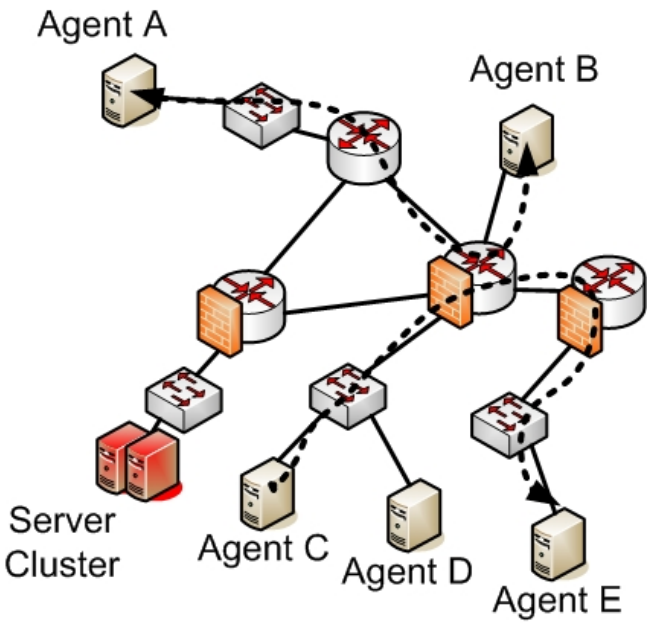
- For monitoring:
  - DBeacon
  - ?
- Testing
  - TTCN3 based tools
  - D-ITG
  - ?

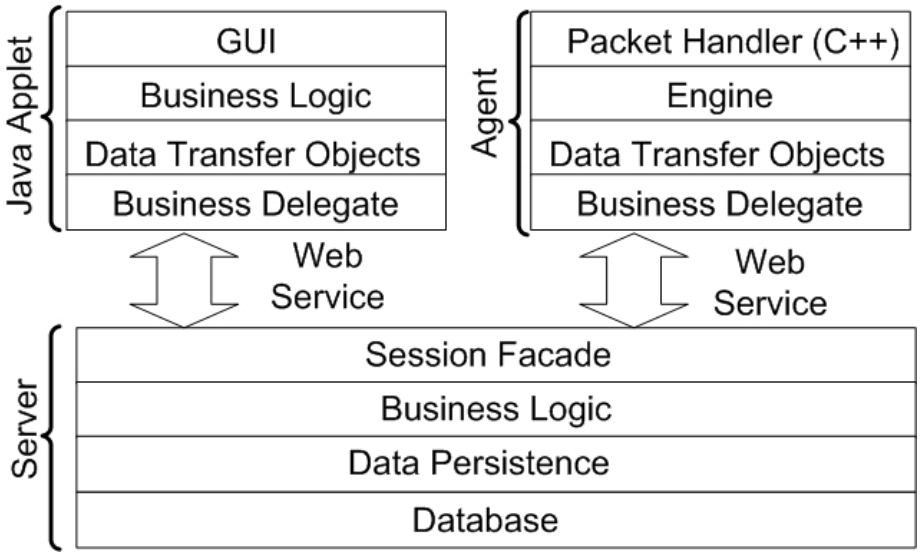
Outline

The NetSpotter framework

- The goal of this framework is to provide an easy-to-use open source tool for system administrators not just for system monitoring but for system testing too.
- The primary goal was to support the monitoring and testing of the IPv6 multicast network
- It provides the following set of services:
  - Network discovery (SNMP, L2, L3, MCast topology)
  - Collecting different performance counters
  - Arbitrary traffic definition
  - Multi-point traffic generation and measuring
- It is implemented in Java (J2EE and J2SE)

High level overview

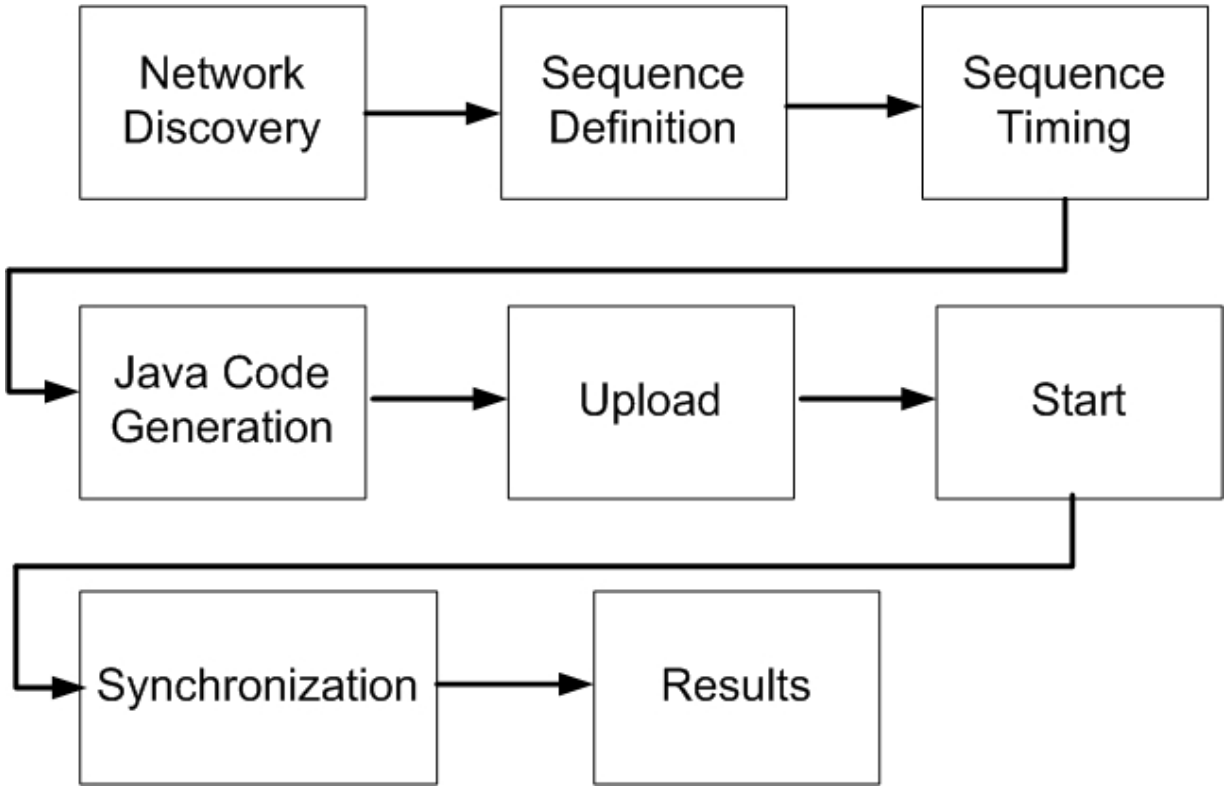




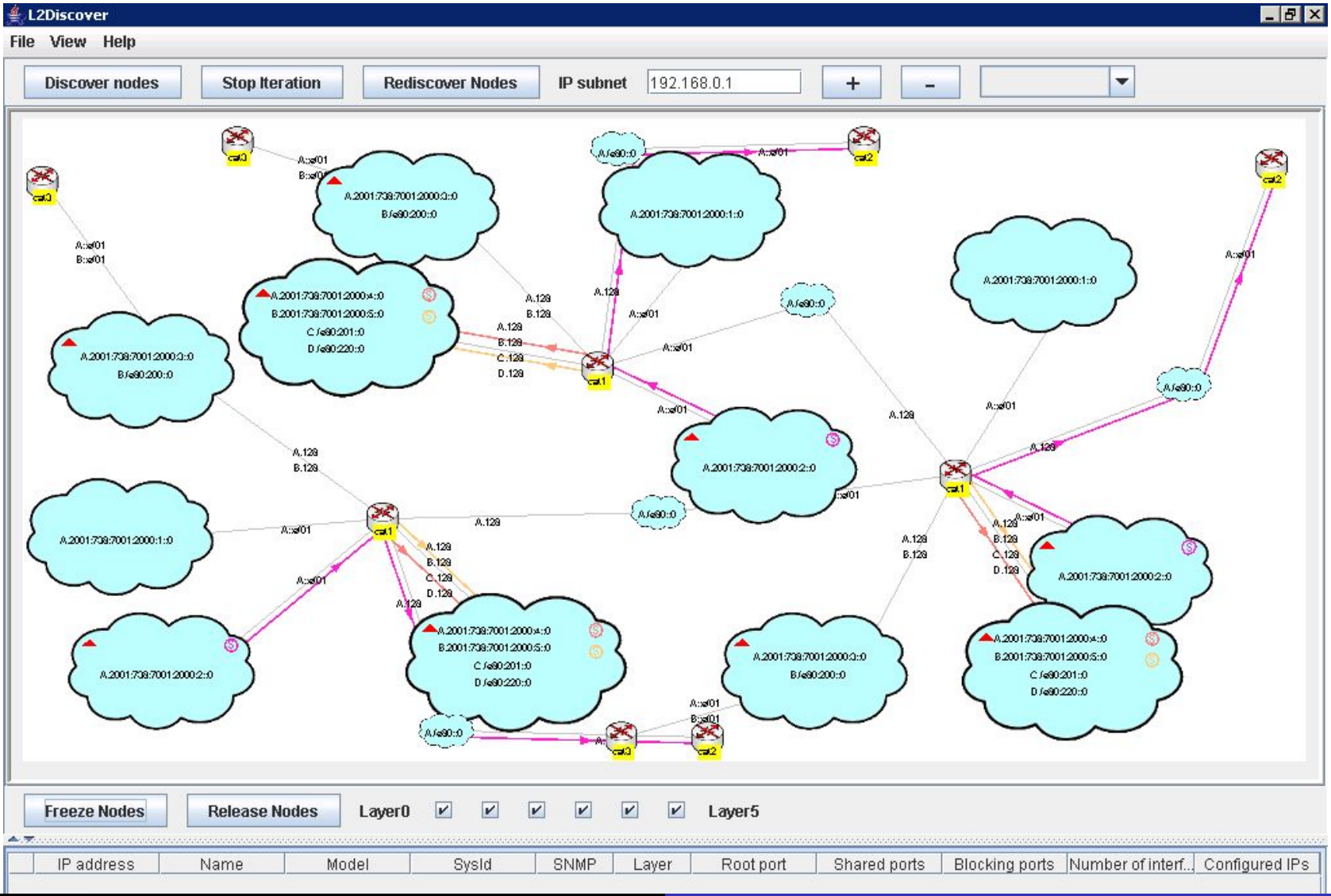
Network testing capabilities

- Traffic classes
  - Frame level
  - Packet level
  - Socket level
- Message construction
  - XML-based solution
  - Inheritance is supported
  - Dynamic values
- Sequence construction
  - MSC-based solution
  - The most important MSC features can be used (loop, reference,...)
  - Synchronization can be achieved using:
    - Delay elements
    - Synchronizing messages from the central cluster

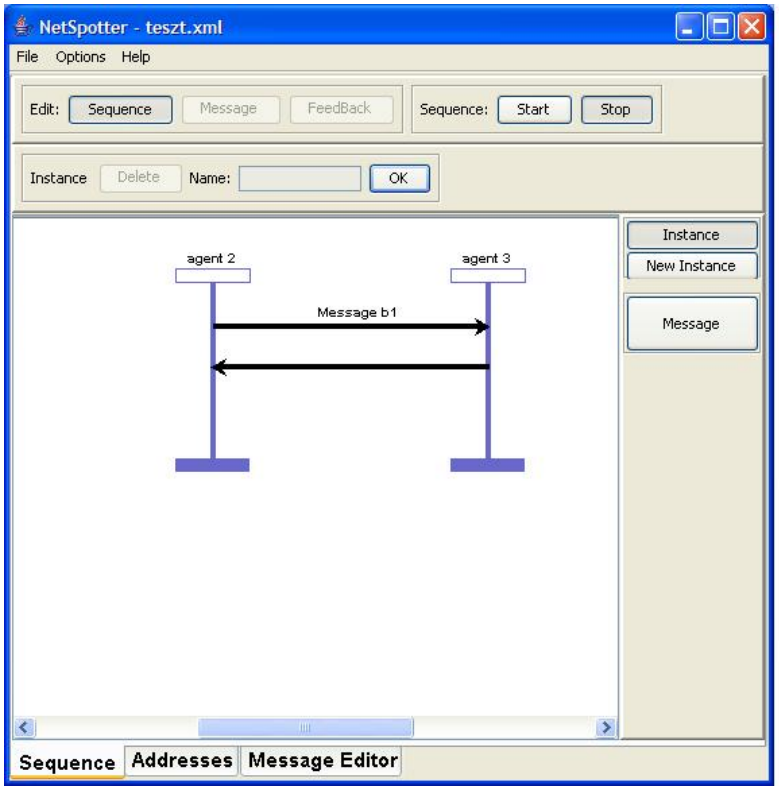
Work flow


Network Discovery
Sequence Definition
Sequence Timing
Java Code Generation
Upload
Start
Synchronization
Results

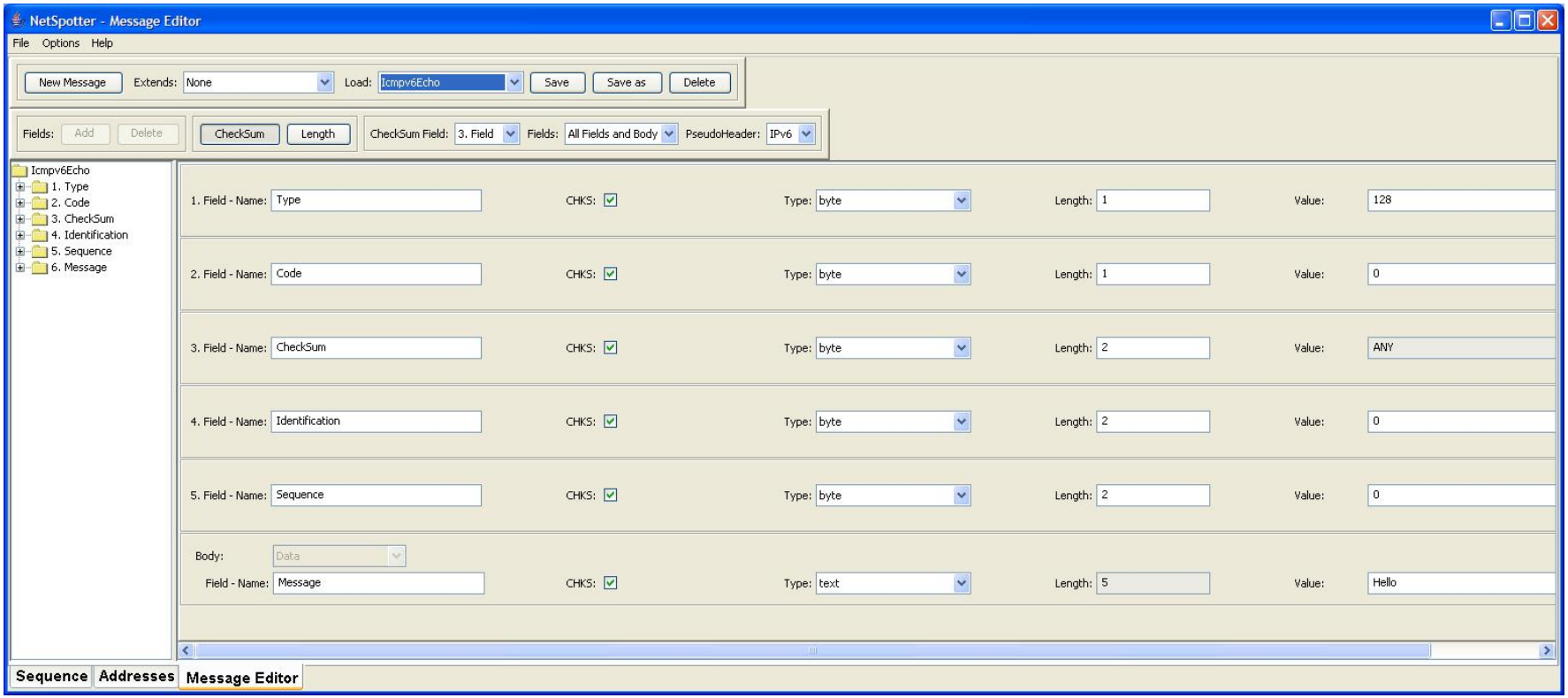
Main screen


L2Discover
File View Help
Discover nodes
Stop Iteration
Rediscover Nodes
IP subnet
192.168.0.1
+
-
Freeze Nodes
Release Nodes
Layer0
Layer5
IP address
Name
Model
SysId
SNMP
Layer
Root port
Shared ports
Blocking ports
Number of interf...
Configured IPs

Sequence construction



Message construction



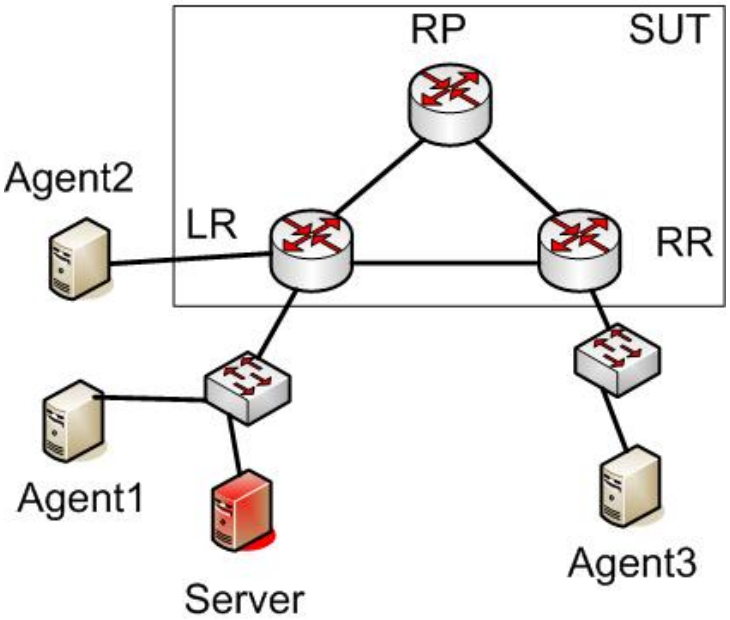
Outline

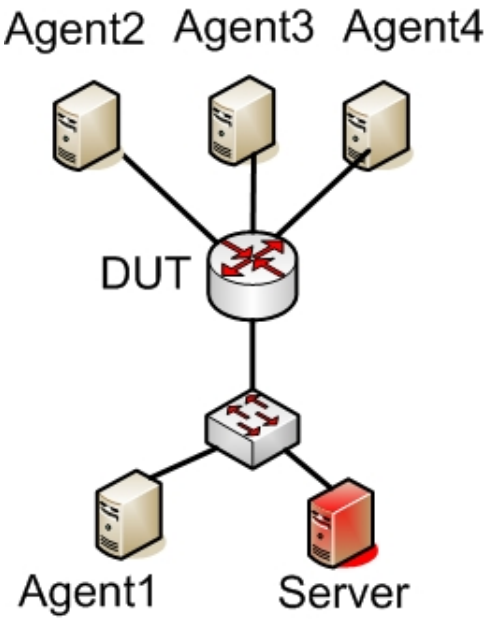
IPv6 multicast measurmenets

- RFC 2544, IPv6 unicast Benchmarking Methodology draft
  - Single-port / Multi-port
  - Throughput
  - Latency
  - Frame loss
  - ...
- RFC 3918
  - The most interesting measurements:
    - Scaled Group Matrix
    - Group Join/Leave Delay
  - Main deficiencies of this RFC:
    - It is only concerned with multicast forwarding not multicast routing
    - Only ASM features are tested; no FSM or SSM testing is defined

Our measurement

- We measured the effects of a large number of channels and sources on the forwarding capabilities of a Linux IPv6 multicast router





Details

- Testing platform:
  - PC P4, 1300 MHz, 512 MByte
  - Debian Sarge, MRD6 0.9.5 as a PIM-SM router
  - Zebra Ripng as unicast routing algorithm
- The measurement
  - We injected 50000 IPv6 multicast UDP packets from the source agent
  - Packet loss: the destination agent measured the number of received packets
  - Delay: The destination joined a defined number of channels and measured the time needed for receiving the subscribed channels

Results

- Without some kind of ACL list the multicast network is vulnerable to DOS attacks
- The number of channels influences the delay and the packet loss significantly

N.Ch	64	512	1500
10	50000	50000	49200
100	49514	49664	43311
1000	46813	43808	41642
10000	n.a	n.a	n.a
60000	n.a	n.a	n.a

Table: Packet loss

N.Ch	64	512	1500
10	17	23	14
100	227	254	319
1000	3800	3700	4200
10000	72777	>70000	>70000
60000	>70000	>70000	>70000

Table: Delay (ms)

Conclusions

- NetSpotter framework
  - We presented the NetSpotter framework which can be used not just for protocol validation but network testing too
- Measurements
  - We measured the MRD6 PIM-SM implementation
  - It turned out that one can easily initiate DOS attacks against the multicast network with several special packages

Future plans

- The first version of the framework will be finished by autumn of this year
- We would like to create protocol templates for the most important protocols (e.g. SIP, http, ...)
- In the second version we plan to do the following:
  - We would like to improve the defining and handling of TTCN3-like logic formulas
  - We would like to extend the network discovering capabilities with L2 information (CAM,...) and with mulitcast information from the agents

Thank you for listening

- The homepage of the NetSpotter project: http://sourceforge.net/netspotter

Q and A