Realistic, Extensible DNS and mDNS Models for INET/OMNeT++

2nd OMNeT++ Community Summit, 2015

Andreas Rain, Daniel Kaiser, Marcel Waldvogel

University of Konstanz, Konstanz, Germany

September 4th, 2015
What is this work about?

DNS

• Design networks using DNS
• Design new extensions to DNS
• Evaluate performance and validate behavior

mDNS/DNS-SD

• Use mDNS for discovery
• Evaluate mDNS in combination with a new multicast transport protocol as a use case

Privacy Extension

• Find new ways to enhance the privacy of users
• Validate your design

Stateless DNS

• Discovery without infrastructure (more or less)
• Test Stateless DNS and check whether it fits your needs
Figure: Overview of the simple modules belonging to the DNS model.
Figure: Overview of the simple modules belonging to the DNS model.
Design DNS zones using the BIND syntax

Example Configuration

$TTL 86400 ; 24 hours, $TTL used for all RRs
ORIGIN uni-konstanz.de.
@ IN SOA pan.rz.uni-konstanz.de. hostmaster.uni-konstanz.de. ( 2003080800 ; sn = serial number
172800 ; ref = refresh = 2d
900 ; ret = update retry = 15m
1209600 ; ex = expiry = 2w
3600 ; nx = nxdomain ttl = 1h )
IN NS pan.rz.uni-konstanz.de. ; in the domain
IN NS uranos.rz.uni-konstanz.de. ; slave
IN MX imap.uni-konstanz.de. ; external mail
IN A 134.34.240.80 ; ip of origin

; server host definitions
pan.rz IN A 134.34.3.3 ; this server
uranos.rz IN A 134.34.3.2 ; the slave server
imap IN A 134.34.240.42 ; mail server imap
www IN CNAME proxy-neu.rz ; test on
proxy-neu.rz IN A 134.34.240.80 ;

Figure: Example zone configuration based on BIND syntax.
## Capabilities, Limitations, and Challenges

### Capabilities
- Model DNS networks
- Hierarchical structures
- Recursive and iterative resolving
- A, AAAA, NS, PTR, SRV, CNAME, TXT
- Name compression

### Limitations
- Manual modeling
- Bailiwick rules
- Not all record types
- Dynamic zone updates
- DNSSEC

### Challenges
- Dynamic generation
- Extensible design
- Mapping of rules
- RFC ↔ Implementation-specific
- Integration
mDNS Simulation Model

Figure: Structure of the mDNS simulation model and various components.
mDNS Simulation Model

Figure: Structure of the mDNS simulation model and various components.
Dynamic mDNS resolver networks

Parameters:
- Number of Resolvers
- Number of Private Resolvers
- Maximum amount of friends
- Minimum amount of friends
- Maximum amount of services
- Minimum amount of services
- Ratio of public to private services

Figure: Dynamic mDNS network in its basic form.
Capabilities, Limitations, and Challenges

Capabilities
- mDNS and DNS-SD
- Dynamic mDNS network generation
- Our privacy extension for mDNS
- Name compression

Limitations
- Shared resource records not handled differently
- Dynamic services
- Internal messages are not used to query or announce
- Not all resource record types are supported

Challenges
- Scheduling
- Reference implementations
- Dynamic generation
- Extensibility
- Integration
Extensions

Privacy

Stateless DNS

- Combine with other protocols
- Validate behavior
- Add new functionality

Implement your own extension!

Figure: Evaluation of traffic reduction by the privacy extension.
Usage

Example DNSCache

1. Extend the DNSCache interface.
2. Implement the methods and thus your caching strategy.
3. Simply change the DNSCache implementation used in the server.

Example DNSServer

1. Extend the DNSServerBase class (if needed).
2. Implement handleQuery
3. Return DNSPacket to send it or nothing when recursion is initiated
Conclusions & Future Work

Possible future work:

- Dynamic generation of DNS networks
- Implementation of DNSSec
- DNS caching analysis
- Evaluation of other extensions
- Better integration with INET

What we are working on:

- Evaluation of the impact of mDNS on WLANs.
- Simulations performed on the \textbf{bwUniCluster} . . .
- . . . with up to 800 Simulations in parallel.


——, “Multicast DNS,” February 2013, RFC 6762.


