VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODELLING NEW NETWORK ARCHITECTURES IN OMNET++

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE TOMÁŠ HYKEL
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODELOVÁNÍ NOVÝCH SÍŤOVÝCH ARCHITEKTUR V OMNET++

MODELLING NEW NETWORK ARCHITECTURES IN OMNET++

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — TOMÁŠ HYKEL
AUTHOR

VEDOUCÍ PRÁCE — Ing. MARCEL MAREK
SUPERVISOR

BRNO 2015

Abstrakt

Abstract

Klíčová slova

Keywords

Citace

Tomáš Hykel: Modelling New Network Architectures in OMNeT++, bakalářská práce, Brno, FIT VUT v Brně, 2015

Modelling New Network Architectures in OMNeT++

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval sám pod vedením Ing. Marcela Marka

....
Tomáš Hykel
February 27, 2015

Poděkování

Contents

Chapter 1

Introduction

(˜2 ns)

1.1 Goals

The theoretical part of this thesis aims to describe alternatives to the currently prevalent network architectures. Since the Internet is by far the largest and most important example of an internetwork, its underlying architecture shall be used as a base for comparison. This is only fitting since nearly all of the network architecture research is directed towards improving the building blocks of the Internet.

The technical report describes design and implementation of a significant part of an OMNeT++ simulation model of one of the presented architectures, RINA.

1.2 Thesis Structure

Part Two describes the historical events that led to the current state of the Internet and hints on the shortcomings and weak parts of current technology which create the need for an alternative architecture research.

Part Three presents an outline of current research directions in field of network architecture and a brief overview of related research projects.

Part Four takes a closer look at Named Data Networking, an information-centric network architecture. It contains description of the architecture and its advantages.

Part Five takes a closer look at Recursive InterNetwork Architecture, an IPC-based network architecture. It contains description of the architecture and its advantages.

Part Six describes implementation of RINA's Relaying and Multiplexing Task in OMNeT++.

Part Seven presents evaluation of the implementation in form of test outputs.

Part Eight wraps it all up. ;=)

Chapter 2

Problems of the current Internet

(˜5 ns)

The Internet could be, with no doubt, considered one of the most important technological achievements of the 20th century. It has brought a previously unimaginable degree of interconnection and information access to the whole world and its importance keeps growing even decades after its inception.

Nevertheless, the very basic core of its technology was constructed over three decades ago, back when the the demands on internetworking capabilities were nowhere compared to the present situation. During the Internet's growth, whenever there has been a problem that required a solution (e.g. bla bla), it's been usually dealt with in a non-intrusive evolutionary fashion by applying a new principle on top of the underlying technology. In another words, problems have been mostly solved by adding a new protocol to the TCP/IP protocol stack (e.g. RSVP for QoS guarantee).

This way of improving the Internet's base technology is only logical since each paradigm shift in foundations of the Internet can require a long and expensive transfer of existing network configrations to the new technology. The most notable example of this is the internet layer protocol IPv6 requiring hardware support from active network components. The problem of IPv4 space exhaustion has been known of since 1992, IPv6 specification arose in 1996 and the first IPv6 routers emerged in 2001 – and yet, as of 2015, two years after the IPv4 exhaustion, IPv6 still represents only 10 % of global Internet communication and the transition isn't getting faster.

As such, some of the Internet's problems are inherent because of the base design and it's usually difficult, if not impossible, to solve them incrementally. The following sections illustrate several of those problems.

2.1 Node naming

TODO: explain Saltzer's proposal and why it isn't used

Thus, both MAC addresses and IP addresses serve as point-of-attachment addresses and effectively locate the same thing, which is a host interface. This implies that the Internet effectively forwards PDUs on interface addresses, which has a great impact on difficulty of mobility and multihoming (both described in another sections). Borrowing a comparison from John Day's Patterns in Network Architecture, the lack of implicit logical addressing in the Internet feels like accessing memory with DOS instead of Unix.

2.2 Lack of Multihoming

Since the IP addresses serve as point-of-attachment addresses (i.e. one per each computing system interface), there isn't any implicit mechanism for distinguishing whether multiple IP addresses belong to a single node.

TODO: check out the IPv6's pseudo multihoming thing and mention Tinker Air Force Base

2.3 Lack of Mobility

Since a host location is determined by its IP address and IP addressing is geographically dependent, mobility is essentialy non-existent.

TODO: just a placeholder, re-read the relevant chapter of Patterns

2.4 Lack of Multicast

TODO: better get well informed on this one first

2.5 Weak security

The specifications of the fundamental protocols of TCP/IP stack – IP, TCP and DHCP – were originally finished at the beginning of 1980s. Computer internetworking in pre-Internet era was still a matter of closed research exercised in small scientific communities, so most of the design effort was given to making things work and network security wasn't the main aim. However, the Internet has since then turned into a massive world-wide internetwork connecting people of different types and agendas. Naturally, once the Internet began to be used for transferring sensitive data (especially by companies), cyber crime started to emerge as well and some attention was turned to security aspects of Internet protocol (or lack thereof).

Over the decades, a large amount of security flaws has been discovered and continually exploited. Some examples of these are Denial-of-service attacks (ICMP flood, SYN flood, CAM flood), Man-in-the-Middle attacks, IP address spoofing, DHCP spoofing, ARP spoofing and ARP cache poisoning. The requirement of every secure network is to carefully mitigate all of them.

2.6 Router Table Size Growth

Default-free zone (DFZ) is the collection of all Internet autonomous systems (AS) that do not require a default route to route a packet to any destination. Since they comprise the root of the Internet's routing infrastructure, their database must be complete.

With the increasing number of hosts connected to the Internet, the DFZ routing table sizes grow as well. While the exponential growth observed during the 1990s was mitigated in 2001 by mass deployment of CIDR and route aggregation, the number of items is still increasing linearly and the high-end router hardware needs to keep up, especially with increasing use of IPv6 and BGP-based multihoming. This can sometimes lead to scalability problems, as in August of 2014 where reaching the 512k entry limit of multiple routers caused world-wide outages.

As of year 2015, the Internet routing table is consisted of over 560k entries, which requires at least TODO MB in a router's memory. It is commonly believed that Moore's law will ensure that the technology of high-end routers will keep scaling along with the increased demands, but recent research proves otherwise.

TODO: revise the exact data

Chapter 3

Overview of Alternative Architectures

(˜4 ns)

This section gives an overview of paths that were pursued in the field of network architecture research.

3.1 Design Approaches

Over the past several years, the networking research community has exhibited many attempts of moving the field forward. The undertaken research directions are often classified into one of two groups:

- evolutionary design: backward-compatible solutions that are incrementally deployable on top of the current Internet (e.g. DiffServ, IntServ, LISP), or
- clean slate design: designing completely new standalone architectures that aren't constrained by Internet technology's limitations (e.g. RINA, NDN)

Considering the scope of this thesis, the focus will be given exclusively to c̈lean-slate designärchitectures.

3.2 Research Initiatives

3.2.1 NewArch Project

DARPA-funded, nada.

3.2.2 Future Internet Architecture

In 2010, National Science Foundation funded five projects as a part of this program: Named Data Networking, MobilityFirst, NEBULA, eXpressive Network Architecture and ChoiceNet.

3.2.3 FIRE

3.2.4 FIBRE

3.3 Information-centric networking

3.3.1 CCN

3.3.2 NDN

3.3.3 GreenICN

3.4 Recursive Architectures

3.4.1 RNA

3.4.2 RINA

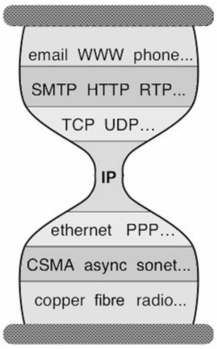
Chapter 4

Named Data Networking

(˜8 ns)

4.1 Premise

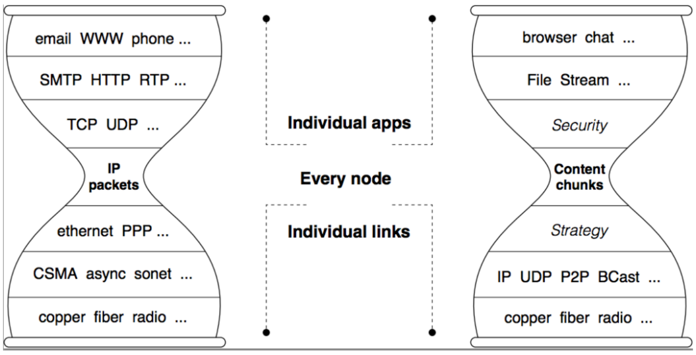
As described in section 2.1, the current Internet has its roots in telecommunication technology as this was the only area from which any inspiration could be taken. Because of this, technology of the Internet has been built on the paradigm of host-to-host communication. This can be visually presented on a „hourglass model“, which indicates that while there's a wide array of technologies in use in the lower and higher layers, the hourglass's thin waist of end-to-end communication is the key static part binding different networks together.



Building the Internet as a communication network was an obvious choice, especially since most of communication of the early ARPANET consisted of connecting terminals to mainframes and executing remote procedures on them.

However, while the basic paradigm of host-to-host communication hasn't changed, the way we use the Internet has gone in an entirely different direction: the Internet has become mostly a content distribution network. Since the mechanism of communication over the Internet is based on creating and maintaining end-to-end connections, this creates an enormous amount of data redundancy.

Named Data Networking proposes a solution for the problem: instead of working with the source/destination node identifiers, the „thin waist“ of the Internet should work with names of data chunks.

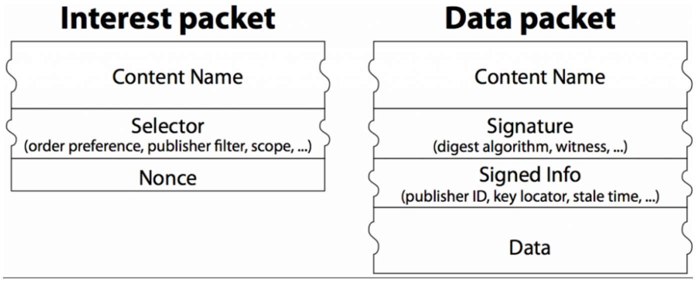


4.2 Concepts

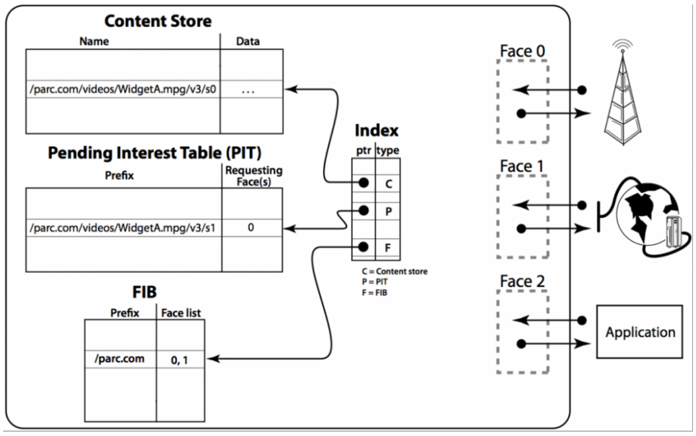
4.2.1 The Building Blocks

The NDN architecture specifies:

- two types of packets: an interest packet (containing the name of desired data) and a data packet (containing the requested data)


Interest packet
Content Name
Selector
(order preference, publisher filter, scope, ...)
Nonce
Data packet
Content Name
Signature
(digest algorithm, witness, ...)
Signed Info
(publisher ID, key locator, stale time, ...)
Data

- two types of hosts: a consumer (data requester) and a producer (data provider)
- a router maintaining three fundamental data structures:
  - Forwarding Information Base (forwarding table)
  - Pending Interest Table (maintaining currently active requests)
  - Content Store (data cache)



4.2.2 The Communication Model

Communication in NDN is driven by the data receiver, i.e. consumer.

1. The consumer sends out an „interest packet“ containing the name of the desired data.
2. When a router receives the interest packet, it first consults its Content Store for requested data.
    - If the data requested by the interest packet are present, they are returned in the direction of the requesting interface.
    - Otherwise, it'll look up the Pending Interest Table.
        - If there's an entry present for the named data request, the entry is updated by adding the originating interface into the list of requesting interfaces, thus aggregating the new request together with an existing one.
        - Otherwise, a new entry is inserted, a FIB lookup is made and the interest packet is forwarded to interface(s) returned by the FIB.
3. A data packet is returned to the router by either the producer or another router with cached data. The router finds a matching PIT entry and forwards the data to all interfaces listed in the PIT entry. The PIT entry is then removed and data are cached into the Content Store.

4.3 Implications

4.3.1 Data Caching

The most significant feature of NDN is its native support for caching all sorts of data inside the network itself. While this should be beneficial mostly for static data such as web pages and images, dynamic content could take advantage of this as welll in case of multicasting or packet retransmission on packet loss.

4.3.2 Routing and Forwarding

NDN routes and forwards packets on names, which eliminates four problems caused by address forwarding in the IP architecture: address space exhaustion, NAT traversal, mobility, and address management.

- There is no address exhaustion problem since the namespace is unbounded.
- There is no NAT traversal problem since a host does not need to expose its address in order to offer content.
- Mobility, which requires changing addresses in IP, no longer breaks communication since data names remain the same.
- Finally, address assignment and management is no longer required in local networks, which is especially empowering for embedded sensor networks.

Since the forwarding mechanism otherwise bears a strong resemblance to the forwarding mechanism in IP networks, lot of the existing research on IP routing could be applied to NDN as well: protocols like BGP, IS-IS or OSPF can be, for the most part, reused with minor modifications (e.g. routers would announce name prefixes instead of IP prefixes).

4.3.3 Security

While TCP/IP with its end-to-end communication paradigm relies on setting up a secure channel between two hosts and transmitting data through it, NDN requires each data chunk to be signed together with its name. Thus, since security is built into the data itself, there's no need for a direct host-to-host secure channel to be created.

4.4 Current State of Implementation

Chapter 5

Recursive InterNetwork Architecture

(˜8 ns)

5.1 Premise

In 2008, computer scientist John D. Day has published a book that marked the culmination of his long-time goal of rediscovering the way we think about computer networks. The book was called Patterns in Network Architecture and in it, Day singlehandedly proposed a clean-slate approach to computer architecture that aims to get rid of most of TCP/IP's drawbacks.

The book can be roughly divided into two separate parts: in the first part, Day attempts to decompose mechanisms used in TCP/IP to their basic parts and put them into historical and socio-economical context. He eventually discovers that the currently prevalent layered approach to network architecture is needlesly complex, because each layer consists of the same mechanisms acting on a different scope.

The second part takes into account all of the facts discovered in the first part and uses them as foundations to build an entirely new concept of network architecture from scratch: the recursive IPC model (originally Network IPC Architecture, NIPCA).

RINA (Recursive InterNetwork Architecture) is a currently researched network architecture based on fundamental principles described by Day.

5.2 Concepts

5.2.1 Error And Flow Control Protocol

5.2.2 Relaying and Multiplexing Task

5.3 Implications

5.4 Current State of Implementation

Chapter 6

Implementation of Relaying and Multiplexing Task

(˜10 ns)

6.1 OMNeT++

OMNeT++ is an open-source discrete simulation framework used primarily in the field of network simulation. In this context, the term „network“ refers to the more general meaning of the word, which means the framework can be used not only for simulation of TCP/IP networks (especially in conjecture with the INET library), but it also provides means for simulating other networked systems such as on-chip networks or queuing networks. As we're implementing a clean-slate architecture from the ground up, this is an ideal approach.

OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using the high-level language NED. In theory, there are no limits for networks modelled by NED and the only constraint is given by a computing platform processing power.

6.2 RINASim

RINASim, developed by networking research group at Faculty of Information Technology of Brno University of technology, is an OMNeT++ library developed for project PRISTINE. The purpose of the library is to provide a framework for building RINA networks. RINASim is used primarily by other PRISTINE researchers to experiment with the architecture and efficiently evaluate their working theories.

6.3 Implementation Design

In RINASim, all functionality of RMT including a policy architecture is encompassed in a single compound module named „RelayAndMux“ which is present in every IPC process. The module serves for (de)multiplexing, relaying and aggregating PDUs of data flows traversing the IPC processes.

6.3.1 Separation of mechanism and policy

Some aspects of RMT functionality such as queue scheduling and performance monitoring are, by their nature, suited for separation of mechanism and policy. In another words, we need to take into account that architecture implementators will probably want to specify custom behavior for some of the logic present in RMT.

For this purpose, several algorithms used by RMT are meant to be user-configurable:

- RMT scheduling policy: The scheduling algorithm that determines the order input and output buffers are serviced (e.g. by number of waiting PDUs in buffers or with fair queuing).
- RMT monitoring policy: A queue monitoring algorithm that is invoked each time a PDU enters or leaves a queue. This policy should compute variables (such as average queue length) to be used in decision process of other policies.
- RMT maxqueue policy: An algorithm that is invoked each time a number of PDUs waiting in a queue exceeds the queue's threshold. This policy is mostly used for congestion control mechanisms (e.g. by dropping or marking the last PDU in a queue).

6.3.2 Module structure

(the nasty guts of RelayAndMux.png)

RMTModule consists of multiple simple modules of various types, some of which are instantiated only dynamically at runtime.

Static modules:

- RMT, the central logic of Relaying And Multiplexing task that decides what should be done with messages passing through the module.
- RMTModuleAllocator, a control unit providing an API for adding and deleting instances of dynamic modules (RMTQueue, RMTPort).
- SchedulingPolicy, a scheduling policy of an RMT instance.
- MonitorPolicy, a monitoring policy of an RMT instance.
- MaxQPolicy, a maxqueue policy of an RMT instance.

Dynamic modules:

- RMTPort, a representation of one endpoint of an (N-1)-flow.
- RMTQueue, a representation of either input or output queue. The number of RMTQueues per RMTPort is determined by Resource Allocator policies.

Chapter 7

Testing and Evaluation

(˜4 ns)

Chapter 8

Conclusion & Future Development

(˜1 ns)

Appendix A

CD Contents

Appendix B

Installation and Usage