Introduction to Content Centric Networking and the CCNx framework

Thibault CHOLEZ

Lab of the 6th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2012)

Interdisciplinary Centre for Security, Reliability and Trust

University of Luxembourg, Luxembourg

04/06/2012

First words

A new way to look at networking !

- "Content Centric Networking" (CCN) : proposal for an alternative paradigm to the current architecture of computer networks mainly based on TCP/IP.
- Initially developed at PARC by Van Jacobson.

The key idea

- Democratize Content Distribution and re-design the Internet by placing content, and not machines, at its core.
- Argument : *named data* better abstraction than *named hosts*.

Plan

Plan

Why a tutorial on Content Centric Networking ?

A trending topic : one idea, many names

- Content Centric Networking [JST+09]
- Named Data Networking [ZEB+10] (NSF project)
- Information Centric Networking [GSK+11]
- Data Oriented architecture

A great source for research and innovation

- Many opened questions (security, interoperability, etc.)
- Many possible improvements in the core design
- Design of new services
- Some opened/philosophical questions about the future of Internet

Inventors

Van Jacobson

- Ambassador of CCN, one of the most famous scientist in networking
- Lots of major past contributions solving Internet congestion issues :
  - "Congestion avoidance and control", Van Jacobson, Proceedings of SIGCOMM ?88, Stanford, CA, Aug. 1988, ACM
  - Many tools : traceroute, tcpdump, etc.
  - RTP protocol

Palo Alto Research Center (PARC)

- Since 1970, Research center in information technology and hardware
- Many notable achievements :
  - Graphical User Interface, WYSIWYG text editor
  - Laser printers, PostScript language
  - Object-oriented programming (Smalltalk)
  - Ethernet

Motivation : networking then and now

Initial context of networking

- Few very expensive computers, very few data
- Need to access computers to execute jobs (resource sharing)
- Focus on point to point communications

Today

- Lots of cheap computers, many devices per user
- Everything is connected
- Lots of data to be retrieved, personal content to be sync
- Upcoming Internet of Things : ubiquitous, wireless
- Focus on content dissemination (90%) [SM09]

Internet is not bad by design, the problem has changed.

Motivation : today's Internet

Not well adapted to Content Distribution

- Conversation (two machines communication) inefficient for content dissemination
- Lack of multicast support : huge waste of resources (bandwidth, power)
- Limited mobility (rock stable IP addresses : connection lost when moving)

How to distribute contents over the inefficient Internet ?

After-thought solutions and overlay networks :

- Web caching
- Content Delivery Networks (CDN : Akamai, etc.)
- P2P networks (Bittorrent, KAD, etc.), P4P, etc.

Motivation : today's protocols and services

The less TCP/IP answer to today's communication needs, the more additional protocols appear. More protocols means :

- more overhead, configuration, standardisation
- less interoperability

No more real seamless connectivity ?

- Inefficient addressing : DNS, IPv6
- Inefficient service discovery : no real standard, many protocols
- Inefficient security : VPN, IPSec, SSL, etc. secure the conversation not the transmitted content (ex : spam)

Objectives of this tutorial

The bright way to speak about CCN : like Copernican revolution !

- CCN places content, and not machines, at the center
- With the right point of view, everything appears simpler
- Similar to placing the sun, and not the earth, at the center

Not the first revolution : from data over telephony to packet switching (focus on paths vs focus on endpoints)

Introduce you to the networking Copernican revolution

With the key to understand :

- Architecture (Routing, Naming), Management and Security, Deployment
- Facilities for experimentations (CCNx framework)
- With a research perspective : principles, limitations and sota

Objectives of this tutorial

The right time to begin working on CCN

Storyline :

- 2006 : First idea, Van Jacobson talk at Google
- 2009 : Major Paper on CCN at CoNext 2009
- 2010 : CCNx community providing tools for experimentation
- 2011 : Increasing number of research papers (Infocom NOMEN 2012 and SIGCOMM ICN 2011 workshops)

Plan

Content Centric Networking

Placing the data at the center

TCP/IP are blind to the data :

- No security of data
- No efficient transmission of data to multiple endpoints

Data storage (hard drive) is proven cheaper than bandwidth

Key ideas

- Data as a name, not a location
- Data is directly requested at the network level (not its holder, no more DNS : more security ; less delay)
- Anybody with the data can answer
- Data is authenticated (and secured), not the connections it traverses
- Rely on close data storage (*caching*)

CCN layer model

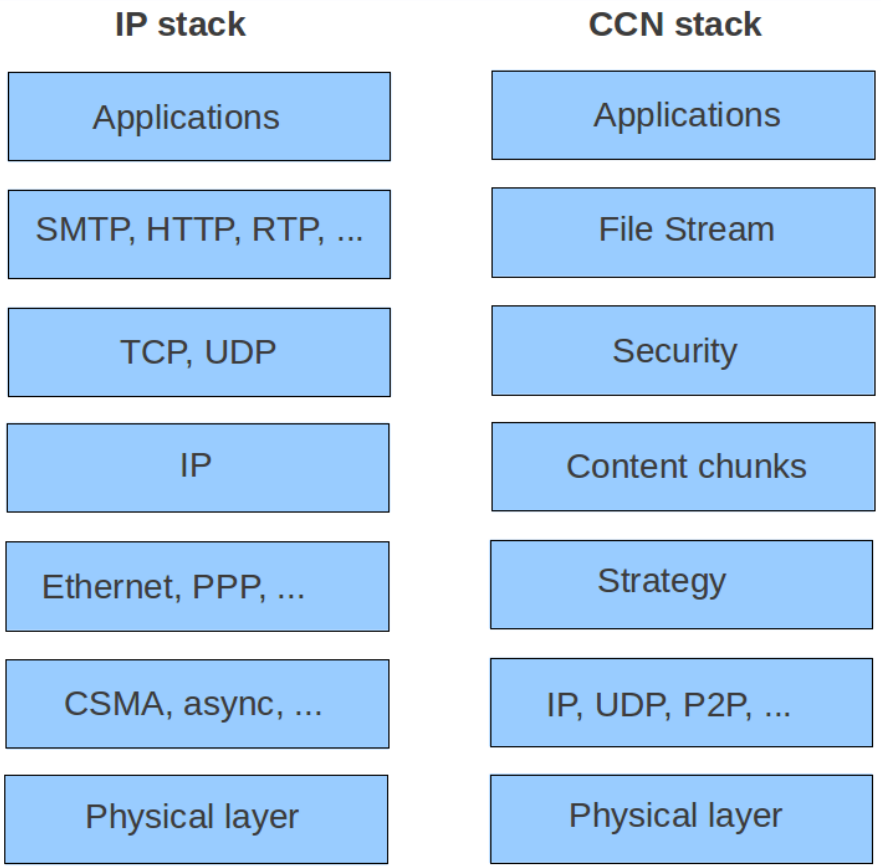

FIGURE: IP and CCN protocol stacks

CCN packets

Pull based model

- 2 types of packets *Interest* and *Data*
- Consumer driven : broadcast Interest, waits for Data
- Data transmitted in response, consume the Interest

Security

- Trust in the content not in the way we got it
- Request based routing, no classical DoS

CCN packets

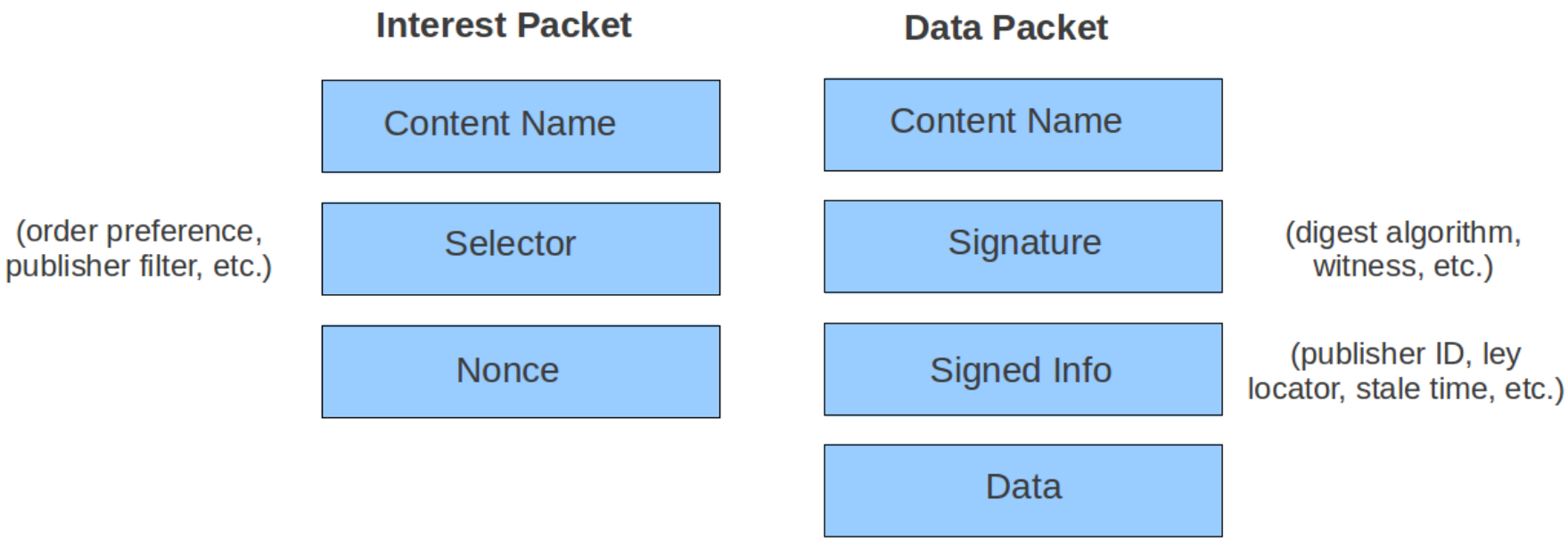

FIGURE: Two types of packets : interest and data

Routing tables

Forwarding Information Base (= IP Routing Table)

- Routes to forward Interests to Content source
- Several outgoing faces per destination (no spanning tree)

Pending Interest Table

- Keeps track of forwarded Interests (from different faces)
- Data follow Interests back to the requester, Time out if no Data

Content Store

- Content chunks still valuable after forwarding
- Used as buffer memory to save bandwidth, all node provide caching
- Several caching policies possible (LRU, LFU)

Table lookup priority : CS $>$ PIT $>$ FIB

Core of a CCN node

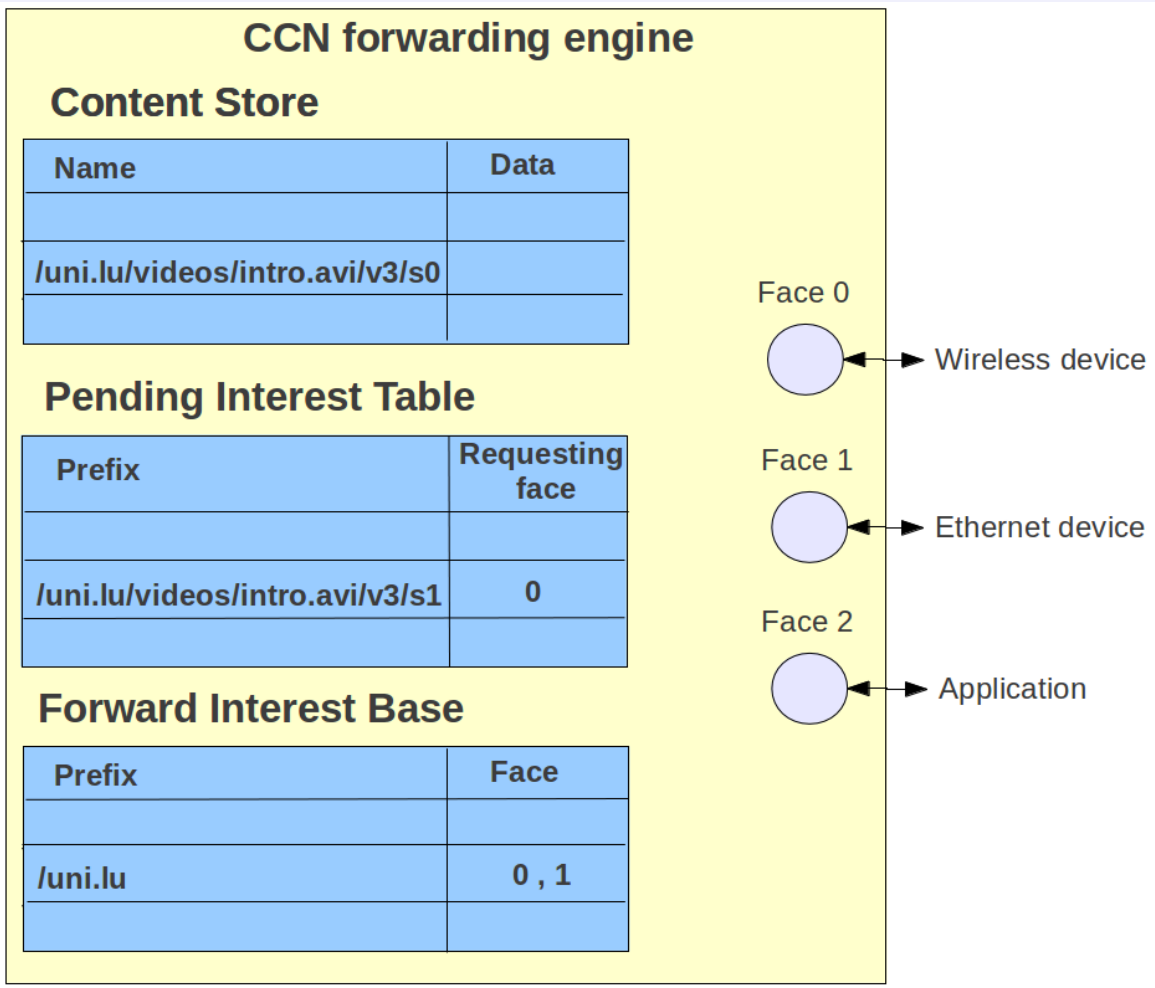

FIGURE: CCN forwarding engine

Hierarchical naming

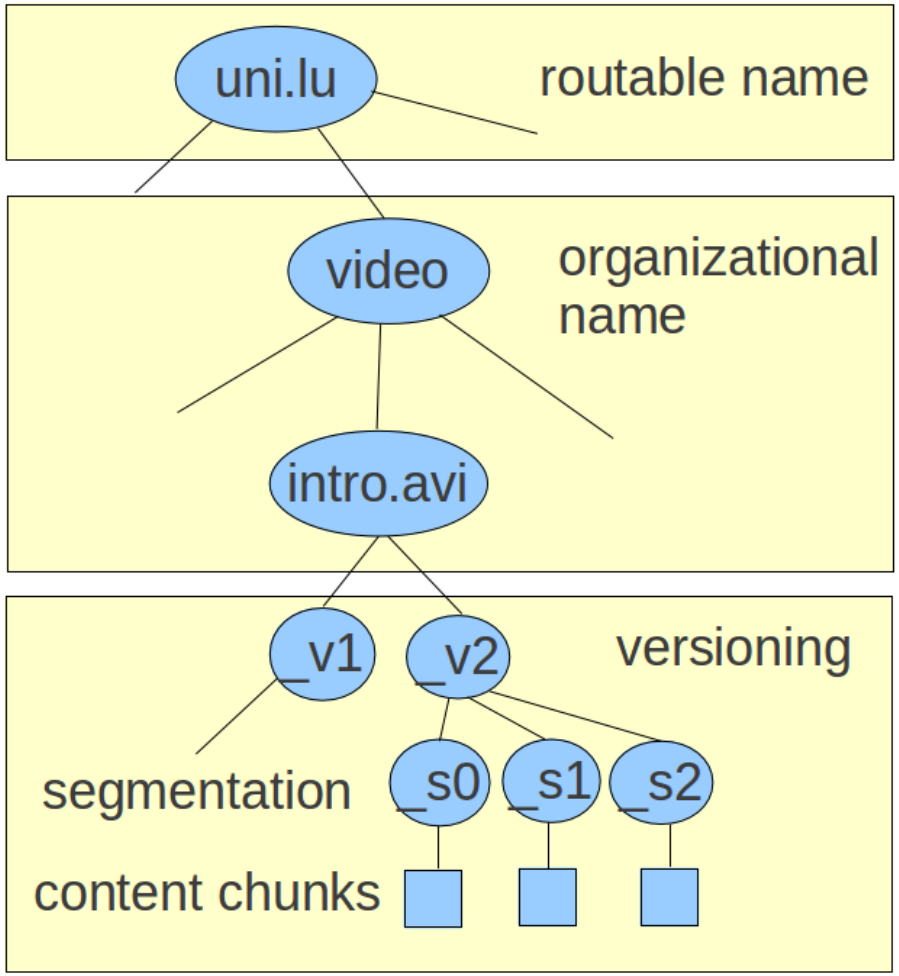

FIGURE: Hierarchical naming of a CCN content

Hierarchical naming

Structure of CCN content names

- Plain-text name : intuitive names, no crypto digest, no need of indirection mechanism between names and content (DNS, DHT)
- Routable prefix, persistence of the names
- Hierarchical aggregation for fast routing and forwarding
- Relative navigation : *RightmostChild* / *LeftmostChild*
- Dynamic generation of content
- Context-aware resolution (*/ThisRoom/projector*)

Coding

- Names divided into components, divided into octets
- 5[6[uni.lu]6[videos]9[intro.avi]7[FD04A...]2[0003]]

Message sequence

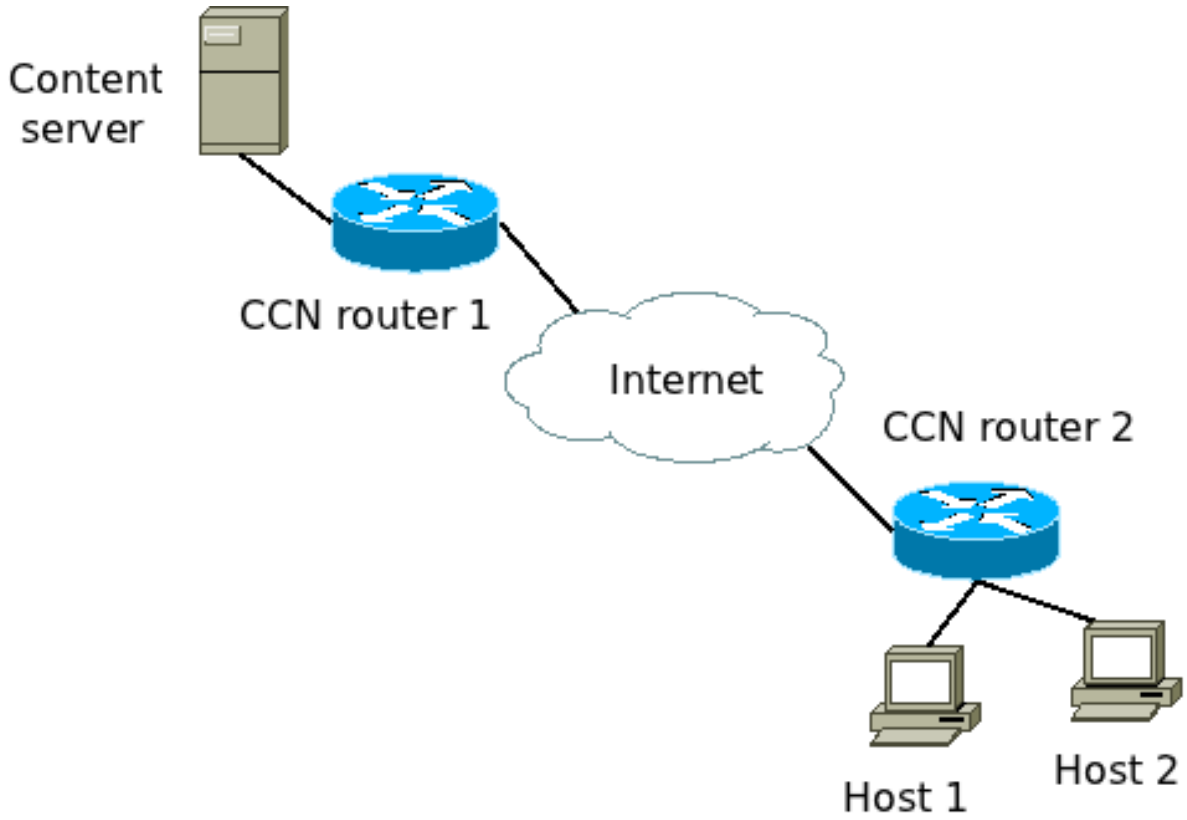

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

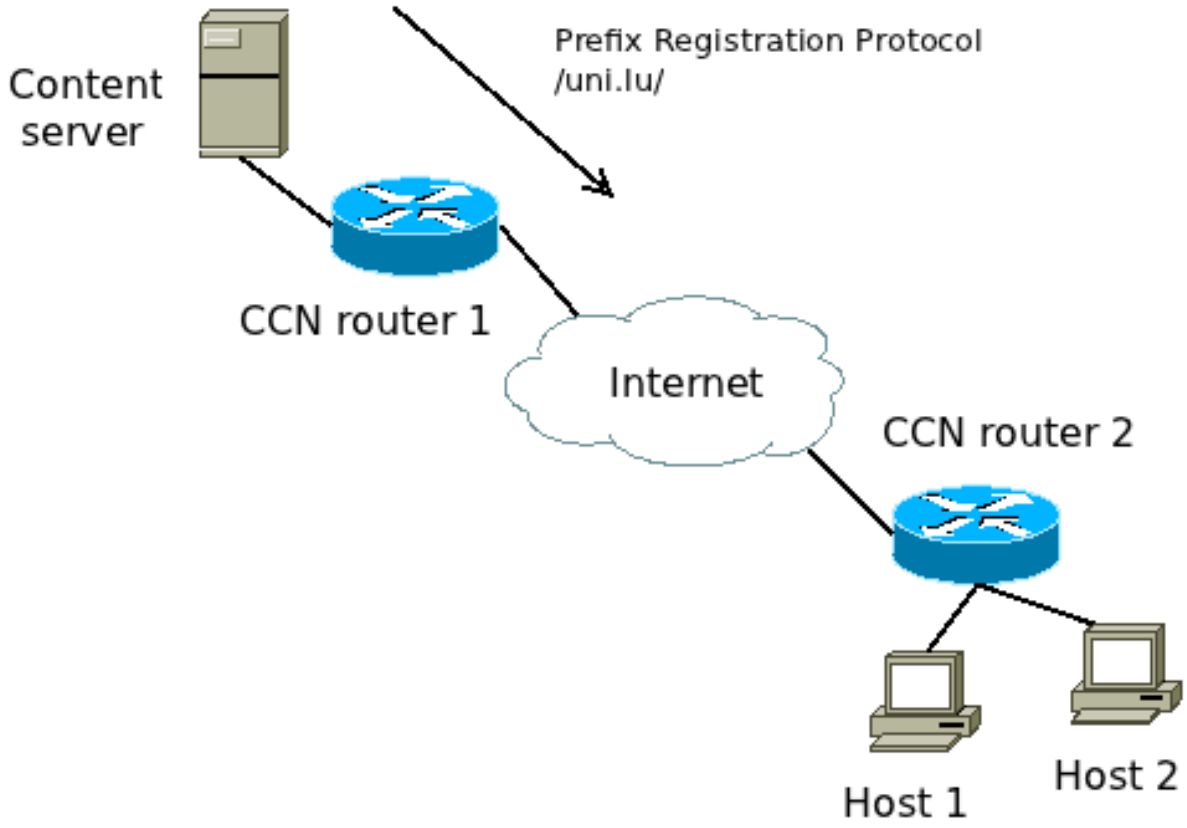

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

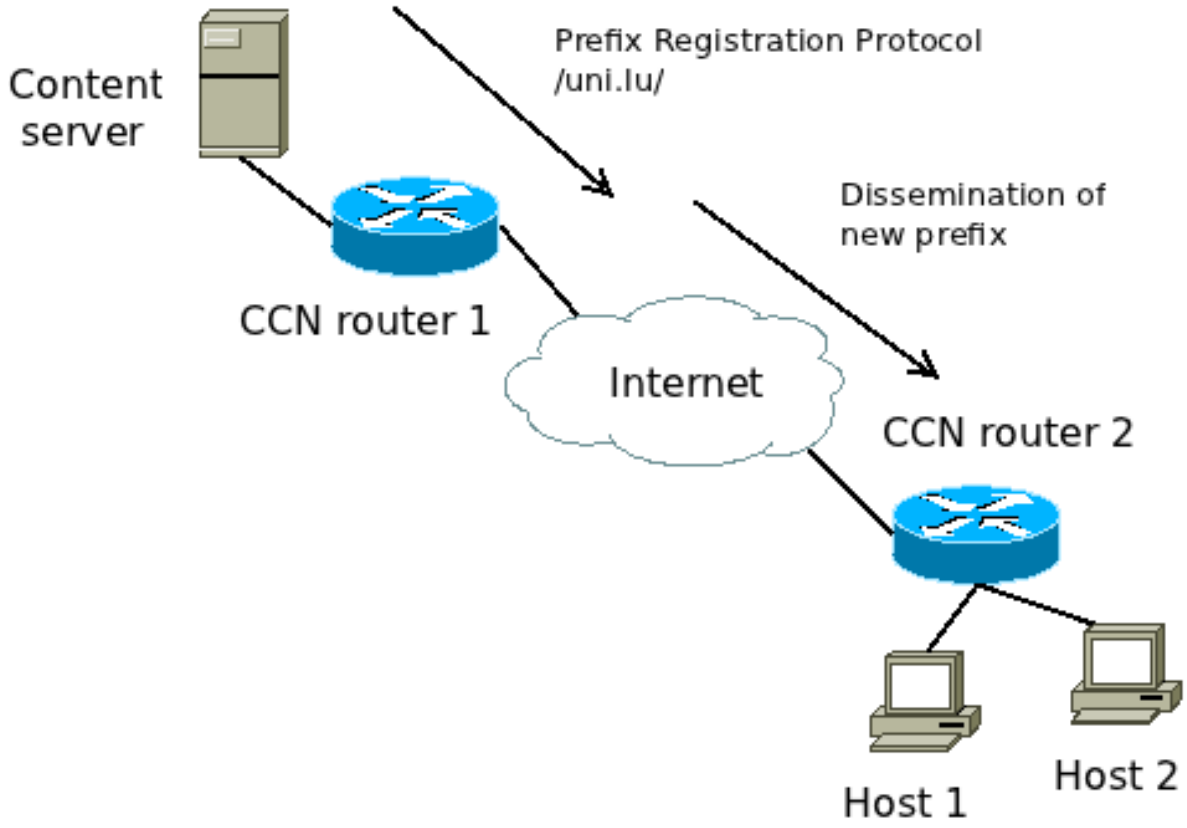

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

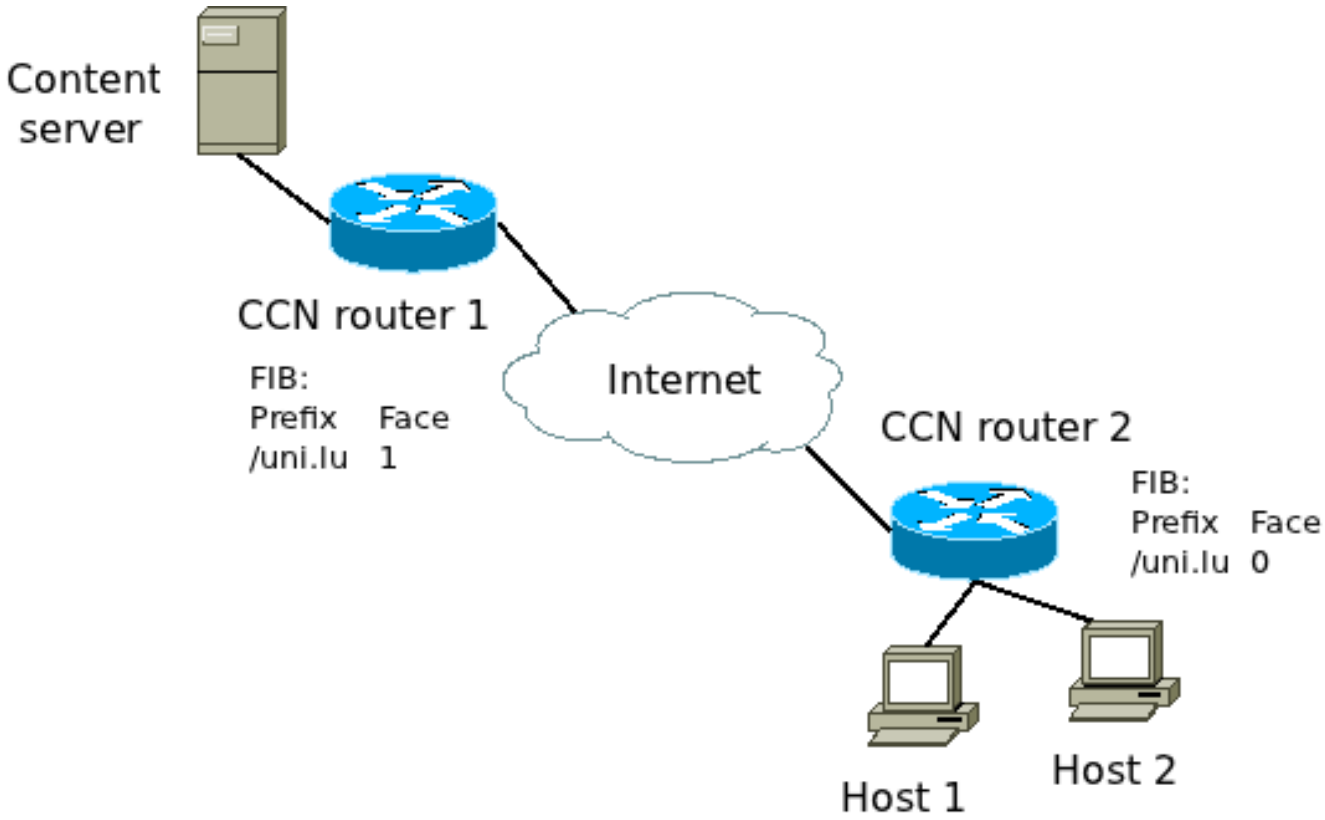

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

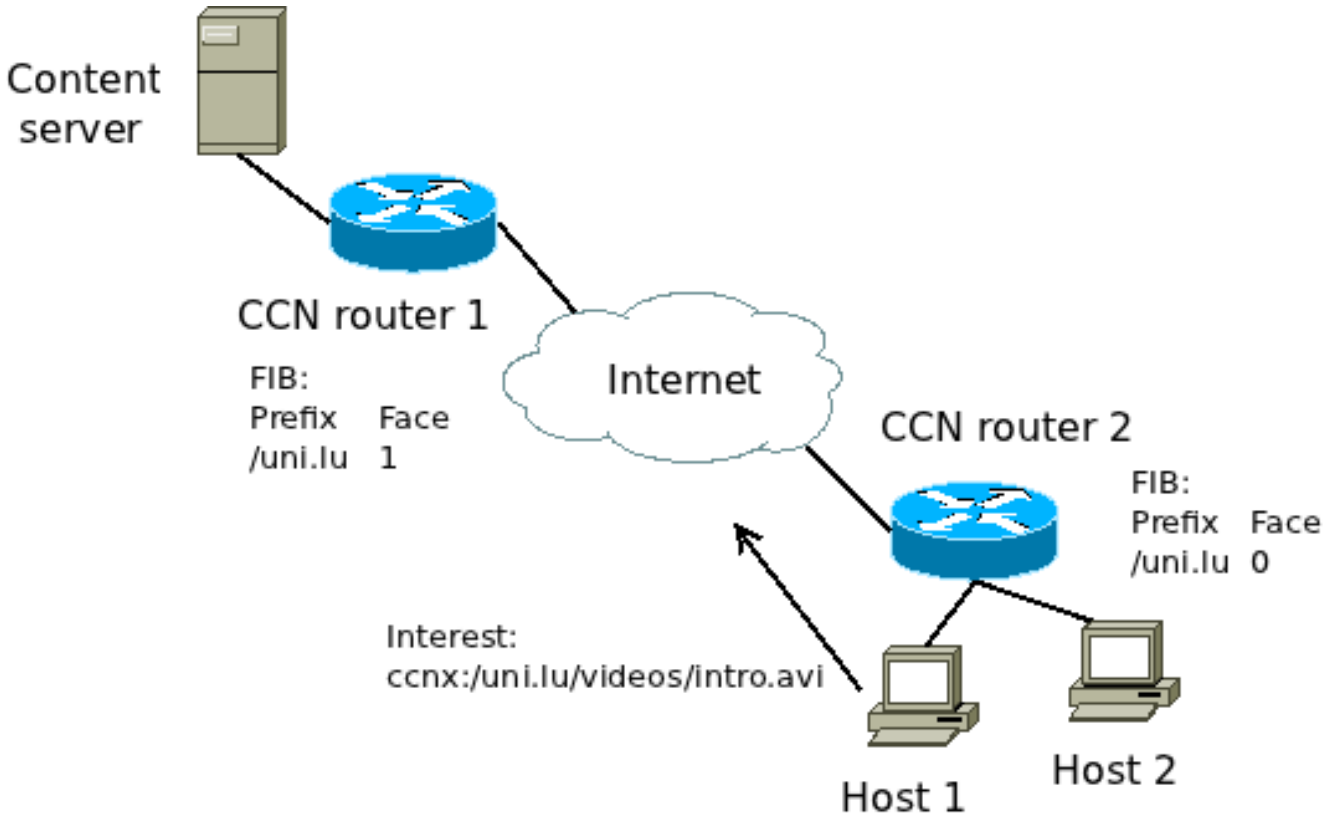

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

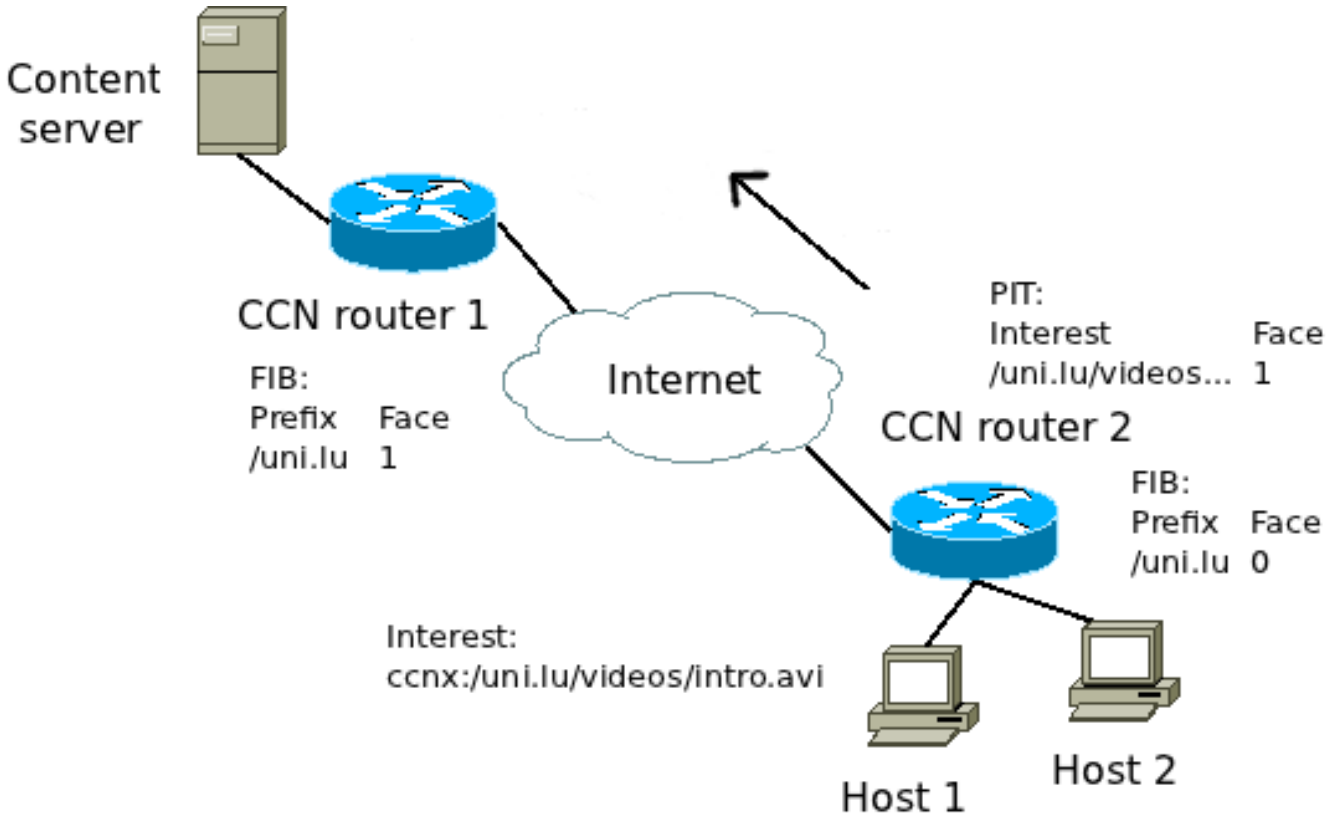

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

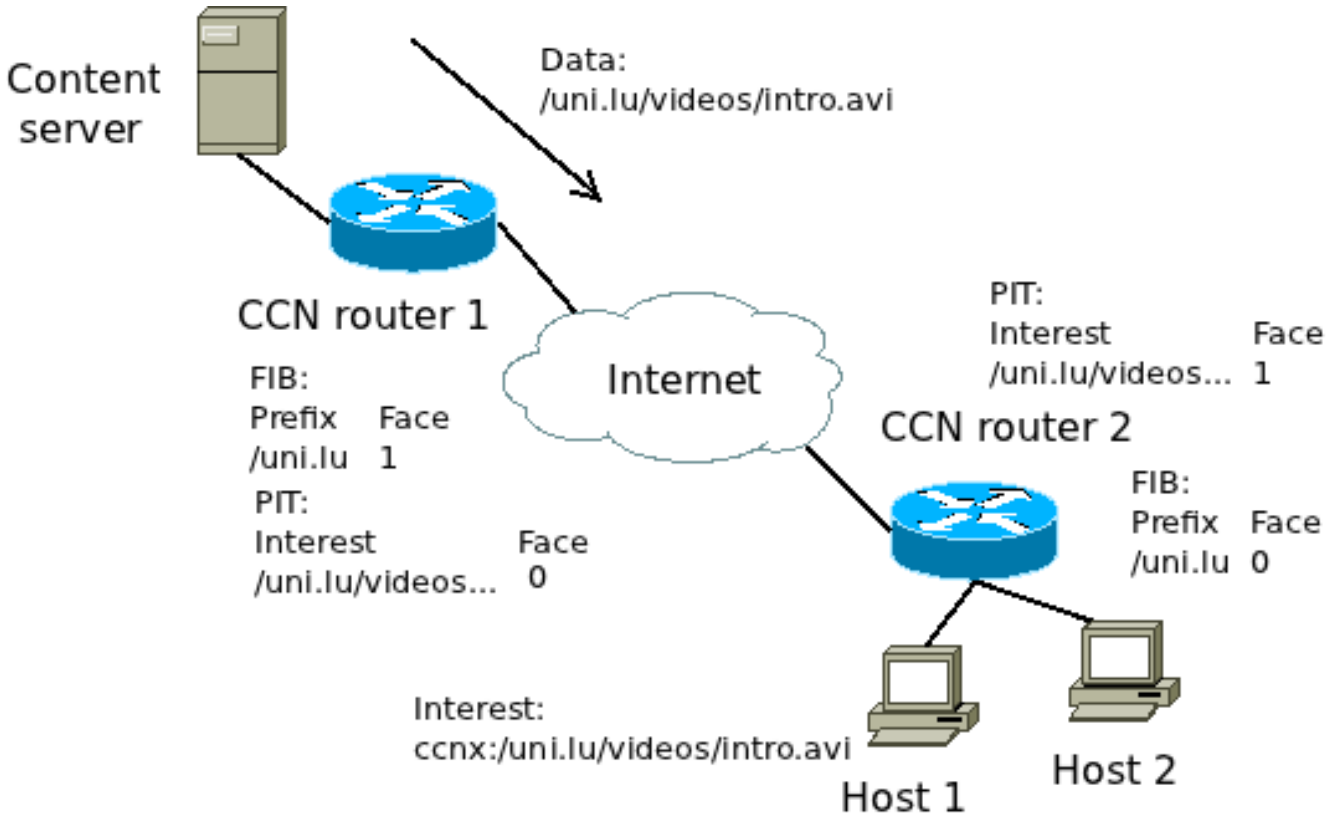

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

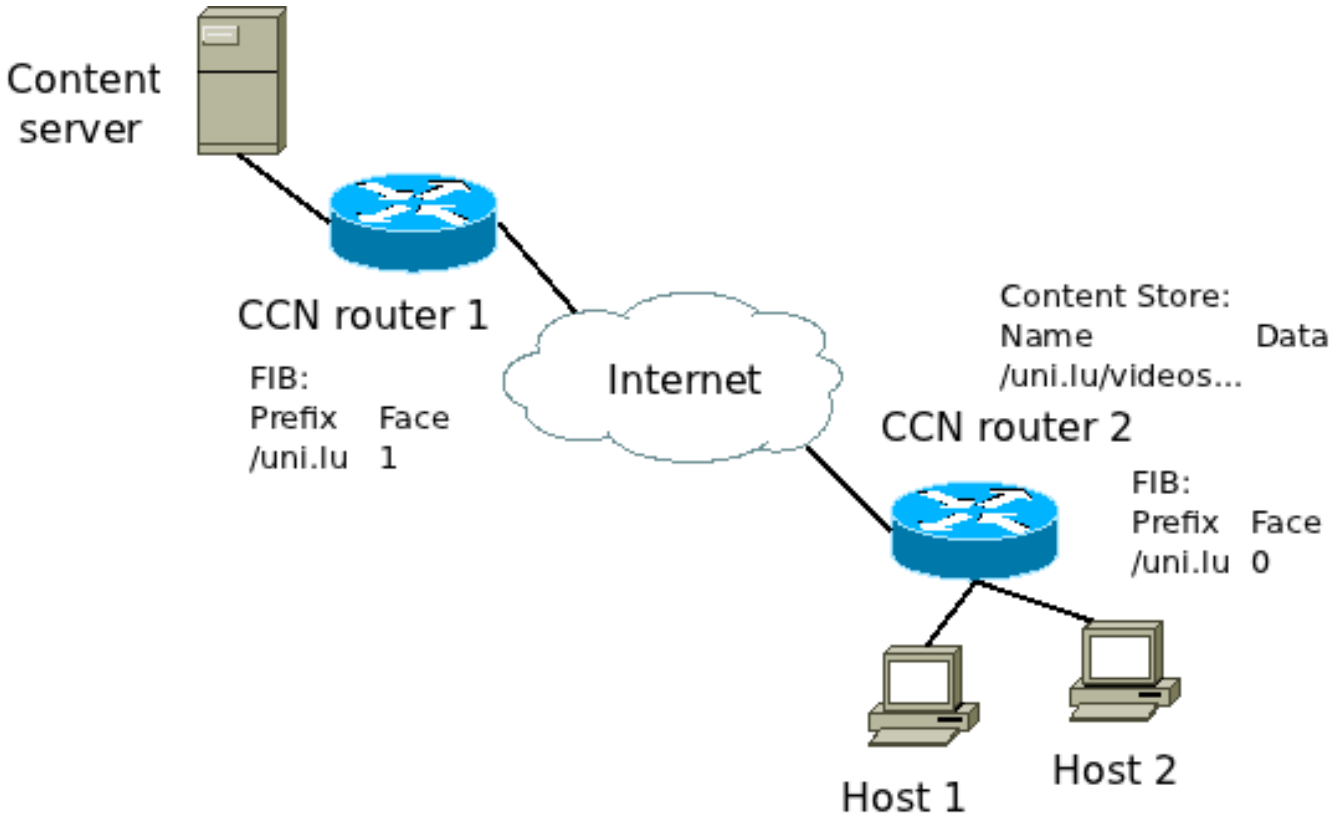

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

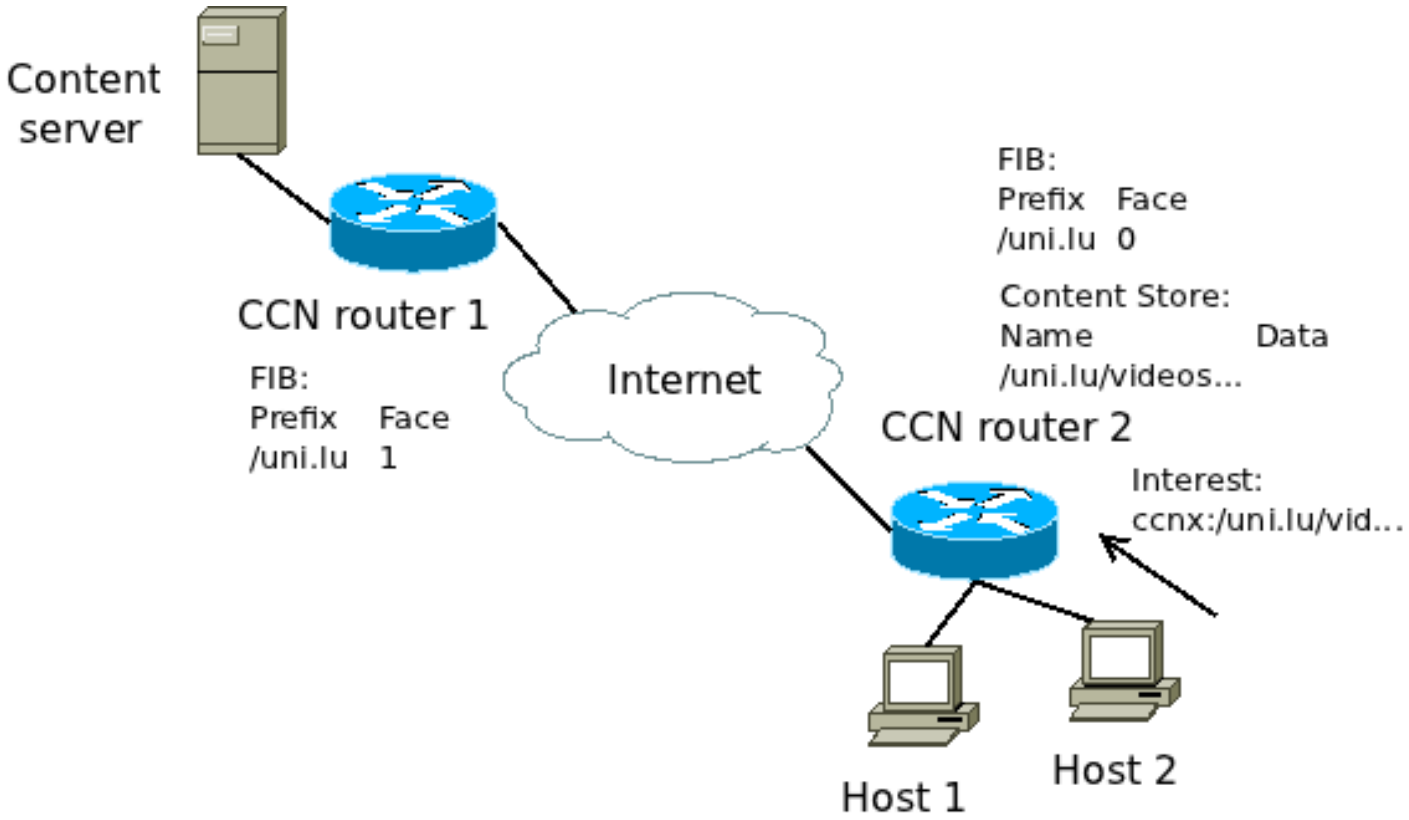

FIGURE: Scheme of a minimal Content-Centric network

Message sequence

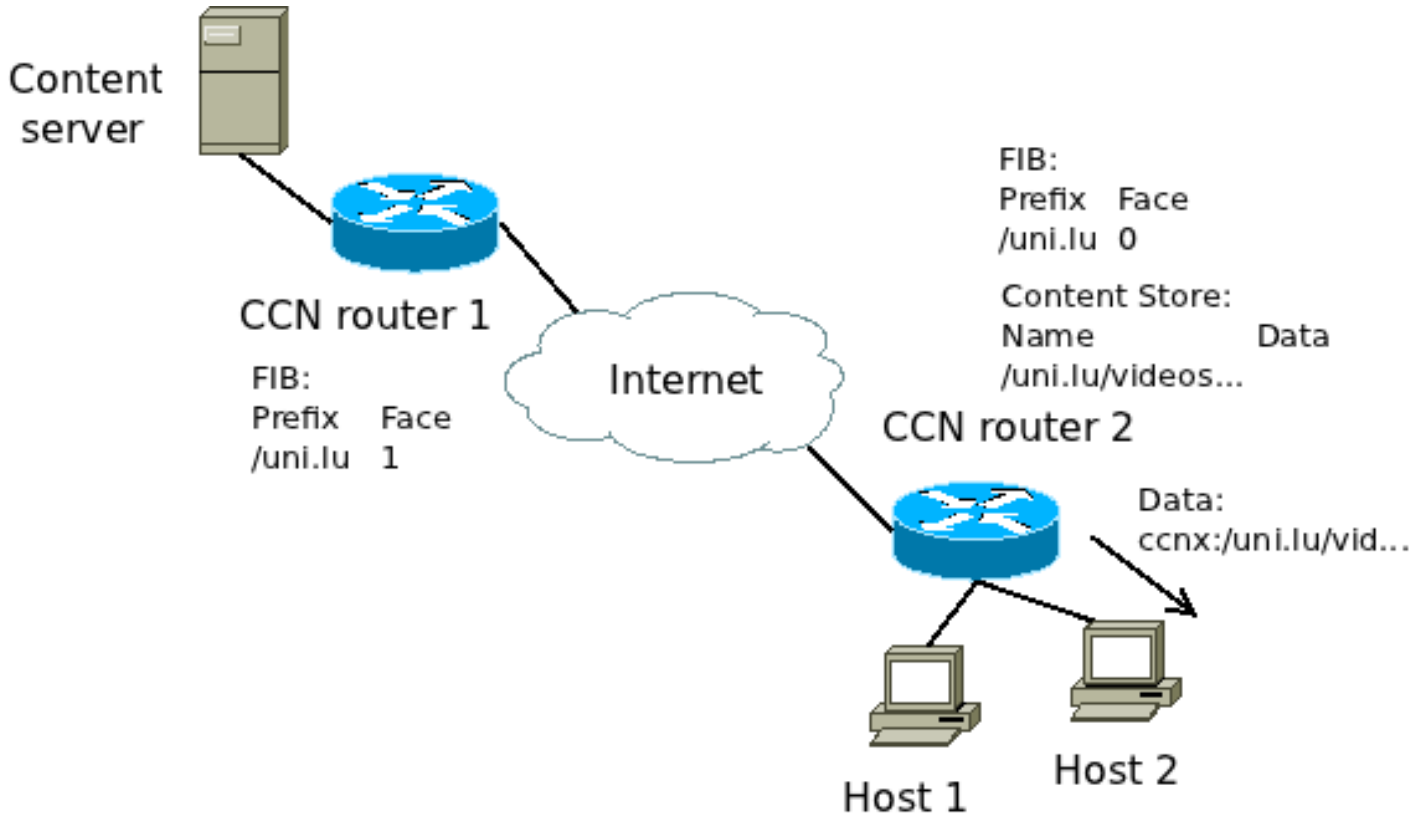

FIGURE: Scheme of a minimal Content-Centric network

Content Centric Networking

Example : getting a web page

```
REQ http://nytimes.com/today ->
<-  RESP http://nytimes.com/today
http://nytimes.com/20120406/index.html
<NameMAC 3fde... />
<DataMAC 07a2... />
<html>
...
<html/>
```

Key points

- Content is signed by the initial provider
- Content is bind to its name
- Content can be dynamically generated

Content Centric Networking

Transport reliability and control

- Network transport content not conversation, popular content does not create congestion
- Request / Response model gives user control of QoS (congestion, priority)
- CCN Interests control the flow like TCP ack
- A group of Interest can be emitted : equivalent to window advertisement
- Hop by hop congestion control
- Failed Interests are retransmitted

Content Centric Networking

Mobility

- No binding needed between layer 2 and CCN layer (named content)
- No more node addressing problems (multiple interface support, easy mobility)
- Multiple interfaces managed by the Strategy Layer

Strategy

- Precise performance evaluation : per-prefix, per-face
- Multiple interfaces managed by the Strategy Layer, for each FIB entry :
  - Actions : sendToAll, sendToBest, etc.
  - Triggers : interestSatisfied, interestTimedOut, faceDown, etc.
  - Attributes : broadcastCapable, isContentRouter, etc.

Point to point communications

Problem

- Point-to-point communications still needed (ex : VOIP)
- How to push data in CCN ?
- Solution : both users reachable under a unique prefix (/domain/user)
- Pushed information encoded in Interest name

Example : voice call

VoCCN

PARC's proof of concept : VoCCN [JSB+09]

- No need of SIP proxy
- Alice encodes SIP invite and emits Interest to call Bob (/domain/bob/encoded-invite-msg) = *on demand publishing*
- Bob sends SIP response in Data msg
- RTP data stream in CCN data, *a priori creation of Interest names*
- Interests name : "/domain/user/call-id/rtp/seq-number"
- Limitations :
  - Useless voice data may be cached
  - Both parties must reachable though unique name

Main questions

CCN architecture and Naming

- How to make domain name globally unique ?
- How to make local data routable (processing of repositories) ?
- Routing algorithm to efficiently locate data
- Incentives to make people distribute content
- Accounting of content
- Which caching strategy (cooperation, size, replacement policy, service differentiation) [GCP11] ?

CCN security

- How to enforce uniqueness of a content name
- Which protocol to check data integrity, name binding
- Revocation of (corrupted) content

Main questions

CCN routers

Not as simple as IP routers

- Work to verify data
- More state in its routing entries, stateful
- More complex entries (names vs IP addresses)
- Associated storage devices

- Potential new DoS attacks
- Increased cost
- Step backward compared to IP routers ?

Plan

Security in CCN

Security strategy

- Digital signature in the core of CCN security, encryption for privacy
- Trust in the content not in the way we got it
- Request based routing, no classical DoS

Content authentication

- Authentication of *binding* between name and content
- Embedded in each CCN data packet :
  - Signature(Name,Content,SignInfo)
  - SignInfo includes : cryptographic digest or fingerprint of publisher's key, key or key location
- Content can be authenticated by every node (public key signatures)
- Different signature algorithm available (security vs performance)

Trust management

Trust strategy

- P2P delivery but end-to-end security : trust in publisher
- Simulating certificate : organization as content name, public key as data
- Content can reflect trust by certifying each other (trust relationship, reduce key management overhead)

Easy key management

- Public key easily retrieved as CCN data
- Publishing a key generates certification
- Content can reflect trust by certifying each other (ex : webpage and its links, images, etc.)
- What PKI for CCN ? SDSI/SPKI : keys are mapped to identities via namespaces = CCN names ; no single source of trust

Trust management through keys

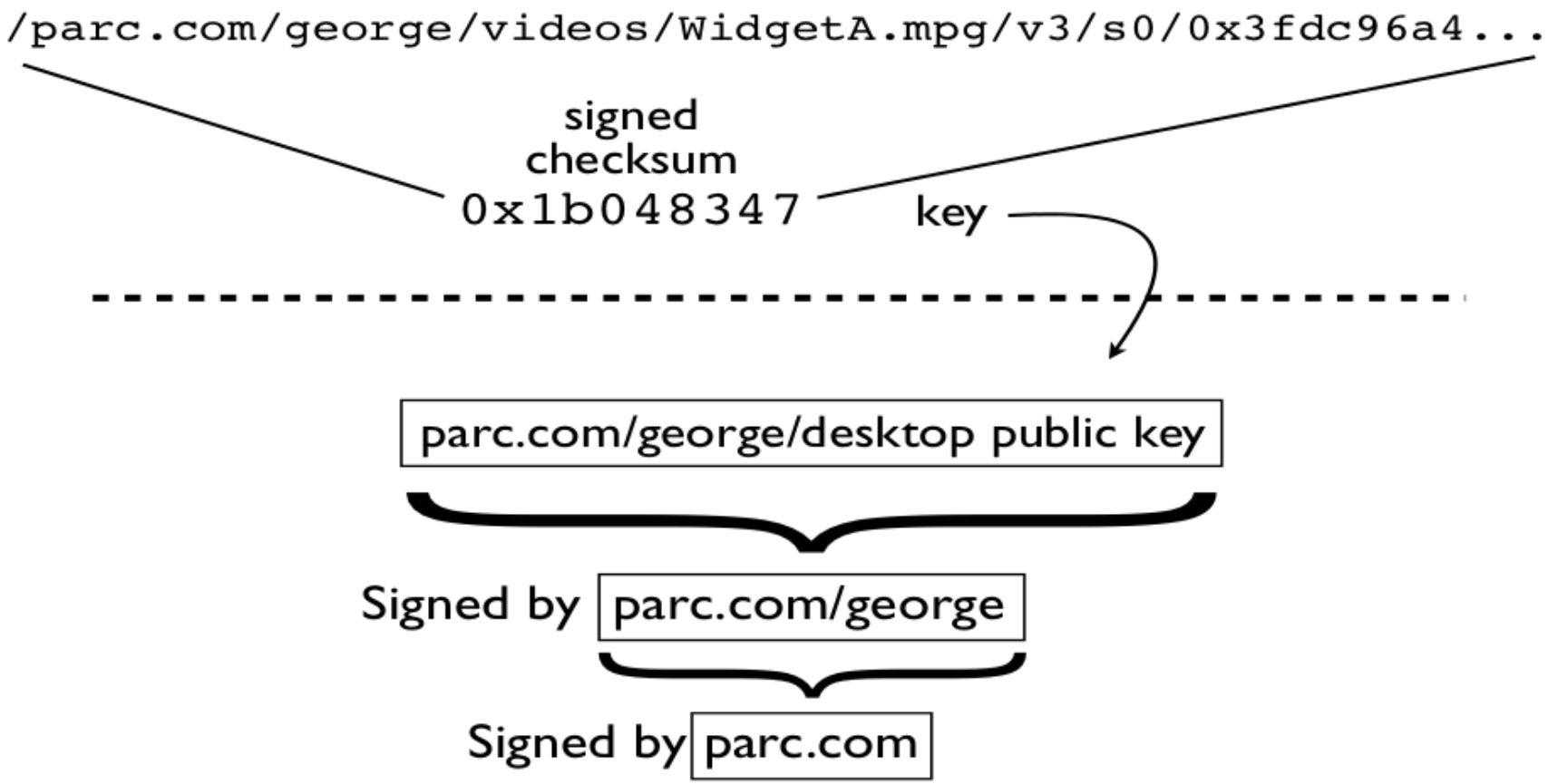

FIGURE: Trust management by associating CCN names with publisher keys[JST+09]

Key management

CCN key management

Open evidence based security, data provenance (traceability) can be checked

- Avoid hierarchical certification (untrustworthy VeriSign issues)
- Individual trust in subtrees, no central authority

Content Protection

- No access control at directory level
- Content protection through encryption and PKI

Network Security

Mitigated network attacks

CCN design improves :

- Host protection : CCN can not directly talk to host, difficult send malicious packets
- Content protection : authenticated content, difficult to spoof
- Content policy : Content firewall based on name prefix or signature
- Aggregation of Data in the network : no DDoS attack

Remaining issues

- DoS : Interest flooding attack
- Possible detection on the path : emitted interest rate vs transmitted data

Secure Naming of Content

Authenticating link between Names and Content

From PARC's technical report : [SJ09]

- Unrestricted names, Publisher certify name for content
- New content creates mapping triple : $M_{(N,P,C)} = (N, C, Sign_P(N, C)$
- With the data must be provided : key of publisher + mapping evidence

Zooko's Triangle

- Choose two properties for names : human-readable, secure, decentralized and unique
- CCN name are Human-readable, secure (mapping)
- CCN name are content locator : must be globally unique, open question

Monitoring of CCN

Purpose

- Collect information and status of CCN nodes
- Detection of anomalies
- Accountability, revocation of content

Challenges

- Adapt SNMP to CCN nodes
- Define relevant information for CCN monitoring
- Define relevant architecture (CCN ready) for collection of information
- Enable collaboration between CCN nodes (network of trust)

Attacking CCN tables

FIB

- Announcing conflicting domains, non-aggregatable names
- Announcing existing content without having it
- Announcing non-existing content

PIT

- Flooding content providers or routers with fake Interests
  - DoS of content provider ? Tracing of query impossible ?
  - Legitimate Interest dropped on router ? Creation of routing loops ?

Content Store

- Decrease caching efficiency by downloading unpopular content
- Spy on network neighbors by probing cache

Security issues : new DoS

Cache Snooping attack

Tobias Lauinger Master Thesis (Eurcom) [Lau10]

- Caching can leak private information
- Exploits users' traces in cache to monitor neighbors activity :
  - Send Interest for the monitored content
  - Compare response time between data response to detect cached content
  - Low traffic cost
- However, private data are encrypted

Security issues

And also...

- Computational DoS of router (heavy signatures, slow key retrieval)
- Revocation of corrupted keys, unwanted or malicious content
- Robustness of current cryptography in the future

From a research point of view

- Creation of a realistic dataset of CCN traffic
- Several applications running on CCNx

Plan

Large scale deployment of CCN

Integration with IP networks

Critical element for CCN adoption

- Anything can run on CCN, CCN can run on anything
- CCN overlay over IP, IP overlay over CCN
- Similarity with IP : hierarchical name aggregation with longest match lookup
- Routing schemes working for IP work for CCN
- Compatible with existing infrastructure, incremental deployment possible

Main question : CCN and IP prefixes being different, are some IP routing protocol compatible ?

Intra-domain Routing

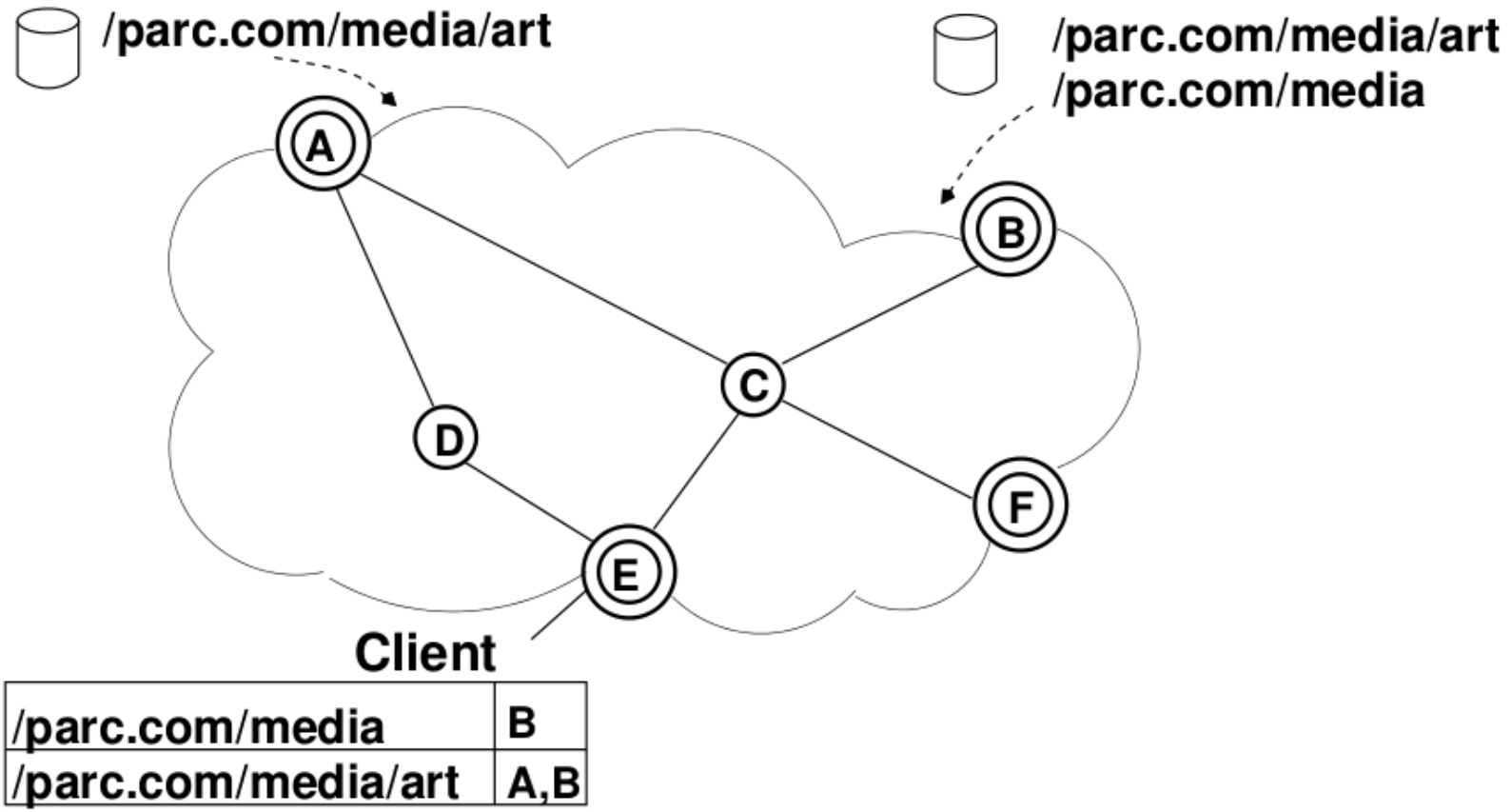

FIGURE: Intra-domain routing in mixed IP/CCN network [JST+09]

Intra-domain Routing

Local connectivity

- Classical protocols : distance vector (RIP, EIGRP) and link state (OSPF, IS-IS) inside AS
- Router Capability tlv (type label value) option for IS-IS or OSPF describes connected resources
- Used to distribute CCN prefixes
  - Media repository announce available content with full name
  - CCN capable router create a FIB associating prefix and face
  - Advertise (broadcast) the prefix with IGP LSA

Intra-domain Routing

Multiple announcement of prefix

Difference between IP and CCN

- IP : "all hosts with this prefix can be reached by me" (a priori loopfree : single outgoing if)
- CCN : "some of the content with this prefix can be reached by me"
- All faces for a FIB entry can be contacted (a posteriori loopfree : PIT)

Inter-domain Routing

Connection between ISPs

Bottom-up deployment of CCN

- Incentive for ISP, reduce peering cost via caching
- Problem : connecting two CCN capable ISPs through a gap, how to advertise ISP customer's content
- Solution : BGP domain level content prefixes eq. to IGP tlv at AS lvl

Content Networking over OpenFlow

- Junho Suh paperhttp ://www.asiafi.net/org/ndn/hands-on2012/presentations/4.OF-CCN-jhsuh.pdf
- Clean slate approach directly implemented in network
- Supported by many vendors
- Content caching/retrieval in line-card

CCN for MANET, IoT

CCN in disruptive environment

- Native support of context-awareness
- Opportunistic routing, easy mobility
- Multiple network if usage, Automatic fail over
- Security and privacy by encryption
- Context-aware name space (*/ThisRoom/projector*)

CCN for Emergency MANET (UCLA) [OLG10]

CCN in IoT

- CCNx implementation on Android
- INRIA Nancy MADYNES : CCN implemtation on Contiki OS
- Alternative to 6lowpan

Scalability issues

CCN architecture

- Scalability of routing on names
- Shift of the address space from one billion IPs to at least one trillion content names

CCN routers

CCN needs more powerful routers

- Stateful (track of Interests)
- Able to verify signatures
- Able to cache content

[PV11] : today technology sufficient for ISP/CDN deployment, not Internet

Energy efficiency

Green Networking ?

Energy balance of CCN ?

- Energy saving through optimization of content delivery
- Energy consumption by high cryptography usage
- What benefit under large deployment ?
- [LRH10] shows better energy efficiency than CDN and P2P networks
  - Reduced hop count to content
  - Content servers more energy efficient than routers

Performance Evaluation

Link usage and Overhead

Based on the 2009 CCNx prototype [JST+09]

- Local (large) file transfer :
  - TCP : 90% of link bandwidth
  - CCN : 68% (22% CCN header + IP/UDP encapsulation)
  - 3 times slower to reach stable maximum throughput (may be optimized)
- Web page download :
  - Less overhead than HTTP (HTTP 15%, CCN 7%, while secured)
  - Outperforms HTTPS (HTTPS 25%, CCN 8%)

Performance Evaluation

Data dissemination

- Multi-clients distribution : CCN vs TCP client-server
- Increasing number of client : CCN=constant dl time, TCP=linear increase

Multi-interface support

VOIP between two host, each connected to two networks

- Strategy layer periodically checks faces and uses the fastest responding
- One interface disconnected : automatic failover

Plan

PARC's library

CCNX

- Nice set of tools for scientists and hackers
- Library provides essential functionalities to run a CCN network : routing, communication, key/trust management, signing, encryption

Features

- Multi-OS support (Linux, MacOS, FreeBSD)
- Android implementation for smartphones
- Written in two languages :
  - C implementation (forwarder, repository, persistent storage of CCNx data, library, utilities, skeleton API docs, and unit test suite) : required for all CCNx communication, preferred for system development
  - JAVA implementation (library, utilities, skeleton API docs, and unit test suite) : preferred for applications development

Featured experimental tools

Testing applications

- ccnChat : basic chat communication on local LAN
- ccnFileProxy : basic communication on local LAN

Useful tools

- Wireshark plug-in
- VLC plug-in

Simulator (not part of CCNx)

- ccnSim : scalable chunk-level simulator of CCN (developed by ANR Project Connect)
- Aims to assess CCN performance in large networks (up to 10^8 files and 10^6 chunks)
- Omnet++ plug-in, open-source software for cross-comparison of

Following the practical session

3 choices

- 1 : log into the desktop computer login *aims.2012@uni.lux*, password *A1M$-Uni$*, start VMware Player, Launch "Ubuntu11_CCNready" image, log into the gest OS : login *aims*, password *aims*
- 2 : install VMware Player (3.1.4) on your laptop, ask for the image (3.5GB)
- 3 : Directly install CCNx on your laptop

VMWare Image

Building the image

- Regular Ubuntu 11.04 with standard building tools (gcc, make, etc.)
- Additional libs for CCNx :
  - open-jdk-1.6, ant1.8
  - libssl, libxml2, libexpat (xml parser), libpcap
- Additional libs for Wireshark
  - bison, flex
  - libgtk2.0, libglib2.0

Installation Process

Getting the Source and building

- Go to http ://www.ccnx.org/ , Download section, no need to register
- CCNx software is only available in source code
- Download ccnx-0.6.0.tar.gz
- tar -xvzf ccnx-0.6.0.tar.gz
- export JAVA_HOME=/usr/lib/jvm/java-6-open-jdk/
- ./configure, make

Installation and Test (optional)

- (make install) if admin on computer, otherwise all can run with limited rights
- make test (quite long) ... BUILD SUCCESSFUL

PARC's experimental platform

Documentation

Most important thing :
http ://www.ccnx.org/releases/ccnx-0.6.0/doc/

- Commands and tools MAN pages :
  http ://www.ccnx.org/releases/latest/doc/manpages/index.html
- CCNx Protocol specifications (coding, sequences of messages) :
  http ://www.ccnx.org/releases/latest/doc/technical/CCNxProtocol.html
  http ://www.ccnx.org/releases/latest/doc/technical/index.html
- C, Java, Android code documented

CCNx commands

Core of CCNx

- **ccnd** : routing daemon (ccn forwarder, runs in userspace)
  - ccndstart, ccndstop, config file $HOME/.ccnx/ccnd.conf, default port 9695
  - Interest and Data encapsulated in UDP
- **ccndc** (ccndcontrol) : set routes in the FIB (similar to route)(content announcement) (.ccnx/ccnd.conf)
- **ccnr** : CCNx Repository, stores content and responds to Interests requesting stored content, path in CCNR_DIRECTORY variable

CCNx commands

Content management

- ccngetfile : retrieve a file published as CCNx content and save it to a local file
- ccnputfile : publish a file as CCNx content (local file filename or url to content with the ccnxname), manage segmentation, key signature, etc.
- ccnrm : mark as stale any locally cached content objects matching the given prefix (no further Interest response)
- ccnls : list name components available at the next level for a given CCNx name prefix
- ccnlsrepo : explore content stored under a given prefix (one or more repositories)

CCNx commands

Tools

- ccn_ccnbtoxml : convert ccn binary encoded data into XML form
- ccn_xmltoccnb : convert XML into ccn binary encoded data (ccnb)
- ccndsmoketest : testing of communications, send and receive data on sockets

Monitoring of ccnd

Monitoring commands

- ccndstatus : display the status a running ccnd (nb of active faces, stat of each face, etc)
- ccnpeek : generates an Interest, get one content item matching the name prefix and write it to stdoud (eq to IP ping)
- ccnponk : read data from stdin, send it as a single ContentObject in response to an interest

Web Interface

- http ://localhost :9695/ , similar to ccndstatus
- Limitation : sliding time window (avg of last minute) provides inaccurate results

First communications

Wireshark

- Already compiled with the CCNx plug-in
- To be launched from the local dir : sudo ./home/aims/wireshark/wireshark
- Start capture on loopback if, filer : ccn

CCN chat

- Launch ccndstart (if not yet)
- ccndc add ccnx :/ccnx.org udp machine_name (announce name ccnx.org (or whatever) routable through local machine)
- ccnchat ccnx :/ccnx.org/test_chatroom

First communications

mailly ccnd[14550] local port 9695 api 6000 start 1338724361.760754 now 1338726149.399268

Content items: 23 accessioned, 23 stored, 11 stale, 0 sparse, 112 duplicate, 143 sent
Interests: 31 names, 2 pending, 2 propagating, 14 noted
Interest totals: 586 accepted, 447 dropped, 588 sent, 112 stuffed

Faces

- **face:** 0 **flags:** 0xc **pending:** 0
- **face:** 1 **flags:** 0x400c **pending:** 0
- **face:** 2 **flags:** 0x5012 **pending:** 0 **local:** 0.0.0.0:9695
- **face:** 3 **flags:** 0x5010 **pending:** 0 **local:** 0.0.0.0:9695
- **face:** 4 **flags:** 0x4042 **pending:** 0 **local:** [::]:9695
- **face:** 5 **flags:** 0x4040 **pending:** 0 **local:** [::]:9695
- **face:** 7 **flags:** 0x81412 **pending:** 0 **remote:** 127.0.1.1:9695 **via:** 2
- **face:** 12 **flags:** 0x1014 **pending:** 2 **activity:** 13 **remote:** 127.0.0.1:38200
- **face:** 13 **flags:** 0x1014 **pending:** 0 **activity:** 7 **remote:** 127.0.0.1:38202
- **face:** 14 **flags:** 0x21012 **pending:** 0 **activity:** 7 **remote:** 127.0.0.1:9695 **via:** 2

Face Activity Rates

	Bytes/sec In/Out	recv data/intr sent	sent data/intr recv
face: 0	259 / 31	0 / 0	0 / 0
face: 7	0 / 176	0 / 0	0 / 0
face: 12	128 / 0	0 / 0	0 / 0
face: 13	0 / 0	0 / 0	0 / 0
face: 14	440 / 263	0 / 0	0 / 0

Forwarding

- ccnx:/%C1.M.S.localhost/%C1.M.SRV/ccnd **face:** 0 **flags:** 0x3 **expires:** 2147481862
- ccnx:/ccnx/ping **face:** 0 **flags:** 0x3 **expires:** 2147481862
- ccnx:/ccntuto2 **face:** 7 **flags:** 0x3 **expires:** 2147482037
- ccnx:/%C1.M.S.neighborhood **face:** 0 **flags:** 0x3 **expires:** 2147481862
- ccnx:/%C1.M.S.localhost **face:** 0 **flags:** 0x23 **expires:** 2147481862
- ccnx:/ccnx/%1B%D20%5C%AD%86%99Z%1F%BE%94%09%06%FAy%12%F6%19%E4%8E%B6%F6oI%8B%17%A4%E5%A3.%05%DB **face:** 0 **flags:** 0x17 **expires:** 2147481862
- ccnx:/ccntuto **face:** 7 **flags:** 0x3 **expires:** 2147481922
- ccnx:/ccntuto2/test_chat_room/Users/tibs/Keys/%C1.M.K%00%9D%BA%9Cv%AC%DC%BE%DA%CE%80%21HAYG%1A%D1izN%3A_%2F8s%7F%FC%D1%E9%13cR/%FD%04%EAXbp%00 **face:** 12 **flags:** 0x3 **expires:** 2147483637

FIGURE: Web interface for ccnd monitoring

First communications, wireshark analysis

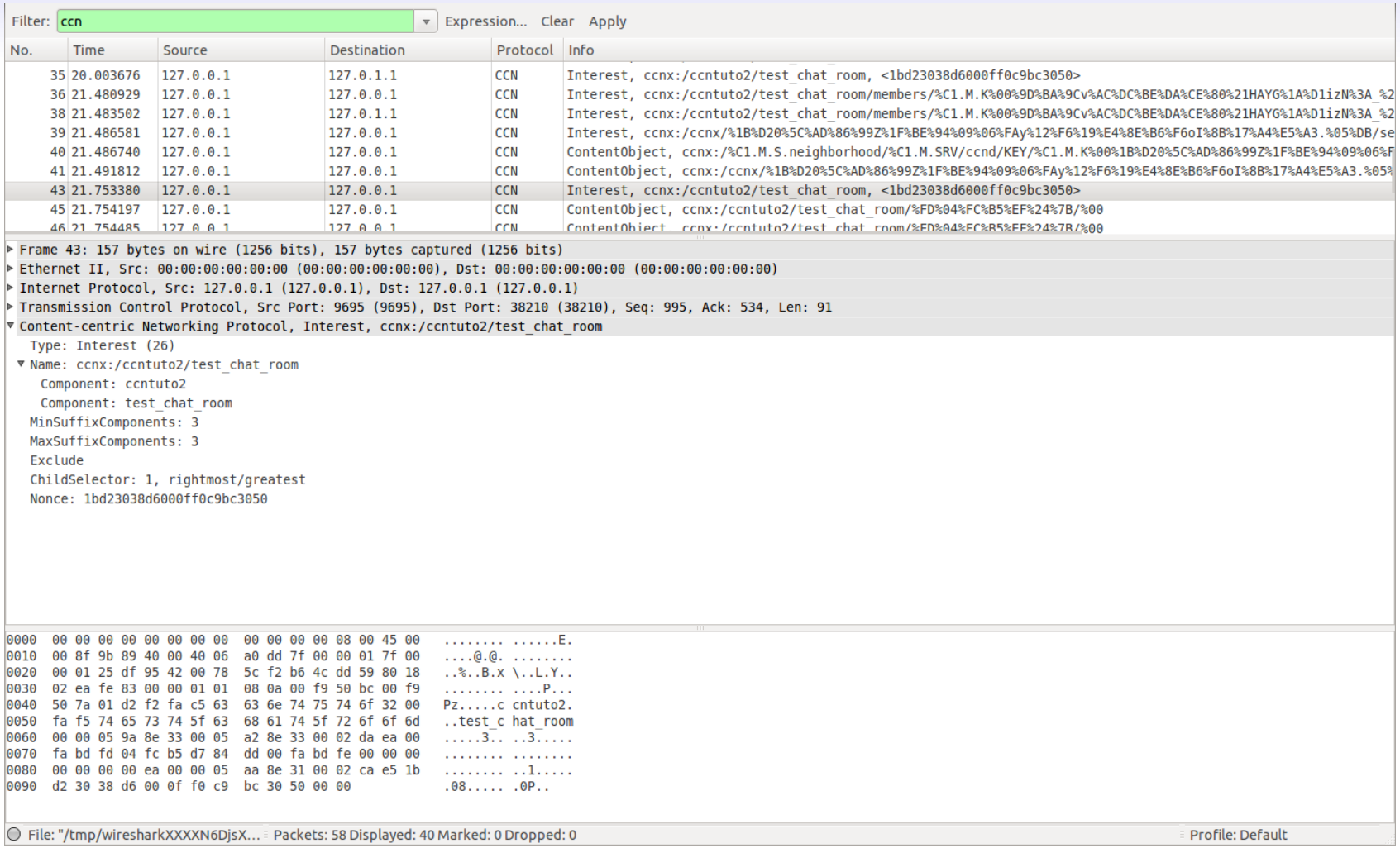

FIGURE: Interest captured by Wireshark

First communications, wireshark analysis

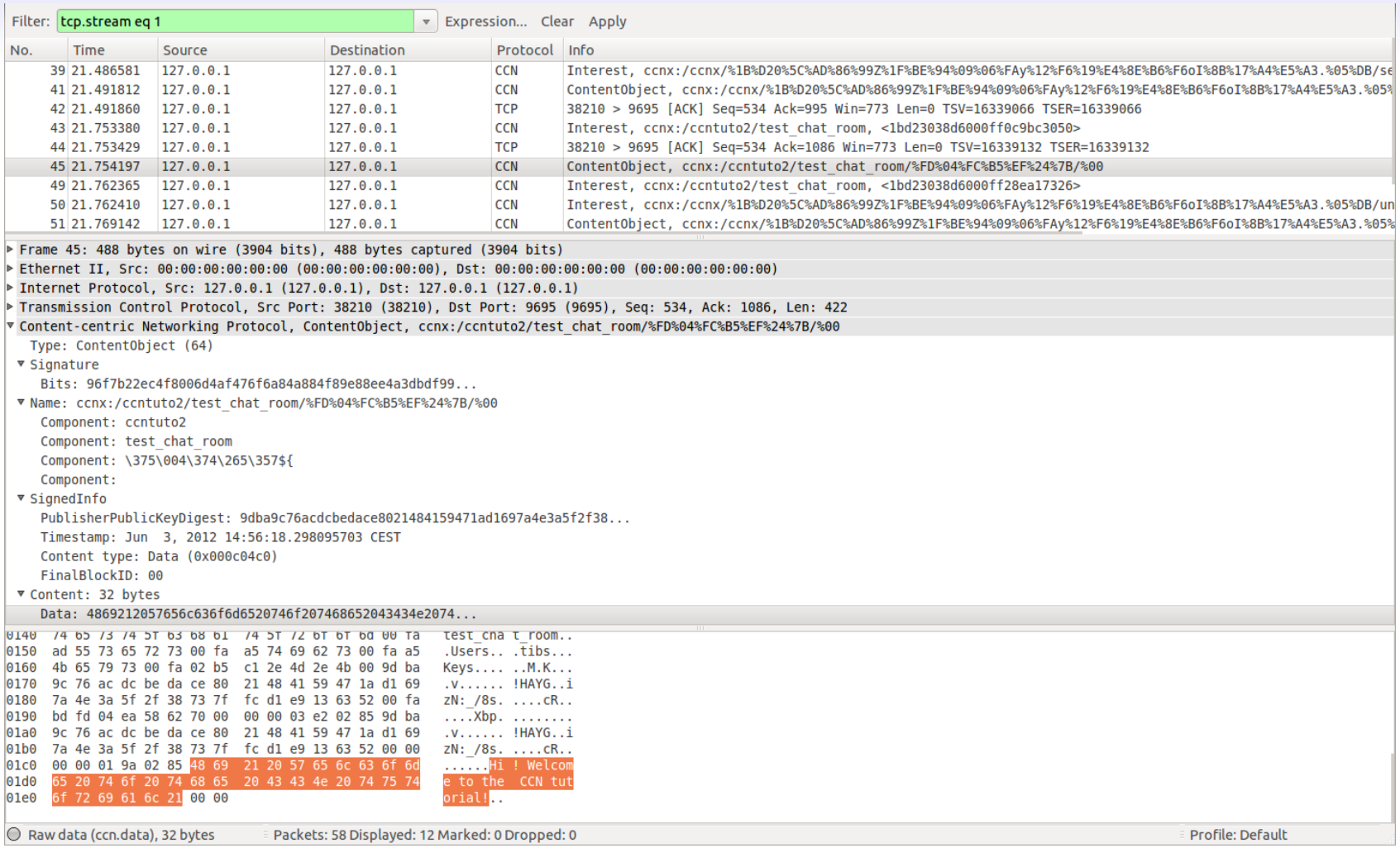

FIGURE: Data captured by Wireshark

Creation of a CCN repository

CCN repository

- Needed to announce content that can be transferred
- Repository path defined in env variable :

  `CCNR_DIRECTORY`

- `mkdir ~/CCN_shared ; export CCNR_DIRECTORY=~/CCN_shared`
- Launch repository daemon : ./bin/ccnr

Adding files

- Once the repository is created : add the content
- `ccnputfile -v ccnx:/CCN_shared/Ubuntu_sample_Howfast.ogg ~/Example/Howfast.ogg`
- List the content published under the name : ccnlsrepo ccnx :/CCN_shared/

Data transfer

Getting files

- We have a populated content repository under ccnx :/CCN_shared/
- `ccngetfile -v ccnx:/CCN_shared/Ubuntu_sample_Howfast.ogg ~/retrieved_file.ogg`
- In Wireshark : observe parity between Interest / Data during transfer

First Hacking : Interest flooding

Forging Interests

- Lets write a script : flooding.sh, first line : `#!/bin/sh`
- We create random Interests with ccn_xmltoccnb
- ```
  I=0
  while [ $I -lt $1 ]; do
  ccn_xmltoccnb -w - <<EOF >unanswered_$I.ccnb
  <Interest>
    <Name>
      <Component ccnbencoding="text">CCN_shared</Component>
      <Component ccnbencoding="text">test$I</Component>
    </Name>
  </Interest>
  EOF
  I=$((I+1))
  done
  ```
  ```

First Hacking : Interest flooding

Sending Interests

- We send the forged Interests with ccndsmoketest
- ```
  I=0
  while [ $I -lt $1 ]; do
  ccndsmoketest -u localhost send unanswered_$I.ccnb
  I=$((I+1))
  done
  return 0
  ```
- Lets try with 100 Interests : ./flooding.sh 100 (remember chmod 755)
- ccndstatus or http ://localhost :9695/ to see the result
  ```

Using the C library

Goal : writing a simplified ccnpeek

- Read the URI, emit Interest, print response on stdout

Include

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <ccn/ccn.h>
#include <ccn/charbuf.h>
#include <ccn/uri.h>
```

Using the C library

Easy start

```
int main(int argc, char **argv)
{

if (argv[1] == NULL){
printf("Error: no URI specified\n");
return 1;
}
else{
//code to do
}
}
```

Using the C library

To Do 1 : declaration

```
printf("%s\n",argv[1]);

int res;
size_t length;
const unsigned char *ptr;
int timeout_ms = 3000;
struct ccn *h = NULL;
struct ccn_charbuf *name = NULL;
struct ccn_charbuf *resultbuf = NULL;
struct ccn_parsed_ContentObject pcobuf = { 0 };
```

Using the C library

To Do 2 : connecting to ccnd

```
//put our uri in a proper buffer
name = ccn_charbuf_create();
res = ccn_name_from_uri(name, argv[1]);

h = ccn_create(); //create a client handle
res = ccn_connect(h, NULL);//connect to local ccnd
if (res < 0) {
ccn_perror(h, "ccn_connect");
exit(1);
}
```

Using the C library

To Do 3 : send Interest and print

```
resultbuf = ccn_charbuf_create();
res = ccn_get(h, name, NULL, timeout_ms, resultbuf, &pcobuf,
 NULL, 0); //get a single matching ContentObject
if (res >= 0) {
ptr = resultbuf->buf;
length = resultbuf->length;
//if (content_only)
// ccn_content_get_value(ptr, length, &pcobuf, &ptr,
// &length);
if (length > 0){
printf("We got something\n");
res = fwrite(ptr, length, 1, stdout) - 1;
}
}
```

Using the C library

To Do 4 : cleaning and exit

```
//Cleaning
ccn_charbuf_destroy(&resultbuf);
ccn_charbuf_destroy(&name);
ccn_destroy(&h);
exit(res < 0);
```

Compilation and execution

```
gcc -o simplepeek simplepeek.c -l ccn -l crypto
./simplepeek ccnx:/CCN_shared/your_content
```

Using the JAVA library

Compilation and execution

```
import org.ccnx.ccn.*;
import java.io.IOException;
public class HelloCCN{
public static void main(String[] args) {
System.out.println("Hello CCN world!");
}
}
```

Compilation and execution

```
javac HelloCCN.java -classpath /home/aims/ccnx-0.6.0/
lib/ccn.jar
java HelloCCN
```

Plan

What to remember on CCN ?

What we learnt on CCN

- Named content rather than named hosts
- Better security, delivery efficiency, mobility and disruption tolerance than TCP/IP
- Incrementally deployable
- CCNx : Great community and tools for experimentation

Community Meeting

CCNx Community Meeting at INRIA Sophia Antipolis in September 12 and 13 !

Key challenges

General

- Scalability of CCN (routing on names)
- Management of network (instrumentation) and content (accountability, revocation)
- Security of CCN nodes
- New applications (Andana : Anonymous Named Data Networking Application [DGTU11])

Key management

- Trust and security are part of the content thanks to crypto
- To tackle unreliable intermediate nodes :
  - content needs to be signed with a publishers certificate key
  - routers need to verify that the content has been produced by its owner

Final words

From Pablo Rodriguez blog

"CCN democratizes Content Distribution and ensures that anyone - not just those in position to pay a CDN - can enjoy the benefits of an Internet-broadcast service that amplifies your data whenever and wherever it is needed."

Political question and positioning

Side effect, goal of CCN

- CCN can turn the Internet into a public service for data dissemination
- ISP friendly (opposite to P2P)

Final words

Political question and positioning

Content diffusion companies will not push CCN !

- They live thanks to the valuable data we must put on their servers
- They have NO interest to give back power to the network

ISP will need strong incentives : (e.g. cost of bandwidth vs cost of deploying storage nodes)

- Already have caching solutions deployed
- Companies do not like changes and investments

Final words

Academic position

- Is the Internet inertia too high for a natural shift ?
- Should academic research actively support CCN (i.e. build the first WAN CCN network) ?
- What is the main evolutionary force for CCN (like RIAA for P2P) ?
  - Ethical ?
  - Technical ?

What kind of Internet do we want ?

- Limited to :
  - 1 : Conversations with servers holding even private content ?
  - 2 : Inefficient and unsafe P2P communications ?
- Free the diffusion of content without sacrificing scalability and security

External resources

Helpful material used to prepare this talk

- Van Jacobson talk at Google :
  http ://video.google.com/videoplay ?docid=-6972678839686672840
- Van Jacobson slides FISS 2009 http ://www.comnets.uni-bremen.de/typo3site/uploads/media/vjCCN-FISS09.pdf
- Pablo Rodrigez thoughts on CCN :
  http ://blog.rodriguezrodriguez.com/

Contact

Questions ? Comments and feedback ? Please send them to :
thibault.cholez@uni.lu

Steve DiBenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun.
Andana : Anonymous named data networking application.
CoRR, abs/1112.2205, 2011.

vinicius Gehlen Giovanna Carofiglio and Diego Perino.
Experimental evaluation of storage management in content-centric networking.
In *IEEE ICC*, 2011.

Ali Ghodsi, Scott Shenker, Teemu Koponen, Ankit Singla, Barath Raghavan, and James Wilcox.
Information-centric networking : seeing the forest for the trees.
In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, HotNets '11, pages 1 :1–1 :6, New York, NY, USA, 2011. ACM.

Van Jacobson, Diana K. Smetters, Nicholas H. Briggs, Michael F. Plass, Paul Stewart, James D. Thornton, and Rebecca L. Braynard.

VoCCN : voice-over content-centric networks.
In *Proceedings of the 2009 workshop on Re-architecting the internet*, ReArch '09, pages 1–6, New York, NY, USA, 2009. ACM.

Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard.
Networking named content.
In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM.

Tobias Lauinger.
Security & Scalability of Content-Centric Networking.
Master's thesis, TU Darmstadt, September 2010.

Uichin Lee, Ivica Rimac, and Volker Hilt.
Greening the internet with content-centric networking.

In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, pages 179–182, New York, NY, USA, 2010. ACM.

S. Y. Oh, D. Lau, and M. Gerla.
Content centric networking in tactical and emergency MANETs.
In *Wireless Days (WD), 2010 IFIP*, pages 1–5. IEEE, October 2010.

Diego Perino and Matteo Varvello.
A reality check for content centric networking.
In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ICN '11, pages 44–49, New York, NY, USA, 2011. ACM.

D. Smetters and V. Jacobson.
Securing Network Content.
Technical report, PARC, October 2009.

Hendrik Schulze and Klaus Mochalski.
Internet study 2008/2009, 2009.

Lixia Zhang, Deborah Estrin, Jeffrey Burke, Van Jacobson, James D. Thornton, Diana K. Smetters, Beichuan Zhang, Gene Tsudik, kc claffy, Dmitri Krioukov, Dan Massey, Christos Papadopoulos, Tarek Abdelzaher, Lan Wang, Patrick Crowley, and Edmund Yeh.

Named Data Networking (NDN) Project, October 2010.