

# A semantic firewall for Content Centric Networking

IFIP/IEEE Integrated Network Management Symposium (IM 2013) - MC2:  
Security Management and Recovery  
May 27 - 31, 2013

*David Goergen*

Thibault Cholez

Jérôme François

Thomas Engel

**SnT** – Interdisciplinary Centre for Security, Reliability and Trust

# OUTLINE

- Introduction
- Content Centric Networking background
- Design
- Implementation
- Evaluation
- Conclusion

A semantic firewall for Content Centric Networking

# INTRODUCTION

# Introduction

- Trend towards content retrieval
- Content Centric Networking is built and designed to follow this
  - Some security measures already built-in
    - Authentication of content
  - But real security tools missing
- Our contribution:
  - Identify the security needs for a CCN architecture
  - Design of a semantic CCN firewall
  - Performance evaluation

# Related work

- Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: **Networking named content**. In: Proceedings of the 5th international conference on Emerging networking experiments and technologies. pp. 1–12. CoNEXT '09, ACM, New York, NY, USA (2009)
- D. Smetters, V. Jacobson: **Securing Network Content** (October 2009)
- Lauinger, T.: **Security & scalability of content-centric networking** (September 2010)
- Goergen, David; Cholez, Thibault; François, Jérôme; Engel, Thomas: **Security monitoring for Content Centric Networking**, Data Privacy Management and Autonomous Spontaneous Security, Volume 7731 (2013)
- partly funded by BUTLER and IoT6 FP7 EU projects under the grant agreements 287901 and 88445

A semantic firewall for Content Centric Networking

# **CONTENT CENTRIC NETWORKING BACKGROUND**

# Content Centric Networking - CCN

- New paradigm proposed by Van Jacobson et al.
- Redesign networking focusing on data instead of hosts (who provide the data)
- Shift from a communication oriented paradigm to a distribution oriented
- To provide the same functionalities as TCP/IP with build in security features, more efficient content diffusion, mobility, ...

# How does it work?

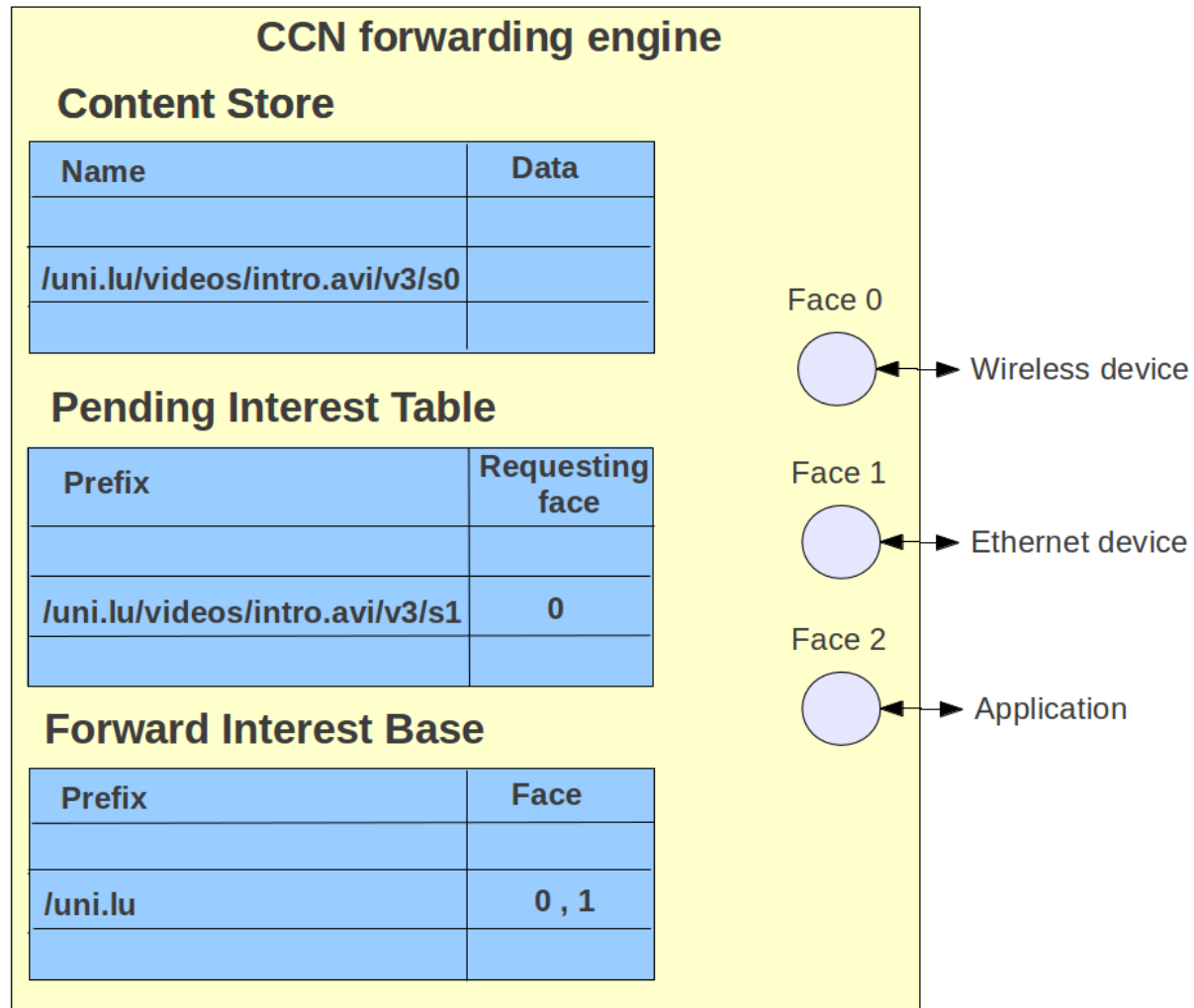
- Routable data instead of routable host
- Content is named in a hierarchical prefix based way
  - Examples:
    - uni.lu/people/goergen/presentation/im2013
    - thisRoom/projector
- Like IP, CCN is semantic free. Meaning is defined by application, global conventions, etc.
- Content is requested by user's Interest
- Anyone who has the solicited content can answer



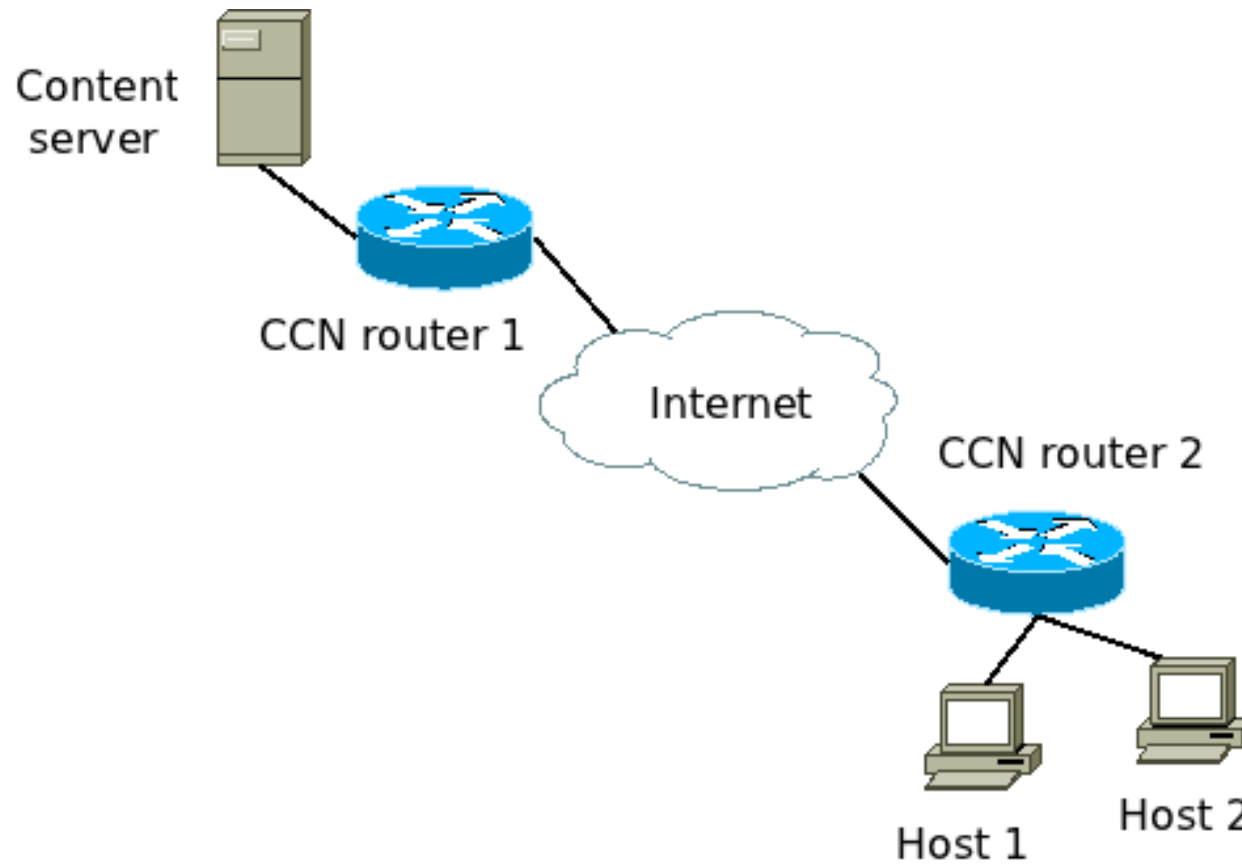
# CCN architecture

- CCN Packets:
  - **Interest Packets** that express Interest for a certain content
  - **Data Packets**, signed by the contents producer, reply to a certain Interest and consume it
- CCN tables:
  - Content store
    - local repository filled with shared content
  - Pending Interest Table (PIT)
    - Contains pending Interest requests send upstream to a content provider
  - Forward Information Base Table (FIB)
    - Contains the faces which correspond to a certain Interest

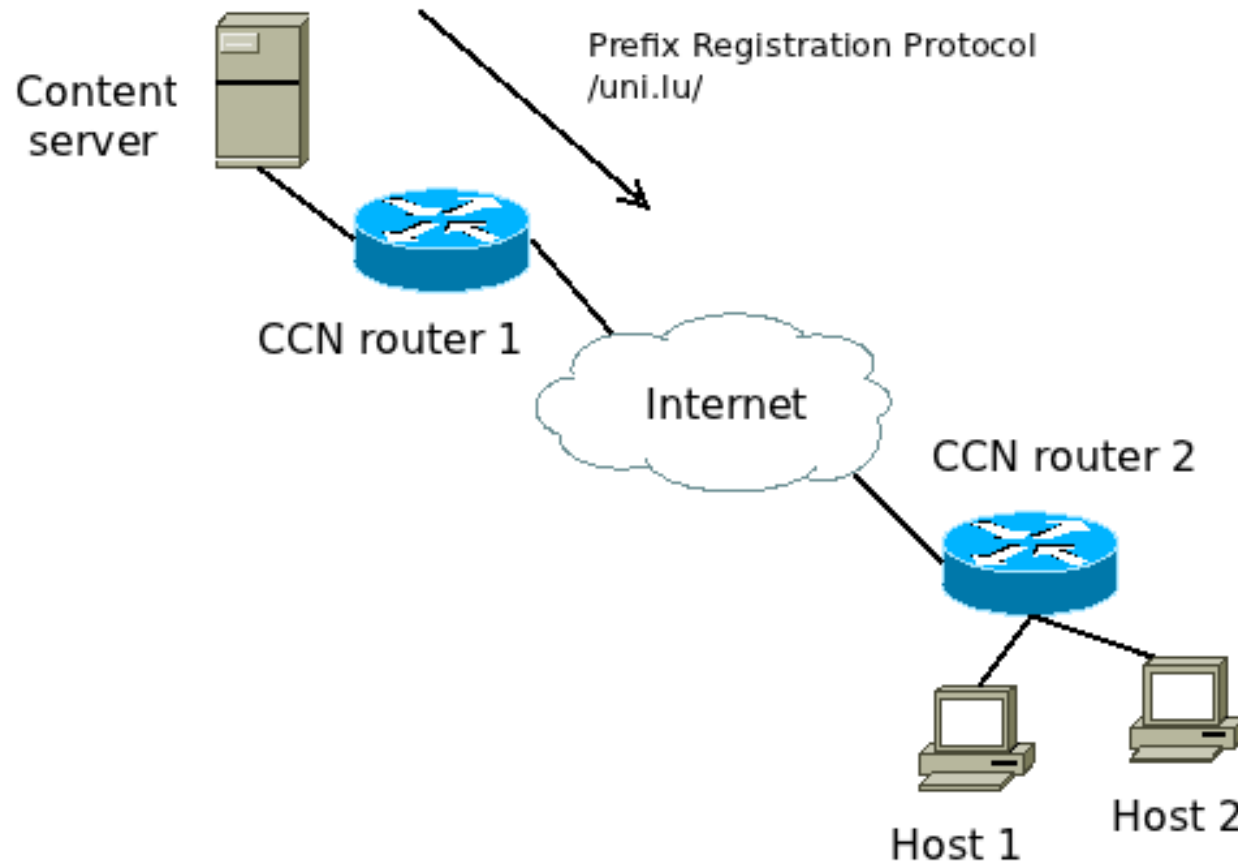
# CCN node model



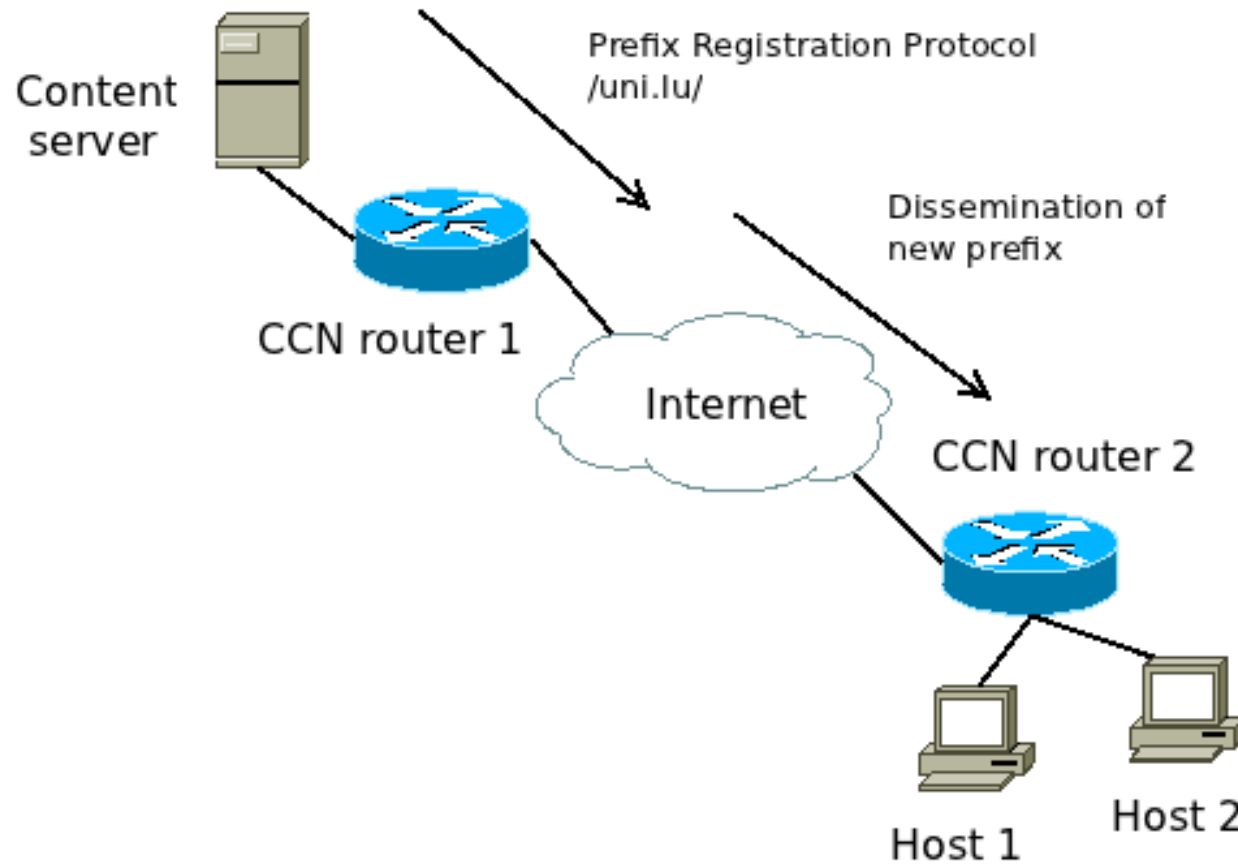
# Routing example



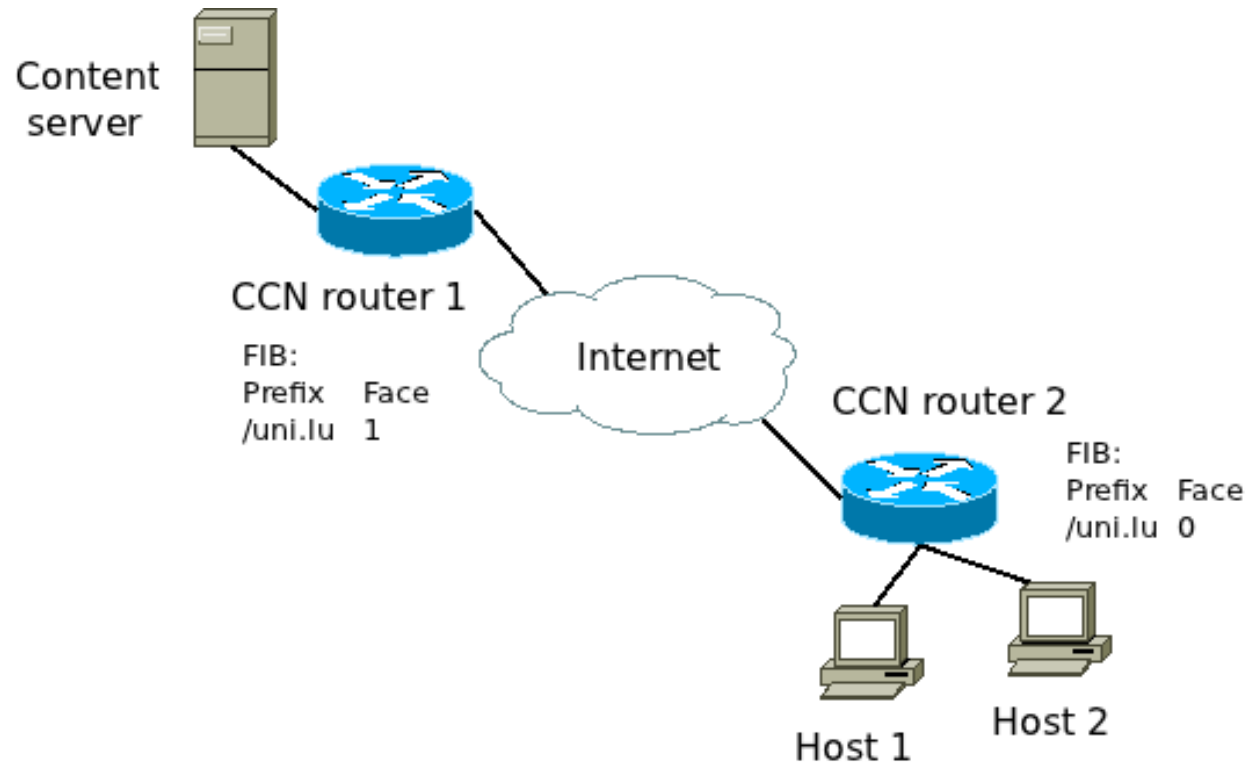
# Routing example cont'd



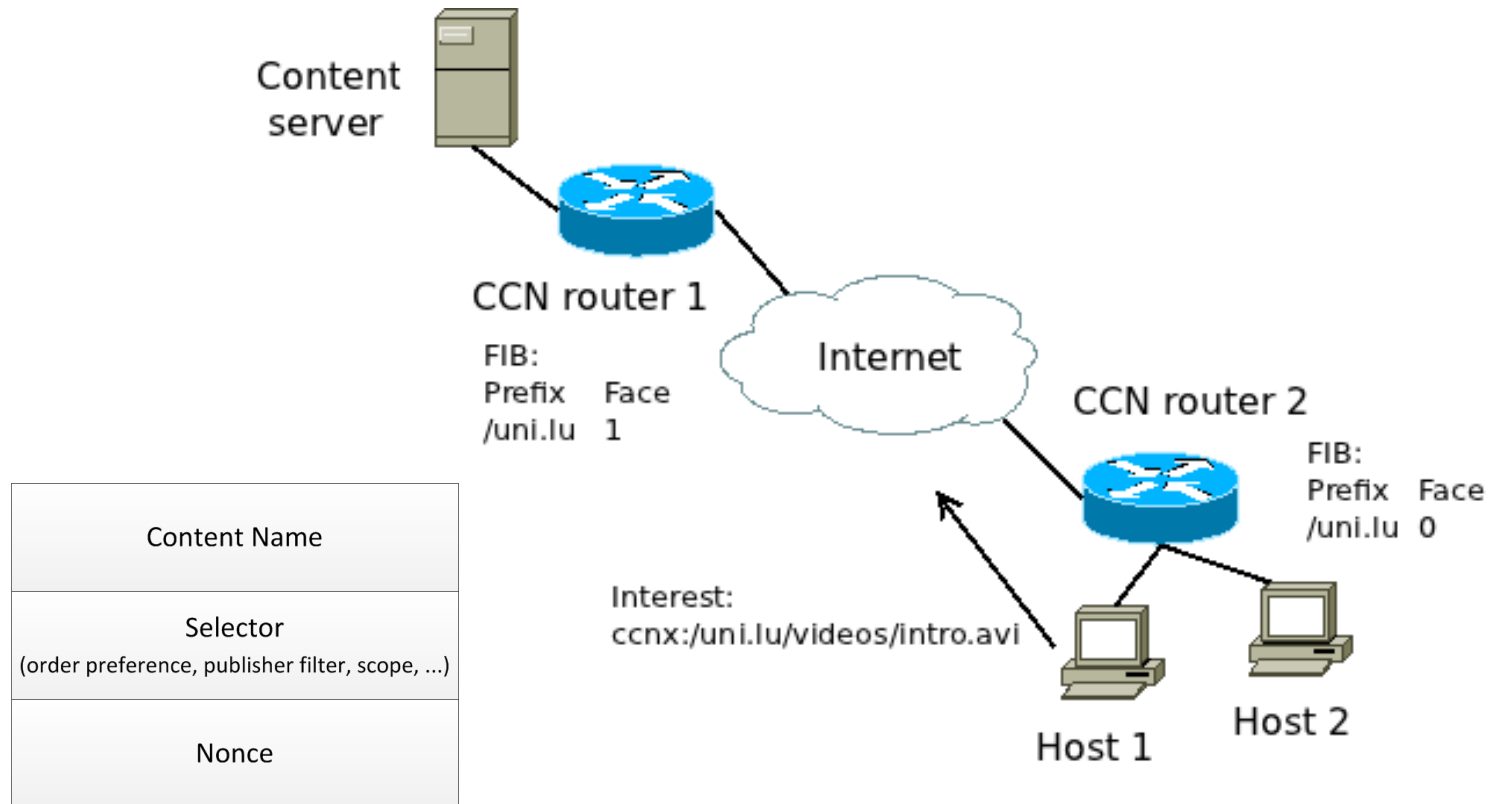
# Routing example cont'd



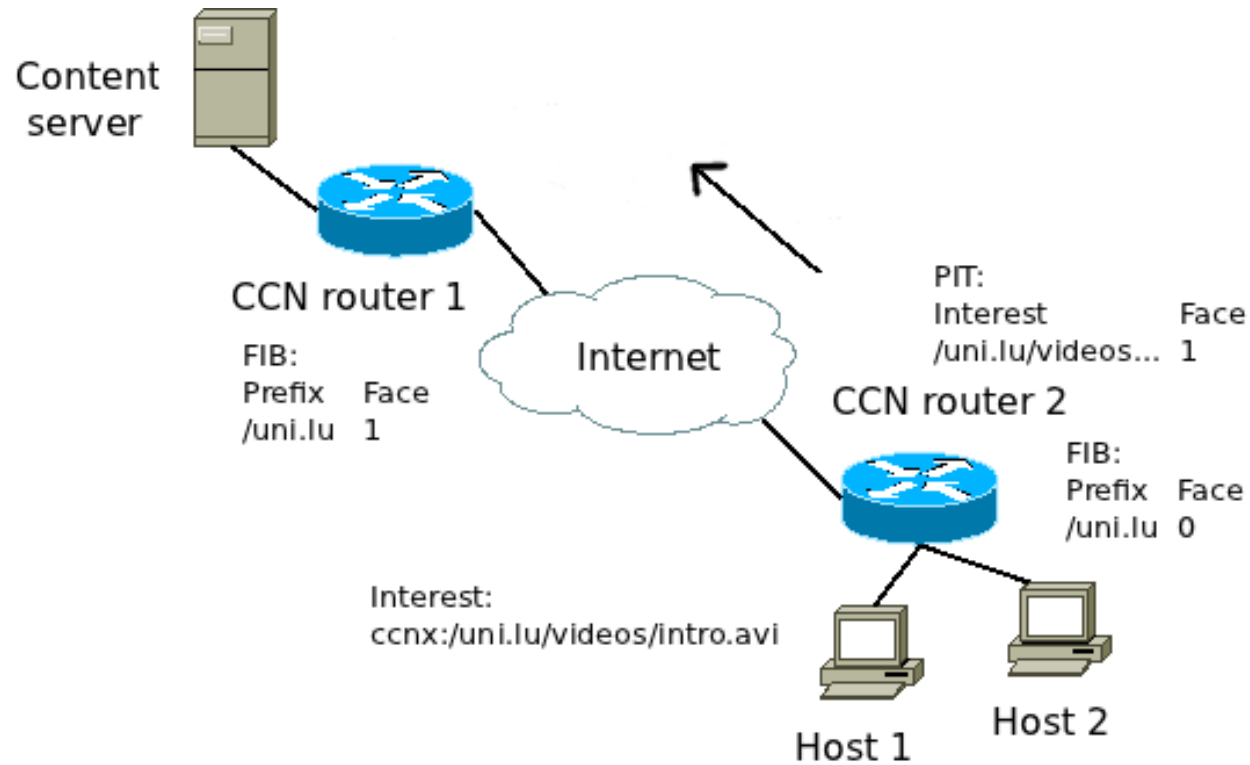
# Routing example cont'd



# Routing example cont'd

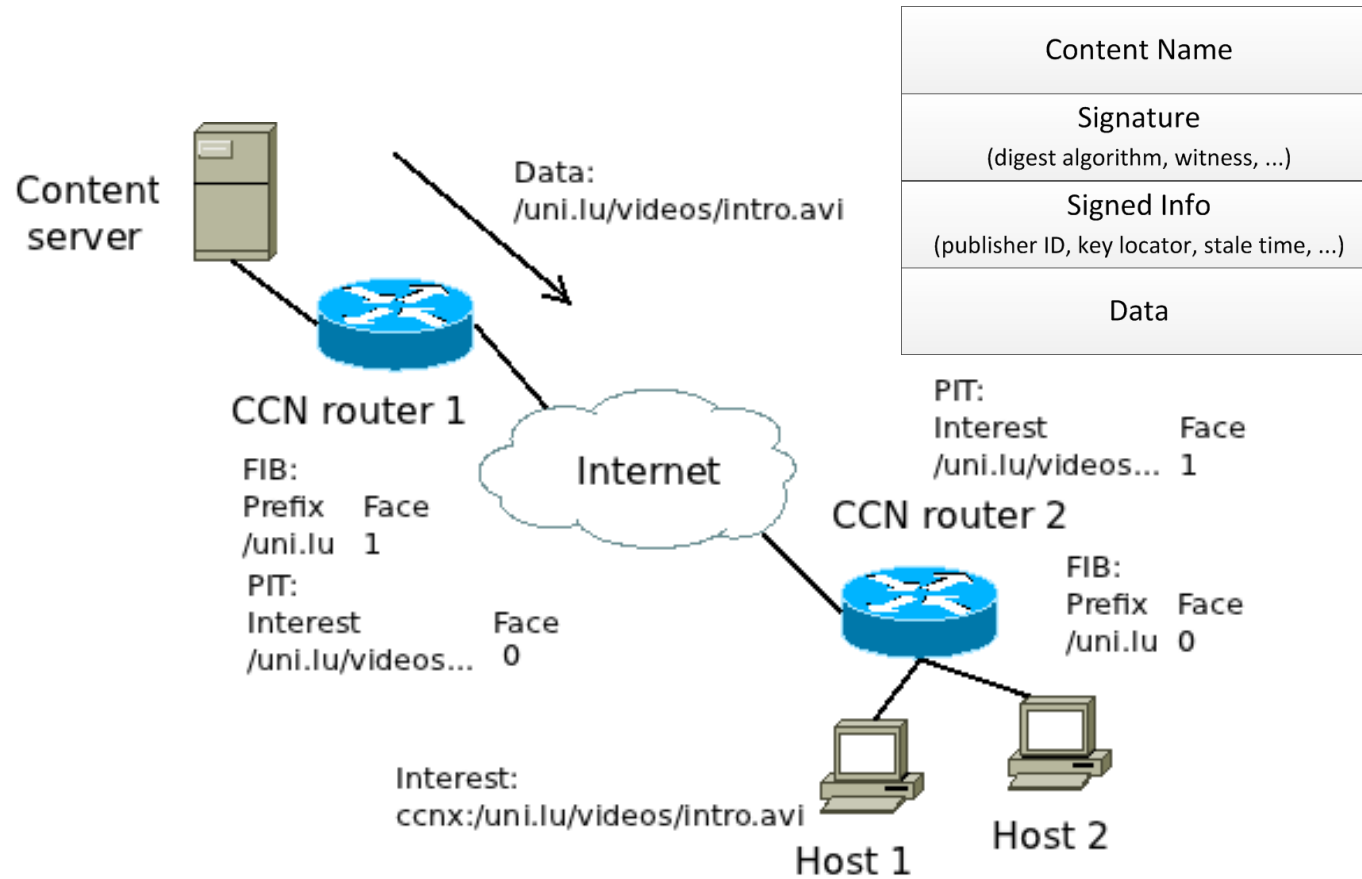


# Routing example cont'd

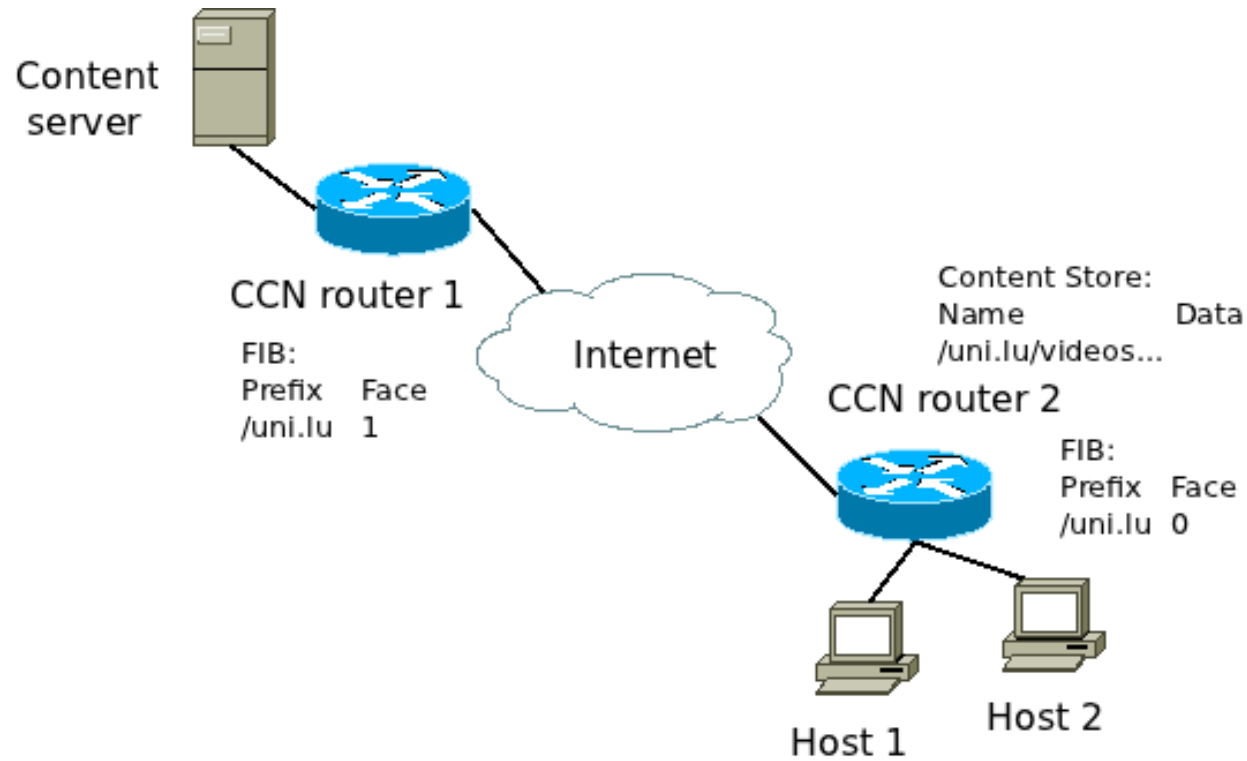




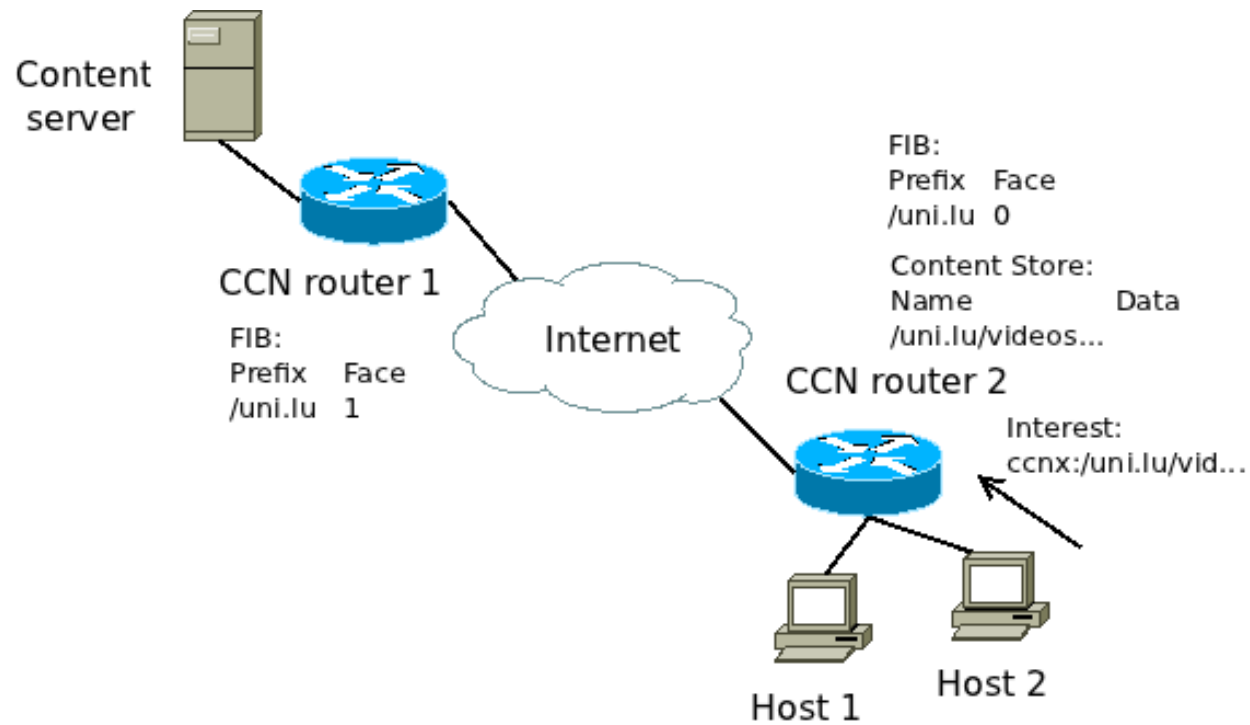
# Routing example cont'd



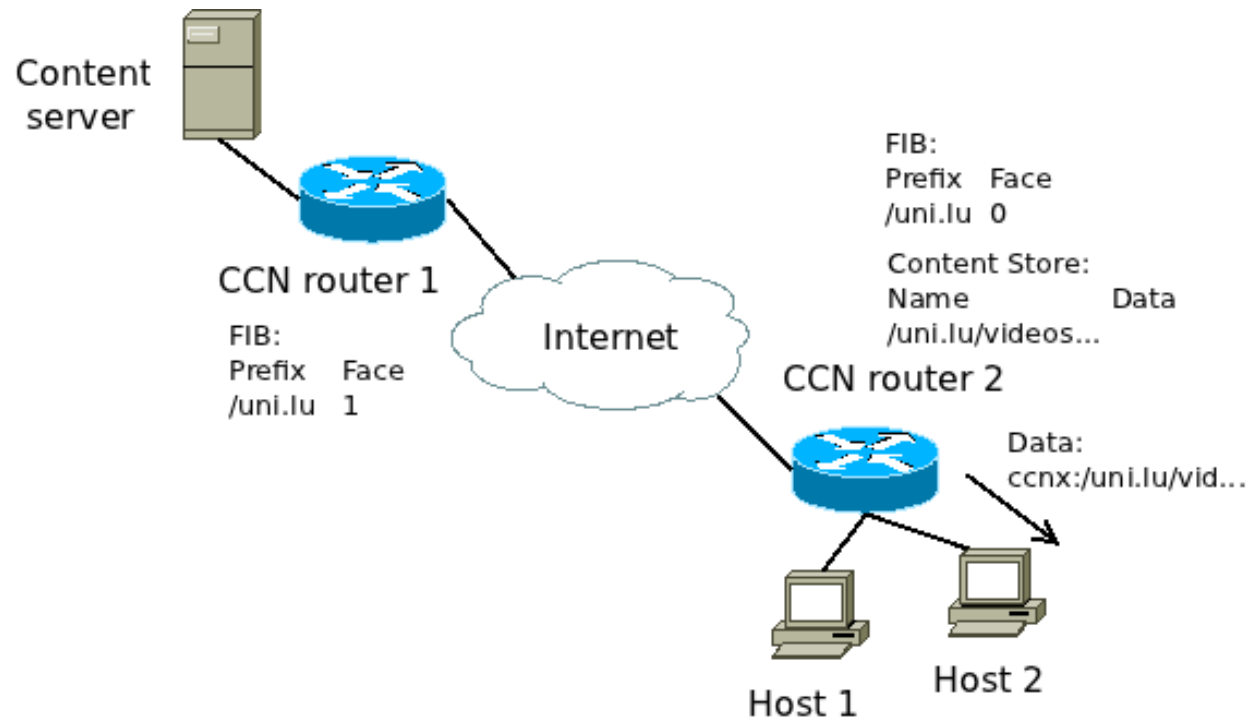
# Routing example cont'd



# Routing example cont'd



# Routing example cont'd



# Security layer

- No Content transmission before Interest reception
  - Renders classic Denial-of-Service, like flooding, inefficient
- Strongly relies on cryptography
  - Authentication of Content and its producer
  - Exclusion of untrustworthy sources
- But new kind of attacks
  - Stateful routers → More vulnerable ?
  - Missing tool for enforcing security policies

A semantic firewall for Content Centric Networking

# DESIGN



# IP firewall general use cases

- IP\_UC1
  - Based on the protocol
    - Example: http, mail, p2p, voip, ...
- IP\_UC2
  - According to the status of the connection
- IP\_UC3
  - Using known blacklisted IP addresses
- IP\_UC4
  - Unusual inbound traffic
    - From a denial of service attack

# CCN-specific use cases

- **CCN\_UC1**
  - Filtering on content provider
    - Example: known untrustworthy or banned
- **CCN\_UC2**
  - Filtering on bad signature
- **CCN\_UC3**
  - Filtering on content name and semantic
    - Example: excluding files with certain extensions
- **CCN\_UC4**
  - Composition (content provider & content name)



# CCN-specific use cases

- CCN\_UC5
  - Filtering on content direction
    - Example: avoid leakage of certain documents
- CCN\_UC6
  - Filtering on heavy traffic
    - Perservation of QoS
- CCN\_UC7
  - Filtering of stored data
    - Example: Only storing specific content

# Comparison

IP use cases	CCN use cases	Filtering on
IP_UC1	CCN_UC3	Protocol / Content name
IP_UC2	--	Status of the connection
IP_UC3	CCN_UC1	Listed IP / Content provider
IP_UC4	CCN_UC6	Unusual / Heavy traffic
--	CCN_UC2	Bad signature
--	CCN_UC4	Composition of filters
--	CCN_UC5	Content direction
--	CCN_UC7	Stored data

A semantic firewall for Content Centric Networking

# IMPLEMENTATION

# Syntax definition

- Syntax based on iptables
  - Ease of use and readability
- Distinguish between 3 types of rules
  - **r\_interest**
    - interest SP direction SP  
match\_interest SP "pit" SP action
  - **r\_face**
    - face SP number
  - **r\_data**
    - data SP direction SP match\_data SP  
["cs" | "pit"] SP action

# r\_interest & r\_face

```
interest SP direction SP match_interest SP "pit"  
SP action
```

- **direction**
  - int | ext | \*
- **match\_interest**
  - \* or regular expression
- **action**
  - forward | drop

- **example :**

```
interest * \@game|play|fun\@ 15 pit drop
```

```
face SP number
```

Number of active faces

- **example :**

```
face 200
```

# r\_data

```
data SP direction SP match_data SP ["cs" | "pit"] SP action
```

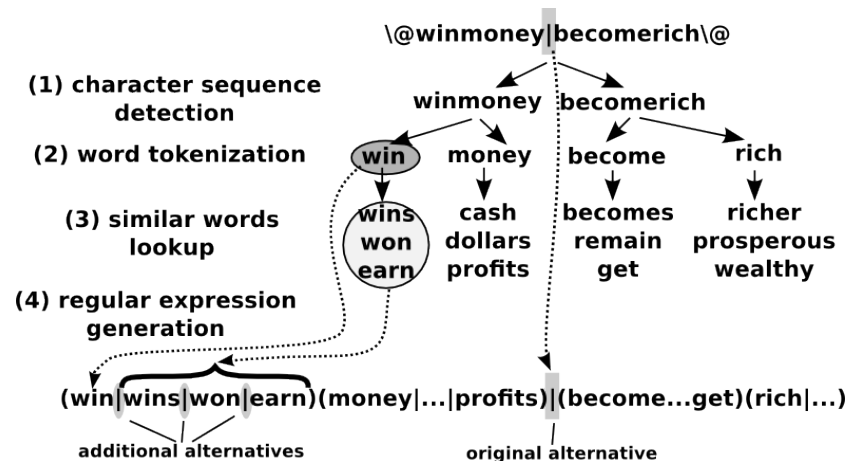
- **direction**
  - int | ext | \*
- **match\_data**
  - content\_name SP provider
- **content\_name**
  - \* or regular expression
- **provider**
  - sign\_check SP provider\_sign
- **signcheck**
  - 0 | 1
- **provider\_sign**
  - \* or hex representation of one or more signatures
- **action**
  - forward | drop

- **example :**

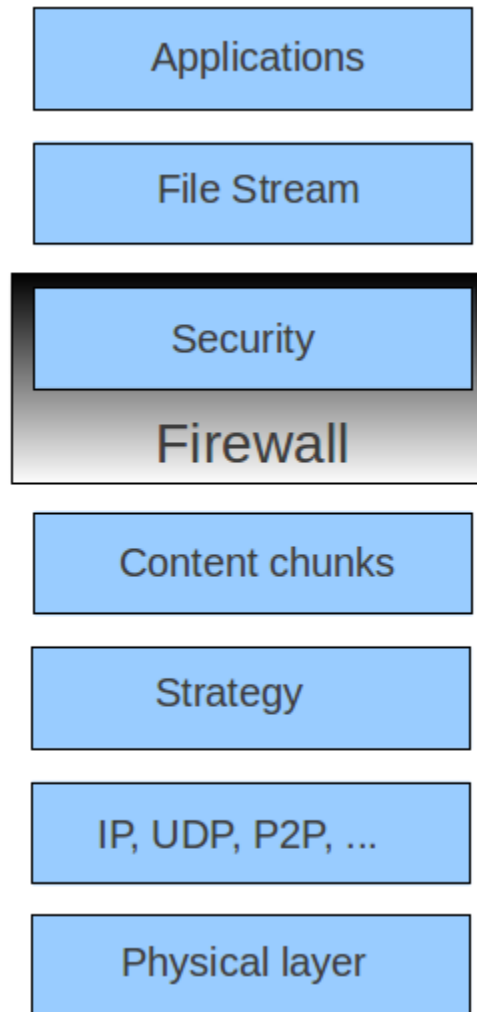
```
data * \@game|fun\@ 0 0 123456789ABCDEF;FFFF0000AAAA pit drop
```

# Pre-processing with Disco

- $\geq 3$  character sequences are extracted
- Segmented as real human-readable words
- For each sequence find  $x$  similar alternative sequences
- Recombine with original to create new regular expression



# Implementation into CCN stack



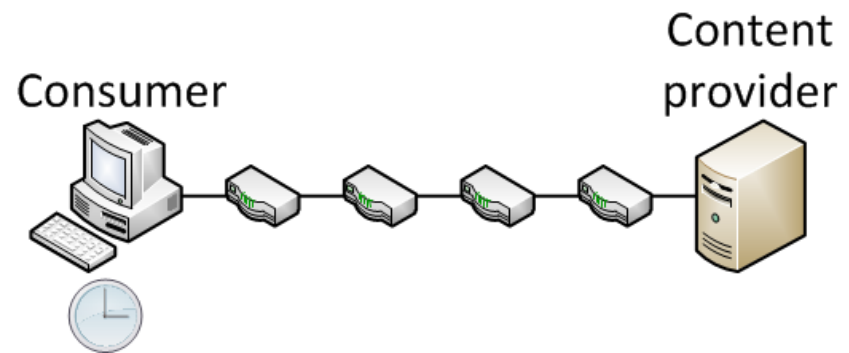


A semantic firewall for Content Centric Networking

# EVALUATION

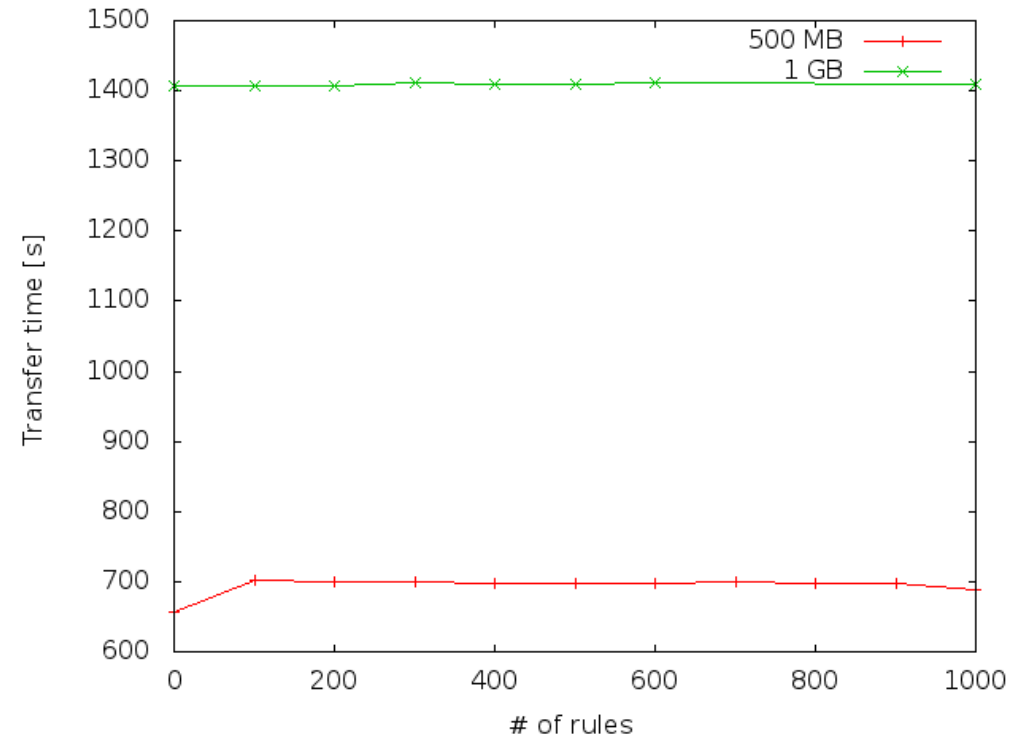
# Setup

- 6 nodes
- Intermediate routers don't cache
- Consumer request single binary file 500MB or 1GB
- Measured transfer time request → received



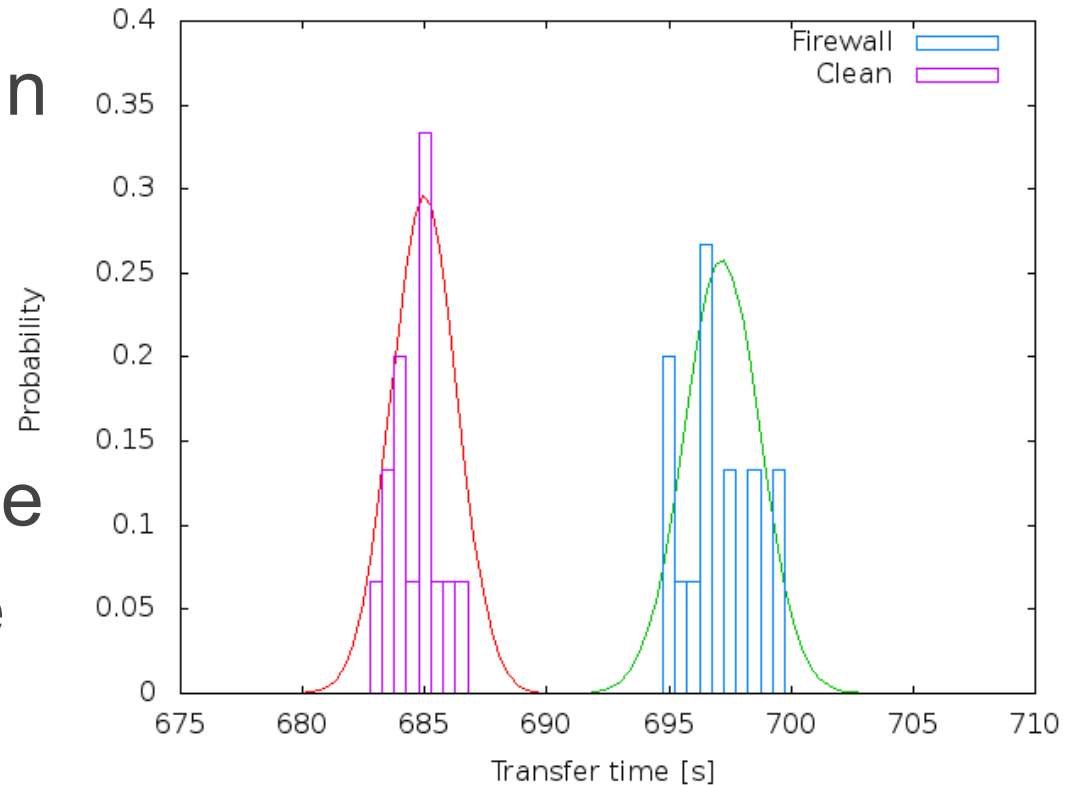
# 1<sup>st</sup> evaluation: Impact of rules

- Impact on the number of processed rules
  - Increasing step 100
  - Request 500 MB and 1 GB file
- Shows small to no impact on transfer time



## 2<sup>nd</sup> evaluation: Clean vs. Firewall

- Repeated experiment to obtain significant results
- Firewallled CCN
  - 1000 rules
- Request 500 MB file
- Applied Chi-square and KS-test on obtain result



A semantic firewall for Content Centric Networking

# CONCLUSION



# Conclusion

- Introduction of a first firewall implementation dedicated to CCN
  - Use case analysis
  - Grammar definition
  - Implementation
- Use of semantic tools
- Overhead of the firewall is neglectable
  
- Future Work
  - Rule reordering
  - Using Bloom filters

**THANK YOU FOR YOUR ATTENTION  
QUESTIONS?**