



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# Dissertation

## OpenFlow based control plane for Information-Centric Networking

Jeeva Rajendran  
[rajendrj@tcd.ie](mailto:rajendrj@tcd.ie)

Supervisor : Stefan Weber

# Synopsis

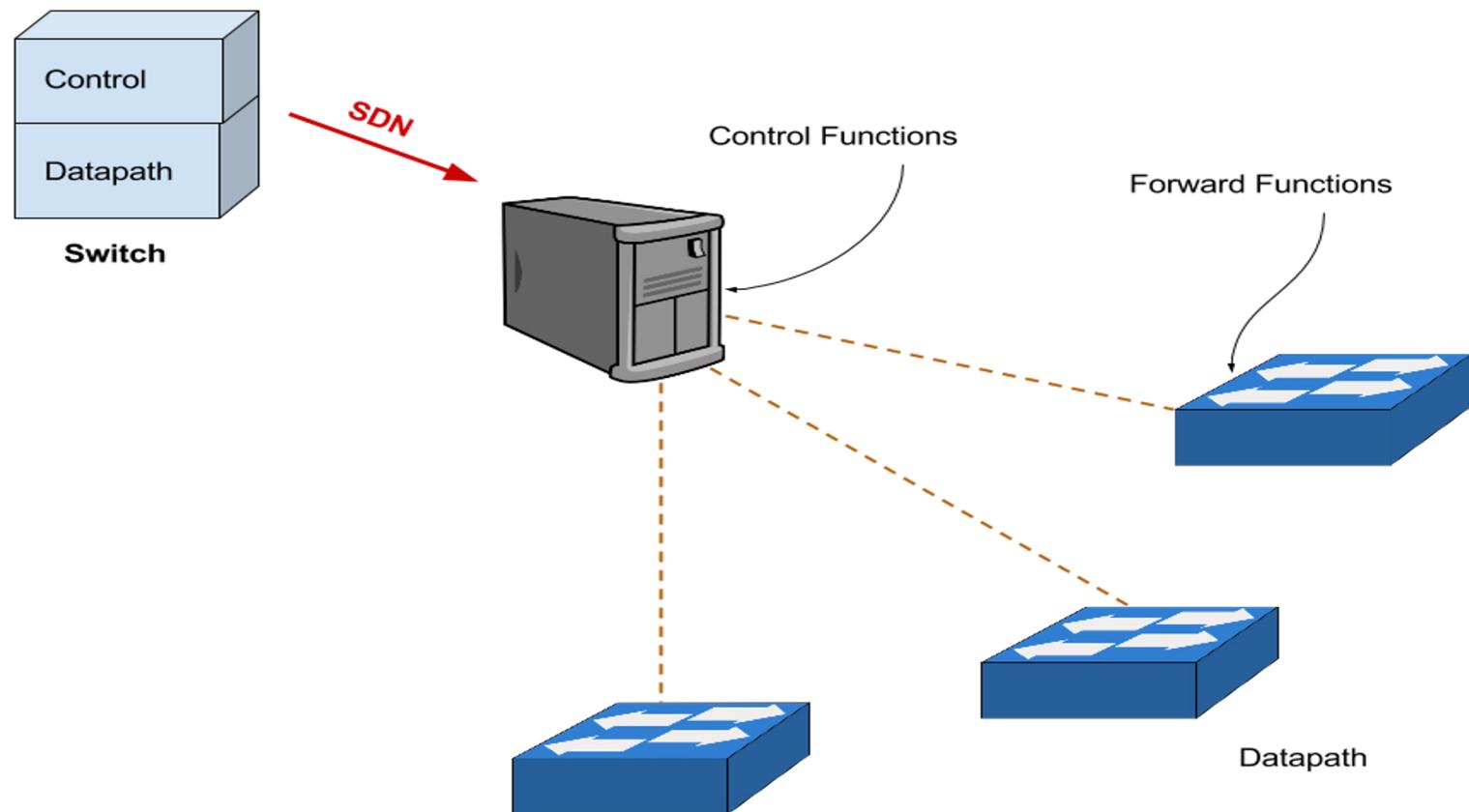
---

- Background
- Motivation
- State of Art
- Difference in approaches
- Problem formulation
- Design
- Reactive processing
- Implementation
- Evaluation and discussion
- Conclusion and Future work
- Demo

# Background

## SDN - Software Defined Networking

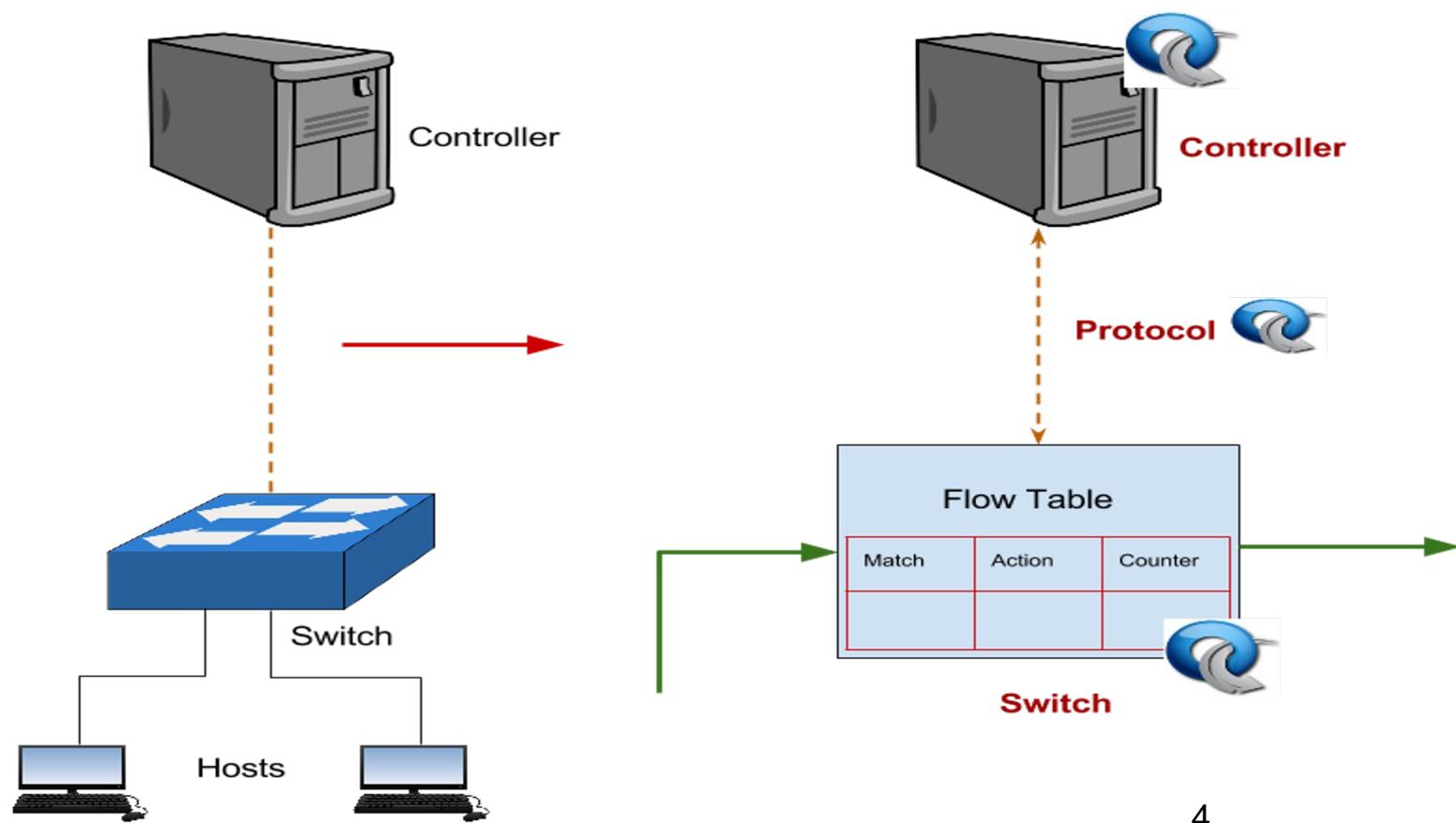
- Network programmability
- Separation between data plane and control plane



# Background

## OpenFlow - an SDN standard

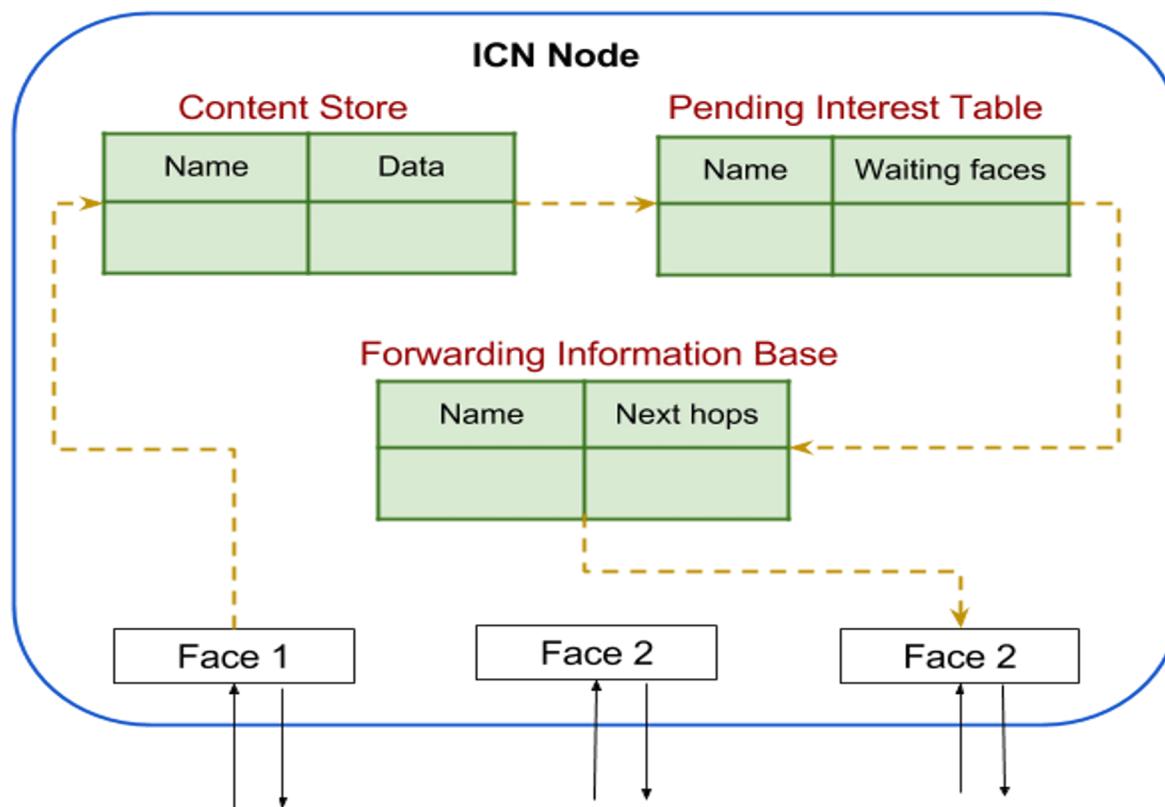
- Realises the programmable substrate with flow table
- API for programming network elements



# Background

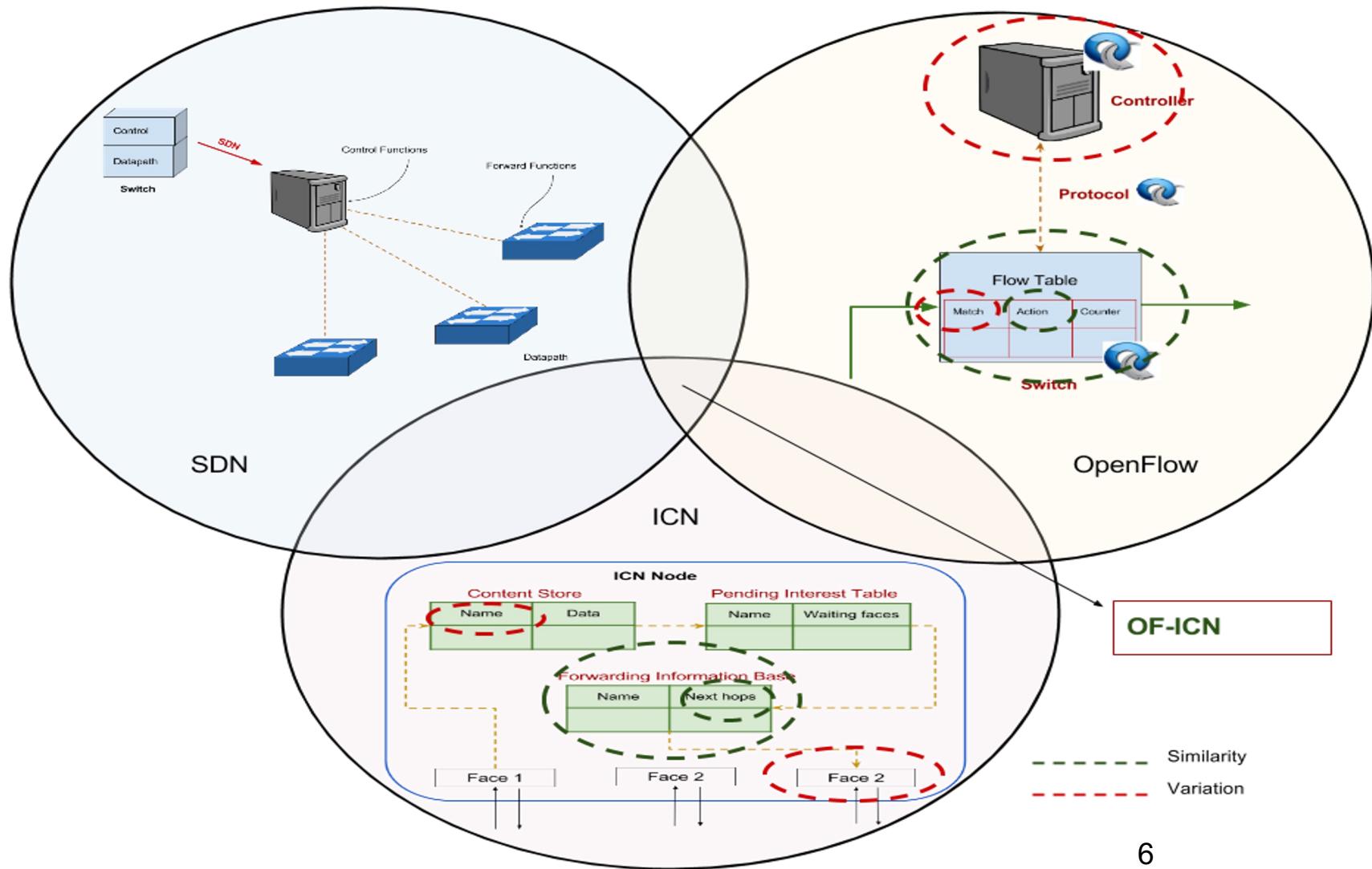
## ICN - Information Centric Networking

- Shift from “host-centric” to “content-centric”
- Future network architecture aims to solve the problems with IP and eventually replace it



# Motivation

What if we integrate them?



# Motivation

---

- Limelight research areas
- Benefits of OpenFlow over ICN
  - Centralized control
  - Flow based operations
  - Less-complex network elements
  - Clear separation between infrastructure and control policies
  - Reactiveness
- Move OpenFlow towards future internet architecture
  - Content based communication
  - non-IP support
  - Cache content
- Lack of work towards standardisation
- Untried gaps between the architectures

# State of Art : Non extension approaches

## Prominent studies:

- OFELIA Short term
- Wrapper
- SDN-NDNFlow
- Hierarchical Hashing
- Overlays
- Cache improvements
- Proxies

Features	2012	2013	2014	2016
<b>On Packet Format</b>	<p>Syriovelis et al <b>'Forward identifier'</b> to identify path between subscriber and publisher</p> <p>Blefari-Melazzi et al <b>'Tag'</b> the packet with ICN related information</p>	<p>Nguyen et al <b>'Hash'</b> the content name and add to an IP field</p> <p>Ooka et al <b>'Hierarchical hashing'</b> of content names to support LPM</p>	<p>Eum et al <b>'Data identifier'</b> in addition to forward identifier to identify content</p>	
<b>On providing ICN functionalities</b>		<p>Blefari-Melazzi et al <b>Short term approach over 'OFELIA'</b> testbed</p> <p>Nguyen et al <b>'Wrapper'</b> to convert content name into hash</p> <p>Song et al Process ICN based packets using <b>bytes</b> through <b>'Protocol Oblivious Forwarding'</b></p> <p>Vahlenkamp et al ICN as service using <b>public network address</b> saved in controller</p> <p>Ravindran et al Register ICN as <b>service</b> with orchestrator using an identifier</p>	<p>Shailendra et al Controller acting like DNS server for <b>name to address resolution</b></p> <p>Van et al <b>NDNFlow</b> - additional plugins for switch and controller</p> <p>Luo et al <b>CoLoR based controller</b> for better cache management</p> <p>Bacher et al Central SDN controller for multimedia dissemination</p> <p>Mougy et al <b>Distributed controller</b> for scalability</p>	<p>Trajano et al <b>Proxies with DHT</b> to identify nearby cache servers</p> <p>Xiulei et al <b>Data centres as cache</b> servers controlled by a controller</p> <p>Torres et al ICN based interest packets to communicate between switch and controller</p>

# **State of Art : What is the need for extension ?**

---

- From the drawbacks of non-extension approaches
  - Efforts to create ICN tags
  - Costly encoding and decoding operations on tags
  - IP field semantic changes
  - Processing delays due to additional modules and plugins
  - Costly requirements on extra machines like proxies
  - Partly based on end-point IP addresses

# State of Art : Extension based approaches

Prominent studies:

- Metadata extraction
- OFELIA long term
- Experimenter & Vendor messages
- Interest aggregation
- Switch architecture

Features	2012	2013	2014	2015/2016
<p>On OpenFlow extensions to support ICN</p>	<p>Salsano et al Long term approach over OFELIA testbed by extending OpenFlow protocol to support ICN</p> <p>Salsano et al Experimenter message based extensions</p> <p>Carvalho et al Extended switch with separate cache server</p> <p>Suh et al ICN-enabled OpenFlow switch architecture &amp; packet format outline</p>	<p>Chanda et al Metadata extraction using extended controller messages</p>	<p>Chang et al CDN-like caching solution using extended controller</p>	<p>Li et al Extended OpenFlow switch with multiple tables</p> <p>Scheider et al Network-level Interest aggregation using extended OpenFlow controller</p>

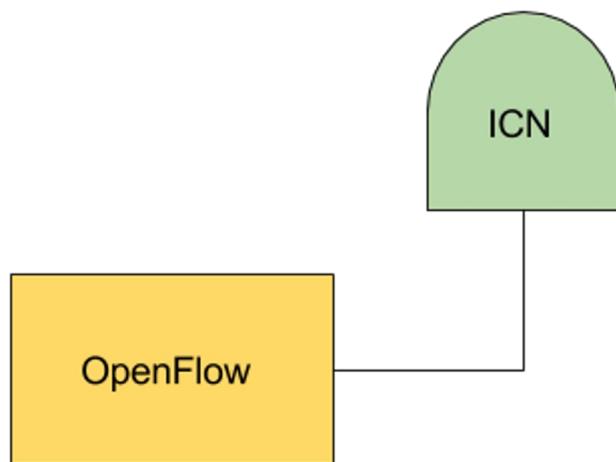
# **State of Art : What is missing ?**

---

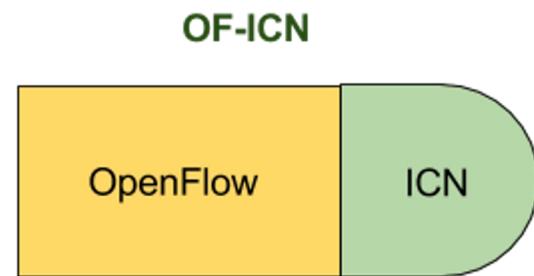
- What is missing in extension-based literature
  - Assumption on ICN-enabled switches
  - Non-OpenFlow controller
  - Standardization issues with experimenter and vendor messages
  - Agnostic to implementation details
  - Agnostic to design and feasibility of the proposed architecture
  - Mostly conceptual

# State of Art : How we are different ?

Most existing approaches



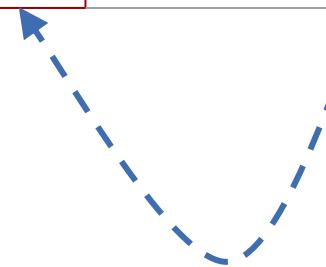
Our approach



# Problem Formulation - Abstracted view

- Abstracted comparison between OpenFlow and ICN

Feature	OpenFlow		ICN
Packet Format	IP		Interest, Data
Matching	IP Header Fields	OF-ICN	Longest Name Prefix Match
Forwarding	Flow table		CS, PIT, FIB
Caching	Flow rule cache		Content cache



# Problem Formulation - Challenges

---

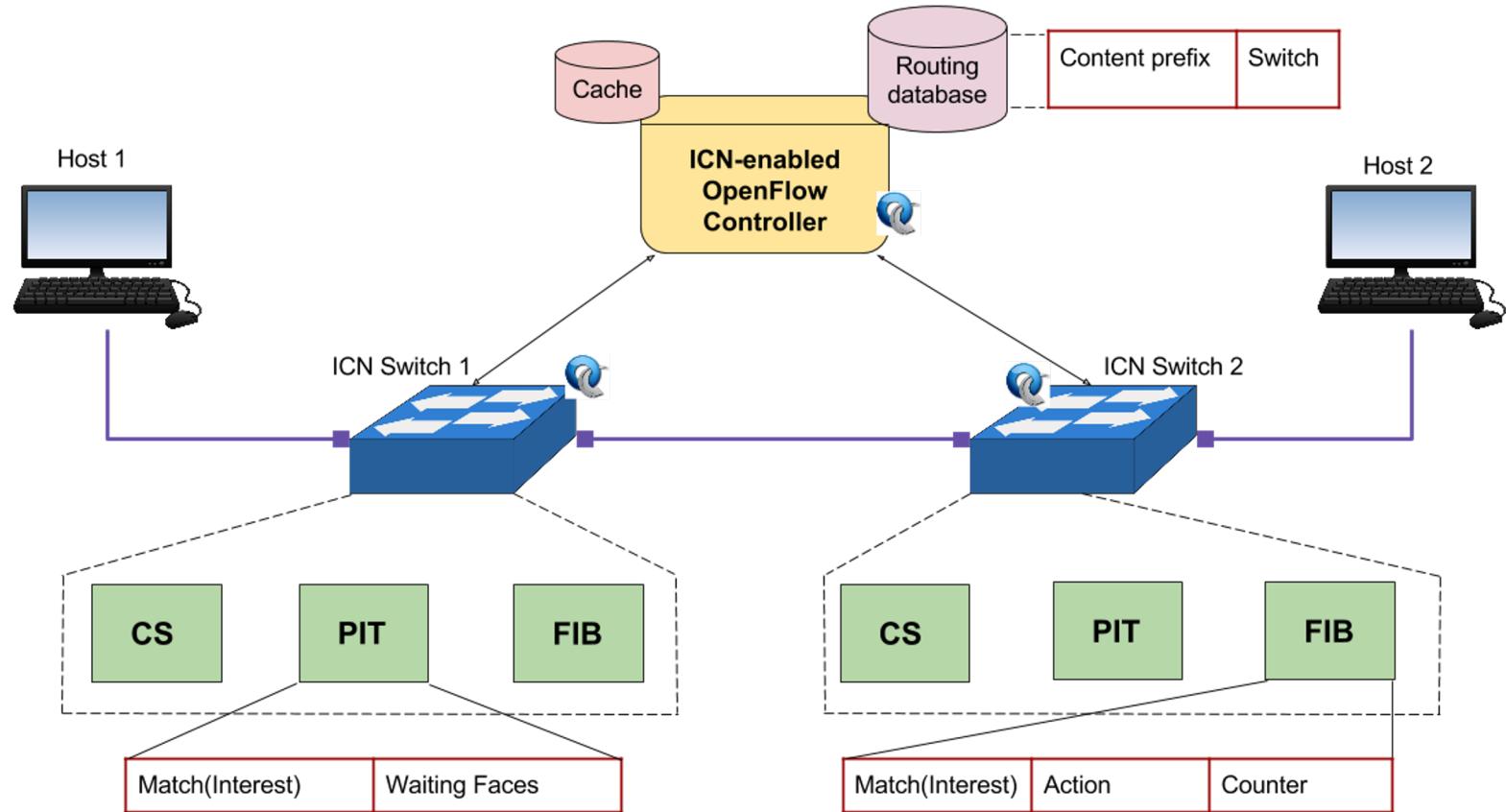
- Design level challenges
  - IP to non-IP support
  - Emerging technologies
  - ICN naming issues
  - Standardization and time
  - Multi table support
- Implementation level challenges
  - Binary protocol
  - Specification supports

# Problem Formulation - Targets

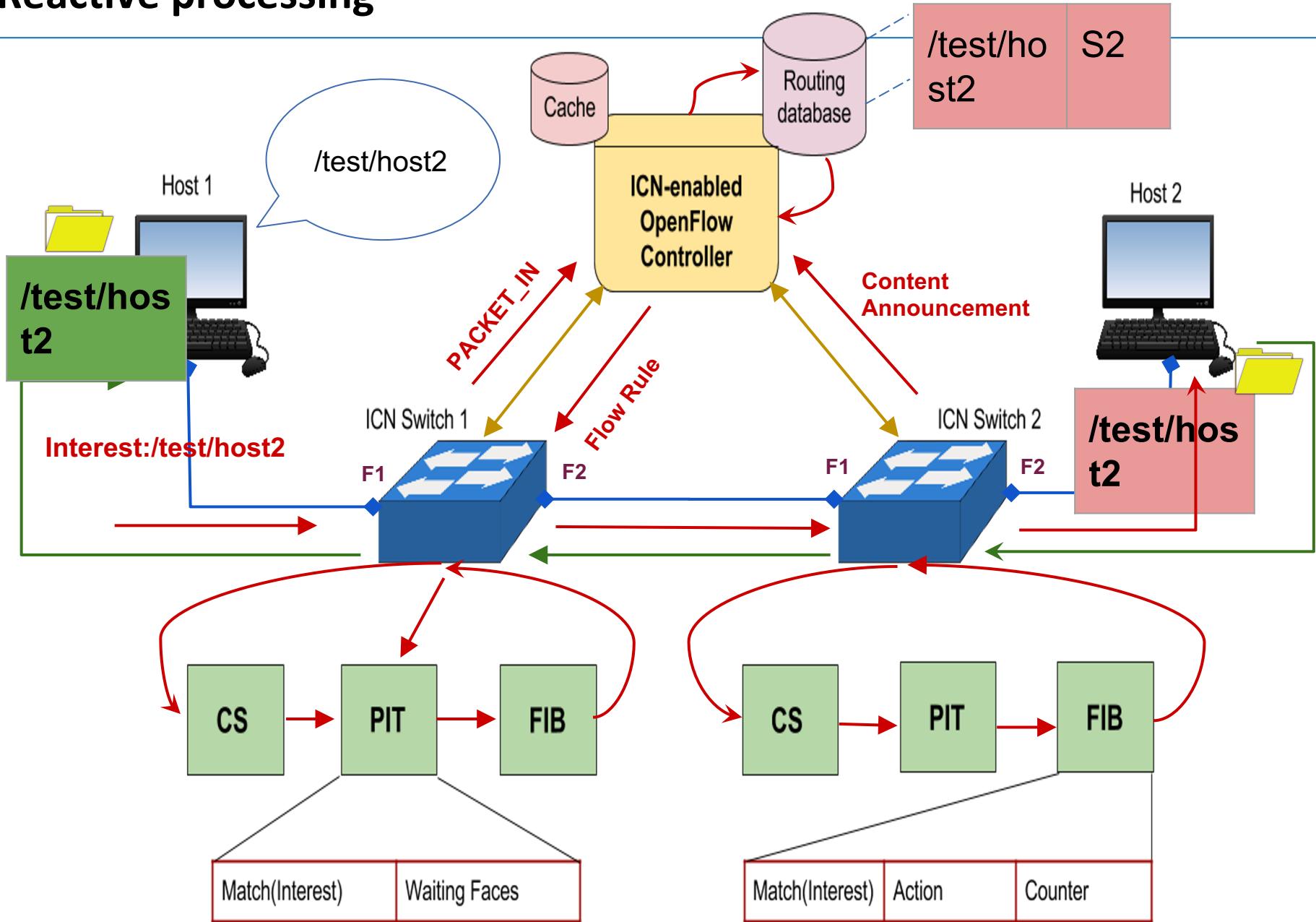
---

- Introduce face abstraction
- Inspect names in the packet
- Consider available similarities between OpenFlow and ICN
- Touch base experimented extensions on OpenFlow
- Introduce new messages
- Introduce new actions
- Introduce new handlers
- Introduce multiple tables aligning to ICN
- Provide flow based communication for ICN
- Be close to standards

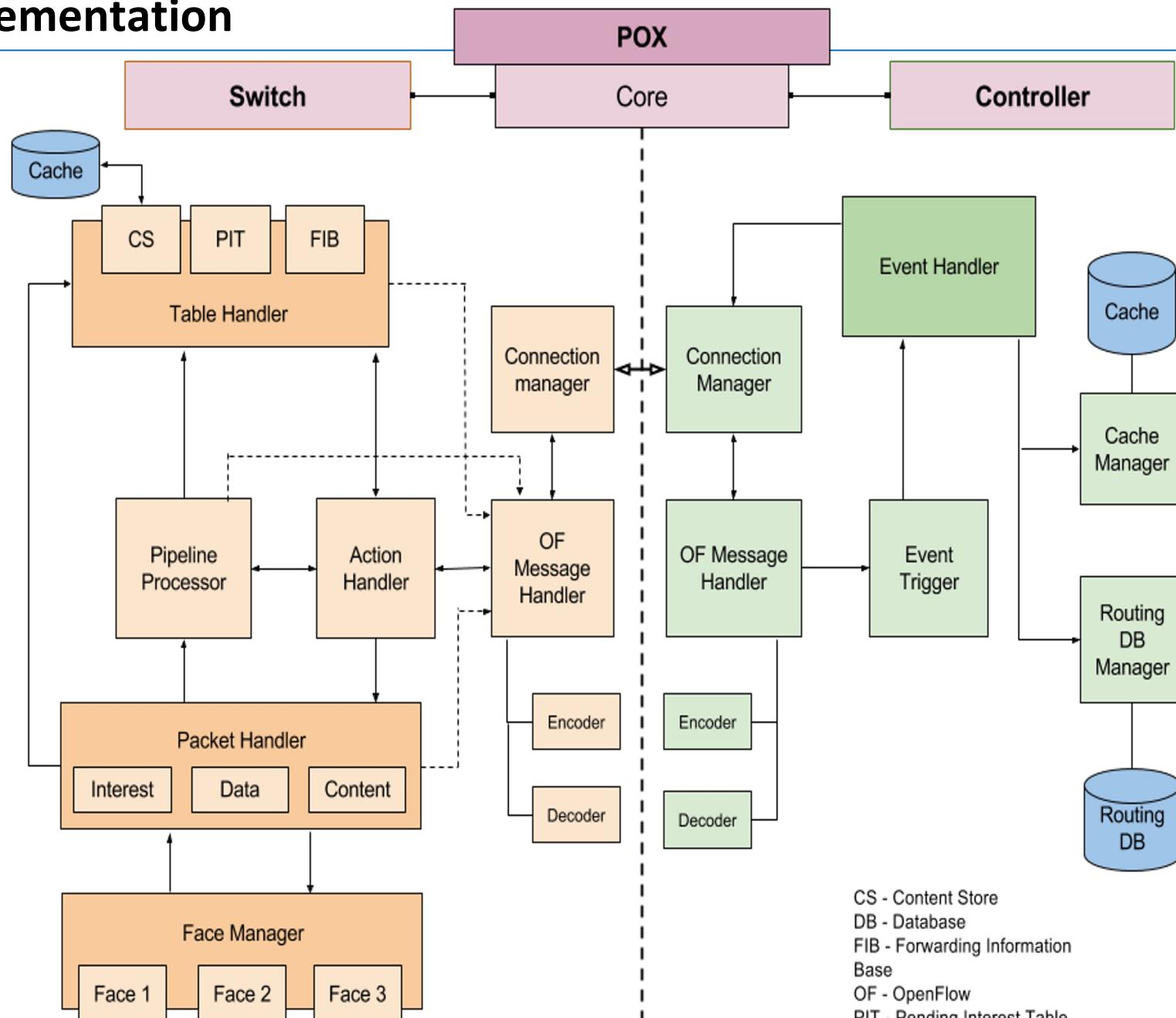
# OF-ICN Design



# Reactive processing



# Implementation



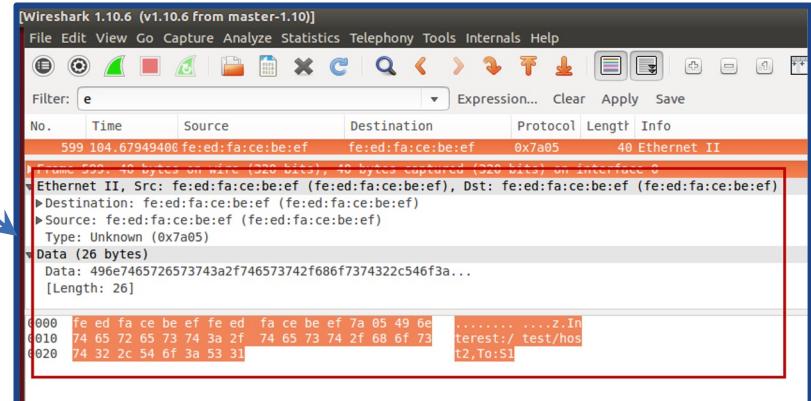
## Evaluation Categories

To ensure:

- Modified OpenFlow affords ICN functionalities
- Basic features of OpenFlow are still preserved
- Easy deployment
- Benefits of OpenFlow over ICN implementations

# ICN Functionalities in OF-ICN

- Labeling information instead of hosts
- Requests aggregation
- Multicasting
- Scalability
- Compatibility with IP
- Flow rules
- In-network caching
- Security
- Flow balancing
- Multiple tables
- Forward-by-name
- FIB management
- Interest and data forwarding
- Caching



PIT : Prefix	Waiting Faces	PIT : Prefix	Waiting Faces
/test/switch1/pittest	2	/test/switch1/pittest	21

```
Rajendra@Jeeva:~/pox/pox$ Content Announcement
Hi User .. You can send an Interest at any time. Just Enter the interest in the terminal
/test/host2 -> Interest for which there is no content in switch's cache

Data: Hi This is Host 2 I got your request How are you?

(' RTT in seconds :', 0.15473599999999976)
/test/switch1csmatch -> Interest for which the switch has content in cache

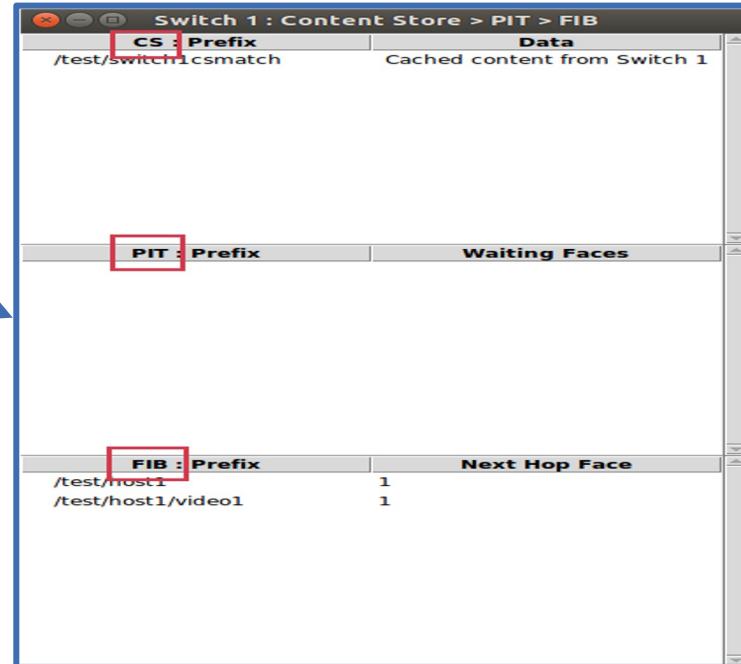
Data: Cached content from Switch 1

(' RTT in seconds :', 0.0003530000000004918)
```

# ICN Functionalities in OF-ICN

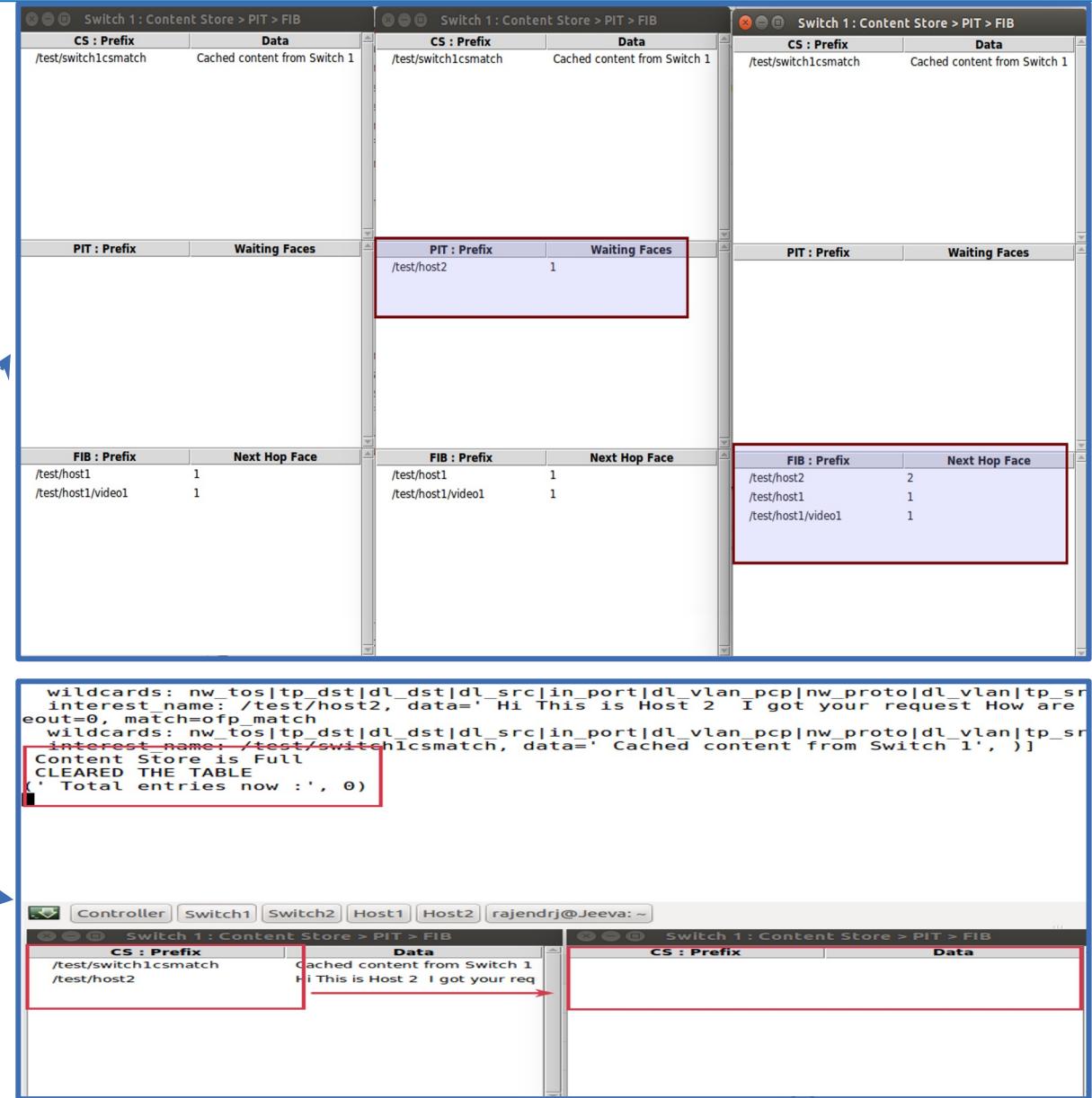
- Labeling information instead of hosts
- Requests aggregation
- Multicasting
- Scalability
- Compatibility with IP
- Flow rules
- In-network caching
- **Security**
- Flow balancing
- **Multiple tables**
- Forward-by-name
- FIB management
- Interest and data forwarding
- Caching

```
53 50 .00770000 73:74:3a:2f:74:65 49:6e:74:65:72:65 OF 1.0 104 of packet_in
►Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
►Transmission Control Protocol, Src Port: 56046 (56046), Dst Port: 6633 (6633), Seq: 268, Ack: 301, Len: 38
▼OpenFlow (LOXI)
version: 1
type: OFPT_PACKET_IN (10)
length: 38
xid: 0
buffer_id: 1
total_len: 20
in_port: 1
reason: OFPR_NO_MATCH (0)
▼Ethernet packet
►Ethernet II, Src: 73:74:3a:2f:74:65 (73:74:3a:2f:74:65), Dst: 49:6e:74:65:72:65 (49:6e:74:65:72:65)
▼Data (6 bytes)
Data: 2f686f737432
[Length: 6]
0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..E.....
0010 00 5a e8 3c 40 00 40 06 54 5f 7f 00 00 01 7f 00 .Z.<@. T.....
0020 00 01 da ee 19 e9 18 e1 d3 19 24 cf da 6e 80 18 .....$.n...
0030 01 5e fe 4e 00 00 01 01 08 0a 00 0d 6b 12 00 0d .^N.... .k...
0040 65 24 01 0a 00 26 00 00 00 00 00 00 00 01 00 14 es...&.....
0050 00 01 00 00 49 6e 74 65 72 65 73 74 3a 2f 74 65 ...Inte rest:/te
0060 73 74 2f 68 6f 73 74 32 st/host2
```



# ICN Functionalities in OF-ICN

- Labeling information instead of hosts
- Requests aggregation
- Multicasting
- Scalability
- Compatibility with IP
- Flow rules
- In-network caching
- Security
- Flow balancing
- Multiple tables
- Forward-by-name
- FIB management
- **Interest and data forwarding**
- **Caching**



22

# ICN Functionalities in OF-ICN

S.No	Test Case	Sub Test Case	Action	Message/ Event/ Action
1	Content Store Match		Send Data packet back in incoming face	PACKET_OUT[FACE]
2	PIT Match		Add the incoming face to waiting list	ADD_PIT[FACE]
3	FIB Match		Forward the interest in outgoing face	PACKET_OUT[FACE]
4	FIB Match		Add the incoming face to PIT	ADD_PIT[FACE]
5	No match in switch	Controller has cached data	Send the data packet from controller to switch	OFPT_DATA_FROM_CONTROLLER
6	No match in switch	Controller has the route	Send Flow Rule to switch	FLOW_MOD[RULE] PACKET_OUT[FACE]
7	Second interest for the same content		Match with FIB in the switch and send the packet in outgoing face	PACKET_OUT[FACE]
8	Data packet	Matches with PIT	Send the data packet back in the waiting face	PACKET_OUT[FACE]
9	Data packet	No match with PIT	Drop the data packet	
10	Content Store Full		Message from controller to clear the cache in switch	OFPT_CS_FULL OFPT_CLEAR_CS
11	Proactive caching		Cache the content from controller in the switch	OFPT_ADD_CS_ENTRY
12	Content Announcement		Store the content name and the switch identifier in controller routing database	OFPT_CONTENT_ANNOUNCEMENT

# Preserving OpenFlow features

---

- Switch-Controller handshake
  - Capability advertisement
  - Keep-alive with controller
  - Network functions separation
  - Flow based operations
  - Content announcement from switch instead of host
  - Global view
- 
- Implementation level evaluation
    - Easy deployment
      - POX modularized components

# Comparison with existing ICN implementations

## CCNPing

- Ping tool based on CCN architecture
- ccnping and ccnpingserver

### CCNPing

```
rajendrj@Jeeva:~/ccnping$ sudo ccnping ccnx:/test/video1
CCNPING ccnx:/test/video1
content from ccnx:/test/video1: number = 457521482
    rtt = 4.856 ms
content from ccnx:/test/video1: number = 785709862
    rtt = 0.607 ms
content from ccnx:/test/video1: number = 815833959
    rtt = 0.667 ms
content from ccnx:/test/video1: number = 800686715
    rtt = 0.628 ms
content from ccnx:/test/video1: number = 478473237
    rtt = 0.606 ms
..
```

### OF-ICN

```
rajendrj@Jeeva:~/pox/pox$ sudo python host1.py
Content Announcement
Hi User ... You can send an Interest at any time. Just Enter the interest in ther terminal /test/host2

Data: Hi This is Host 2 I got your request How are you?

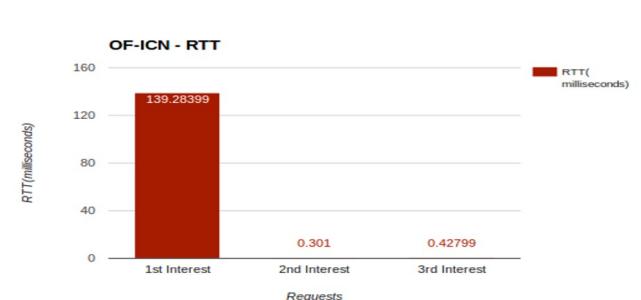
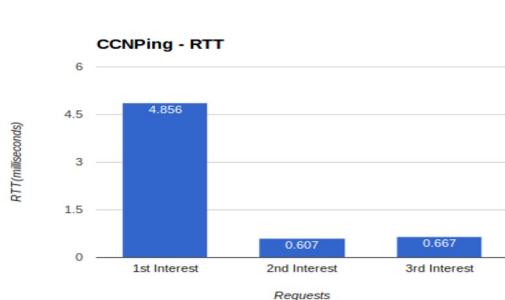
(' RTT in seconds :', 0.13928399999999996)
/test/host2

Data: Hi This is Host 2 I got your request How are you?

(' RTT in seconds :', 0.0003010000000003288)
/test/host2

Data: Hi This is Host 2 I got your request How are you?

(' RTT in seconds :', 0.0004279999999942884)
```

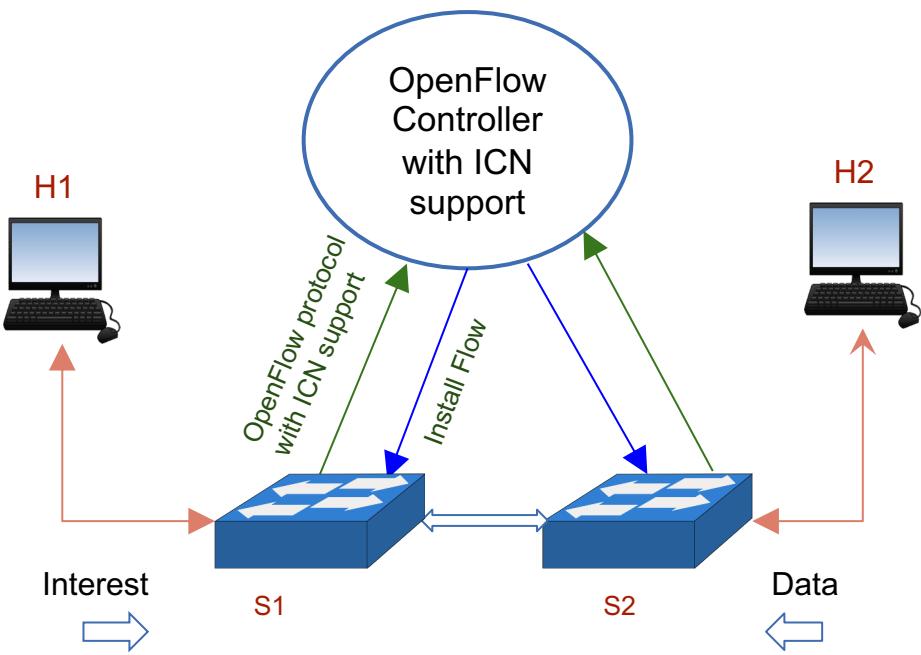
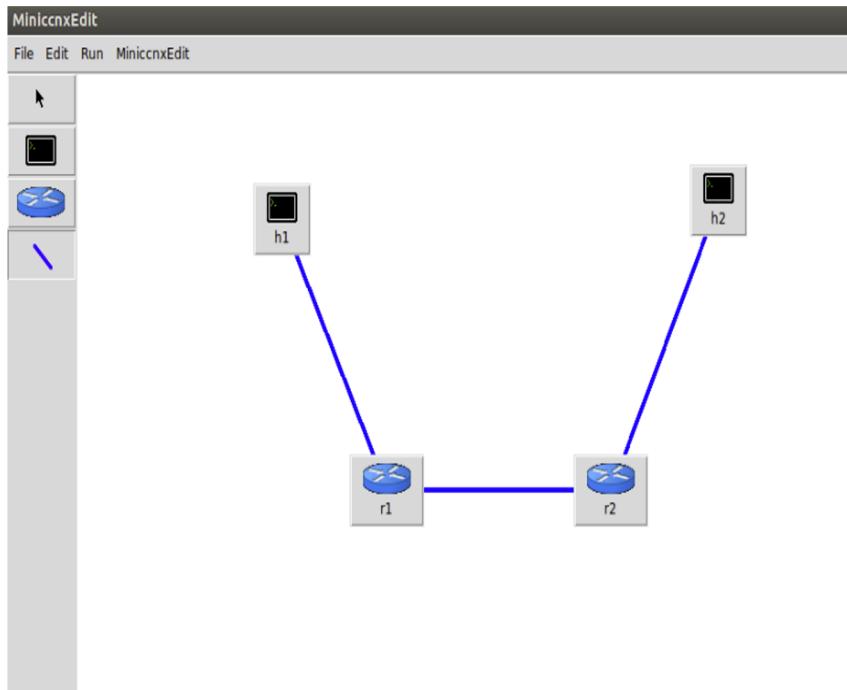


# Comparison with existing ICN implementations

## Mini-ccnx

- Mininet based ICN prototyping
- CCN daemon
- Multiple topologies

## Topologies



# Comparison with existing ICN implementations

## Test 1

Scenario where all routers have enough forwarding details

### Mini-ccnx

```
XTerm
rajendrj@Jeeva:~$ sudo miniccnx
Parse of miniccnx.conf done.
*** Creating network
*** Adding hosts:
h1 h2 r1 r2
*** Adding links:
(h1, r1) (h2, r2) (r1, r2)
*** Configuring hosts
h1 h2 r1 r2
*** Adding metrics collectors:
Host h1: database miniccnx_data already exists: appending
Host h2: database miniccnx_data already exists: appending
Host r1: database miniccnx_data already exists: appending
Host r2: database miniccnx_data already exists: appending
Setup time: 0

*** Done

*** Starting CLI:
miniccnx> ccndump
h1: ('ccnx:/h2/video1', 'r1')
h2:
r1: ('ccnx:/h2/video1', 'r2')
r2: ('ccnx:/h2/video2', 'h2') ('ccnx:/h2/video1', 'h2')
miniccnx> xterm n1 n2
miniccnx>
```

The screenshot shows the Mini-CCNx terminal window. It displays the configuration process, including adding hosts, links, and metrics collectors. The 'ccndump' command is run to show the network topology, which includes hosts h1, h2, r1, r2 and their connections. A separate terminal window titled 'Node: h1' shows the results of a 'ccnping' command, displaying seven round-trip times (RTTs) between h1 and h2. Another terminal window titled 'Node: h2' shows the results of a 'ccnpingserver' command.

RTT (milliseconds)
17.642
1.136
1.269
1.107
1.154
1.153
1.170

### OF-ICN

```
rajendrj@Jeeva:~/pox/pox/hosts$ sudo python host1.py
Content Announcement
Hi User .. You can send an Interest at any time. Just Enter the interest in the terminal
/test/host2

Data: Hi This is Host 2 I got your request How are you?

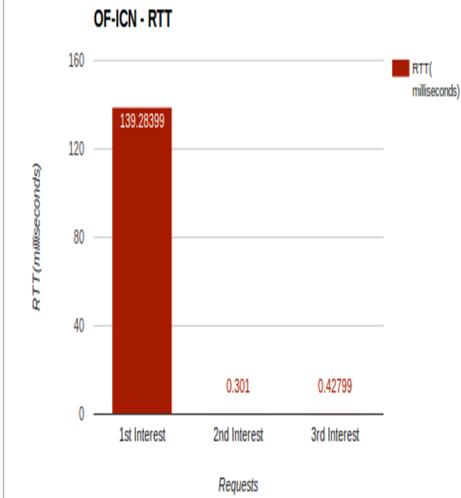
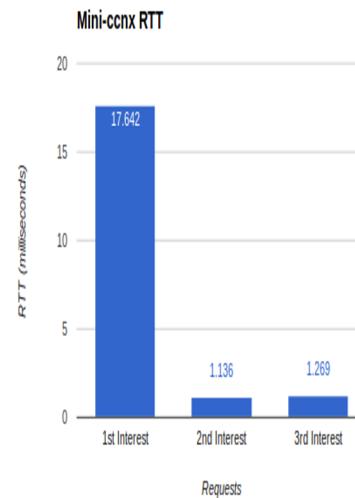
(' RTT in seconds :', 0.13928399999999996)
/test/host2

Data: Hi This is Host 2 I got your request How are you?

(' RTT in seconds :', 0.000301000000003288)
/test/host2

Data: Hi This is Host 2 I got your request How are you?

(' RTT in seconds :', 0.0004279999999942884)
```



# Comparison with existing ICN implementations

## Test 2

Scenario where some routers have less forwarding details

### Mini-ccnx

XTerm  
rajendrj@Jeeva:~\$ sudo miniccnx  
Parse of miniccnx.conf done.  
\*\*\* Creating network  
\*\*\* Adding hosts:  
h1 h2 r1 r2  
\*\*\* Adding links:  
(h1, r1) (h2, r2) (r1, r2)  
\*\*\* Configuring hosts  
h1 h2 r1 r2  
\*\*\* Adding metrics collectors:  
Host h1: database miniccnx\_data already exists: appending  
Host h2: database miniccnx\_data already exists: appending  
Host r1: database miniccnx\_data already exists: appending  
Host r2: database miniccnx\_data already exists: appending  
Setup time: 0  
\*\*\* Done  
\*\*\* Starting CLI:  
miniccnx> ccndump  
h1: ('ccnx:/h2/video1', 'r1')  
h2:  
r1: ('ccnx:/h2/video1', 'r2')  
r2: ('ccnx:/h2/video2', 'h2') ('ccnx:/h2/video1', 'h2')  
miniccnx> xterm h1 h2  
miniccnx>

Node: h1  
root@Jeeva:~# ccnping ccnx:/h2/video2  
CCNPING ccnx:/h2/video2  
timeout from ccnx:/h2/video2: number = 2086848165  
timeout from ccnx:/h2/video2: number = 96252815  
timeout from ccnx:/h2/video2: number = 627247042  
timeout from ccnx:/h2/video2: number = 1081323388  
timeout from ccnx:/h2/video2: number = 101350006  
timeout from ccnx:/h2/video2: number = 522903545  
timeout from ccnx:/h2/video2: number = 557199382  
timeout from ccnx:/h2/video2: number = 1515924892  
timeout from ccnx:/h2/video2: number = 1424788241  
--- ccnx:/h2/video2 ccnping statistics ---  
13 Interests transmitted, 0 Data received, 100.0% packet loss, time 12182 ms  
root@Jeeva:~#

Node: h2  
root@Jeeva:~# ccnpingserver ccnx:/h2/video2

### OF-ICN

This is an Interest Packet → Upon receiving the interest  
No CS Entry found : Gonna look in PIT  
No PIT entry Found : Gonna look in the FIB  
No Matching Entry found in any of the tables : Sending to controller  
Sent the interest packet to controller : Add in PIT  
('Sent INTEREST packet to next hop face : ', 2)  
This is a Data Packet  
(' Faces to send the data :, [1])  
{'/test/host2': [1]}  
\*\*\*\* Gonna delete the pit entry  
{}  
('Sent DATA packet back in the requested face : ', 1)

Upon receiving the data



28

# Conclusion and Future work

---

- OF-ICN : An experimentation on extending OpenFlow to support ICN architecture and functionalities
- Contributions :
  - Analysis of challenges involved
  - OpenFlow modification using generic ICN architecture
  - Flow-based realisation of basic ICN functionalities
  - Experimentation on all three OpenFlow elements: Switch, controller and the protocol
  - New OpenFlow messages in binary format
  - Modularized implementation
- Future work :
  - Supporting future ICN naming scheme(once it is standardised)
  - Performance tuning
  - Security features
  - Scalability

# Demo

30



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# Questions ?



**Trinity College Dublin**  
Coláiste na Tríonóide, Baile Átha Cliath  
The University of Dublin

# Thank You