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

Supervisor : Stefan Weber

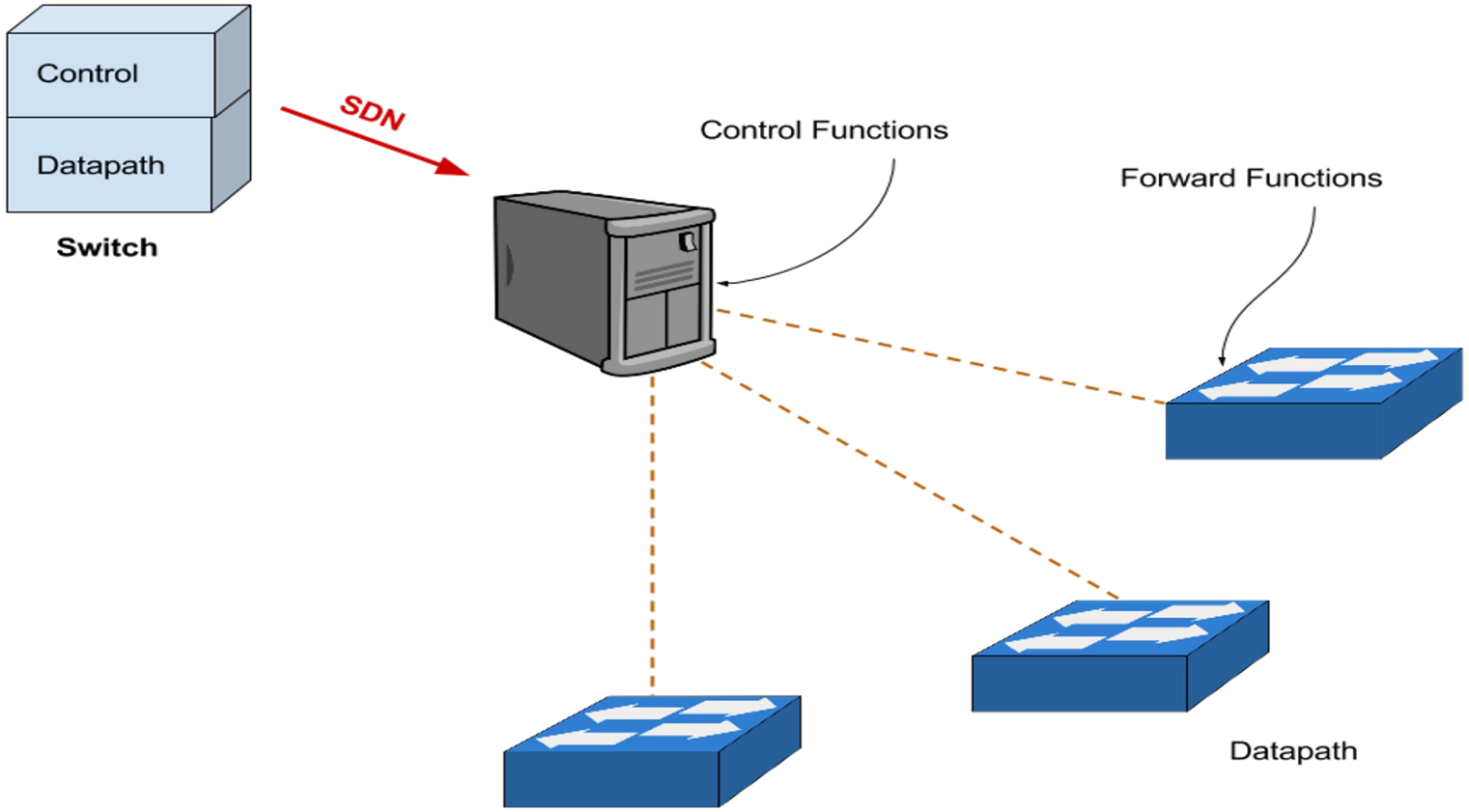
Synopsis

- 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

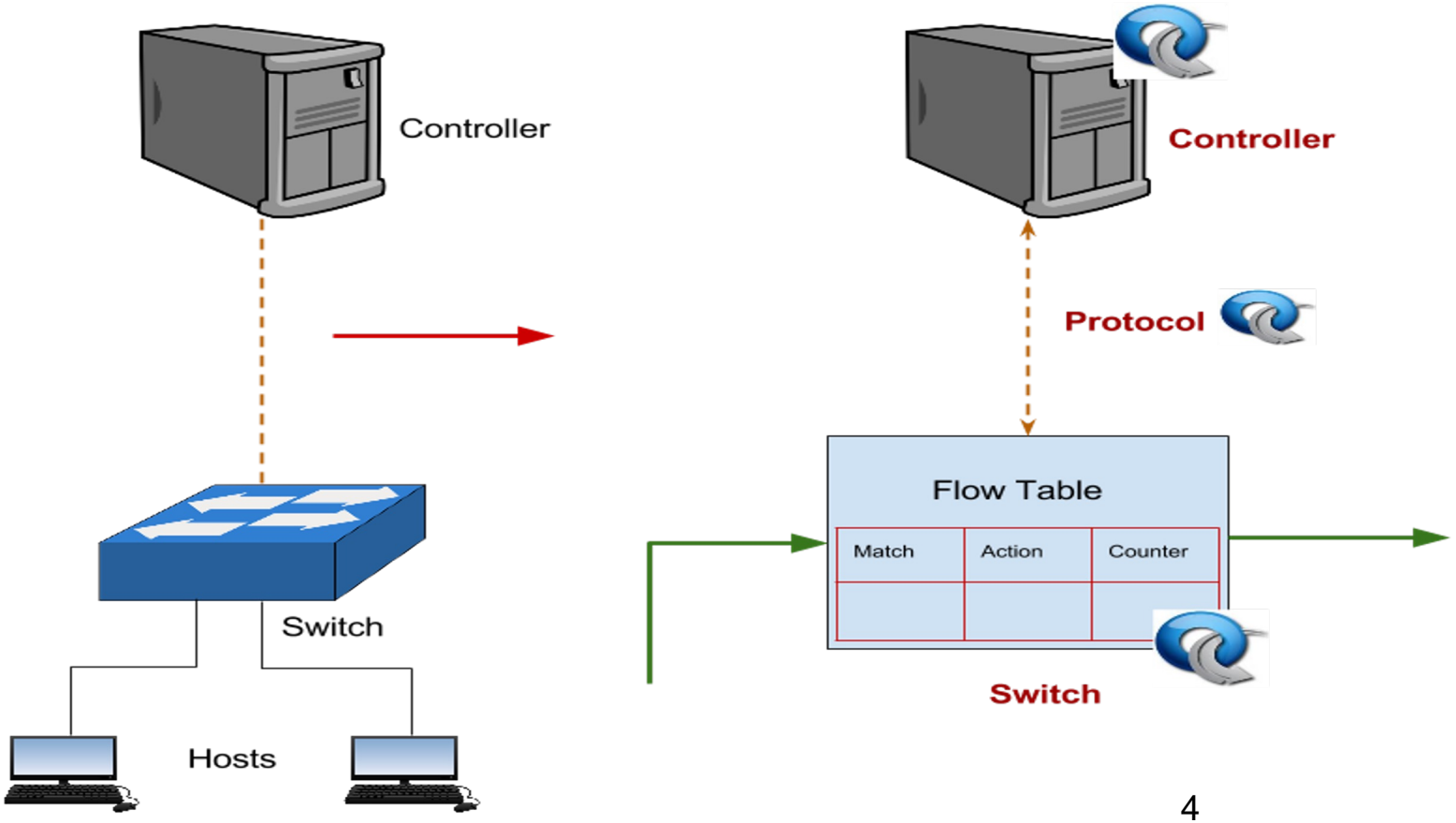
- Network programmability
- Separation between data plane and control plane



Background

OpenFlow - an SDN standard

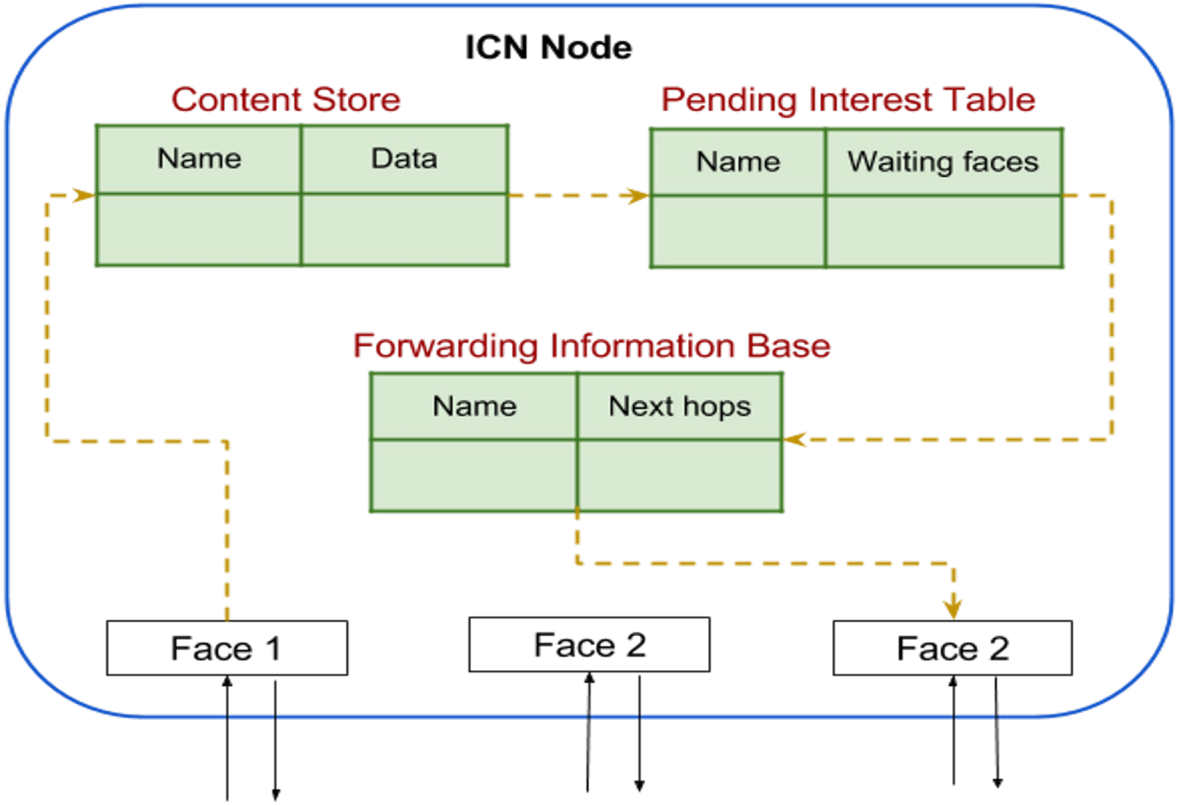
- Realises the programmable substrate with flow table
- API for programming network elements



Background

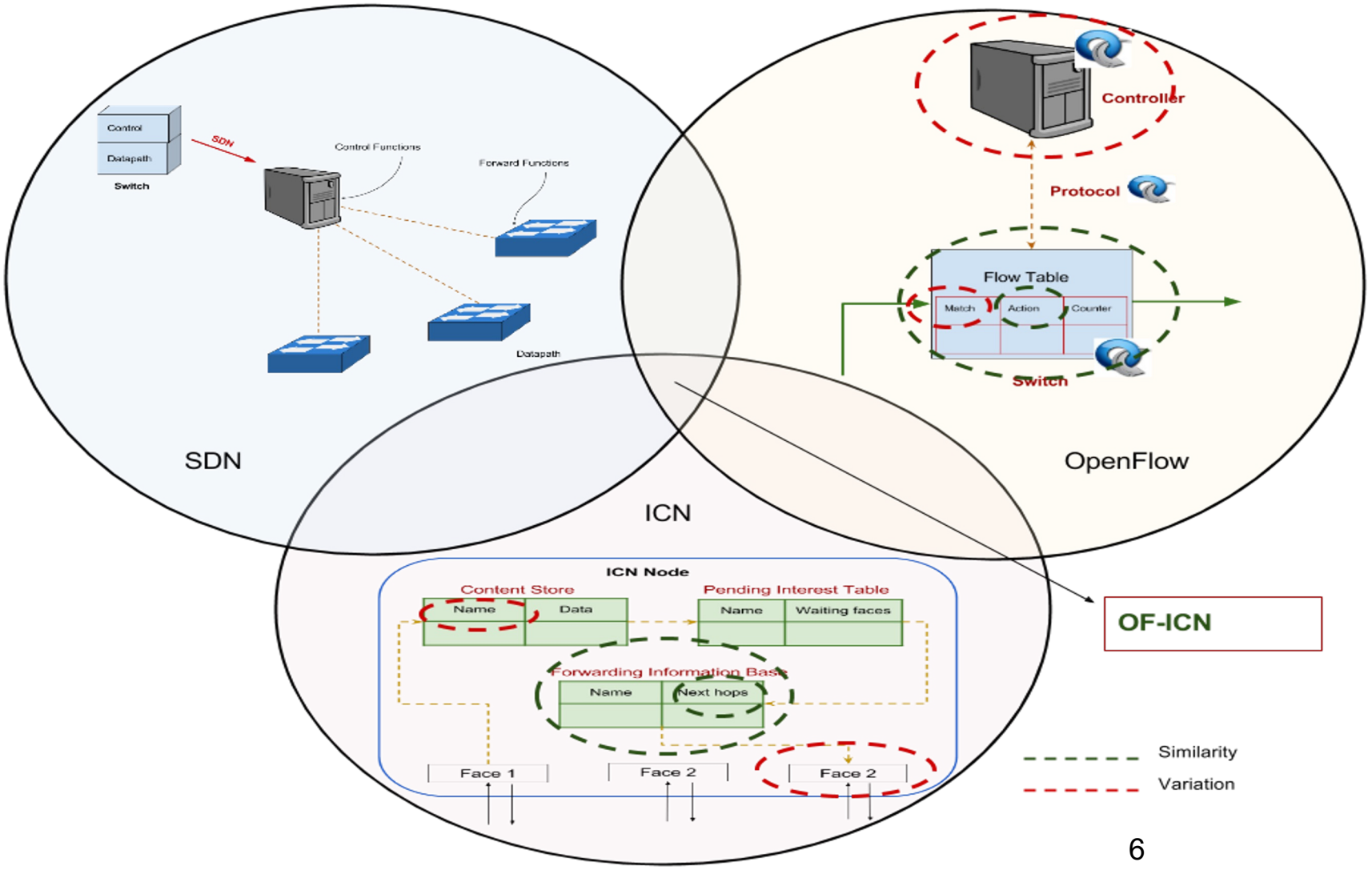
ICN - Information Centric Networking

- 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?



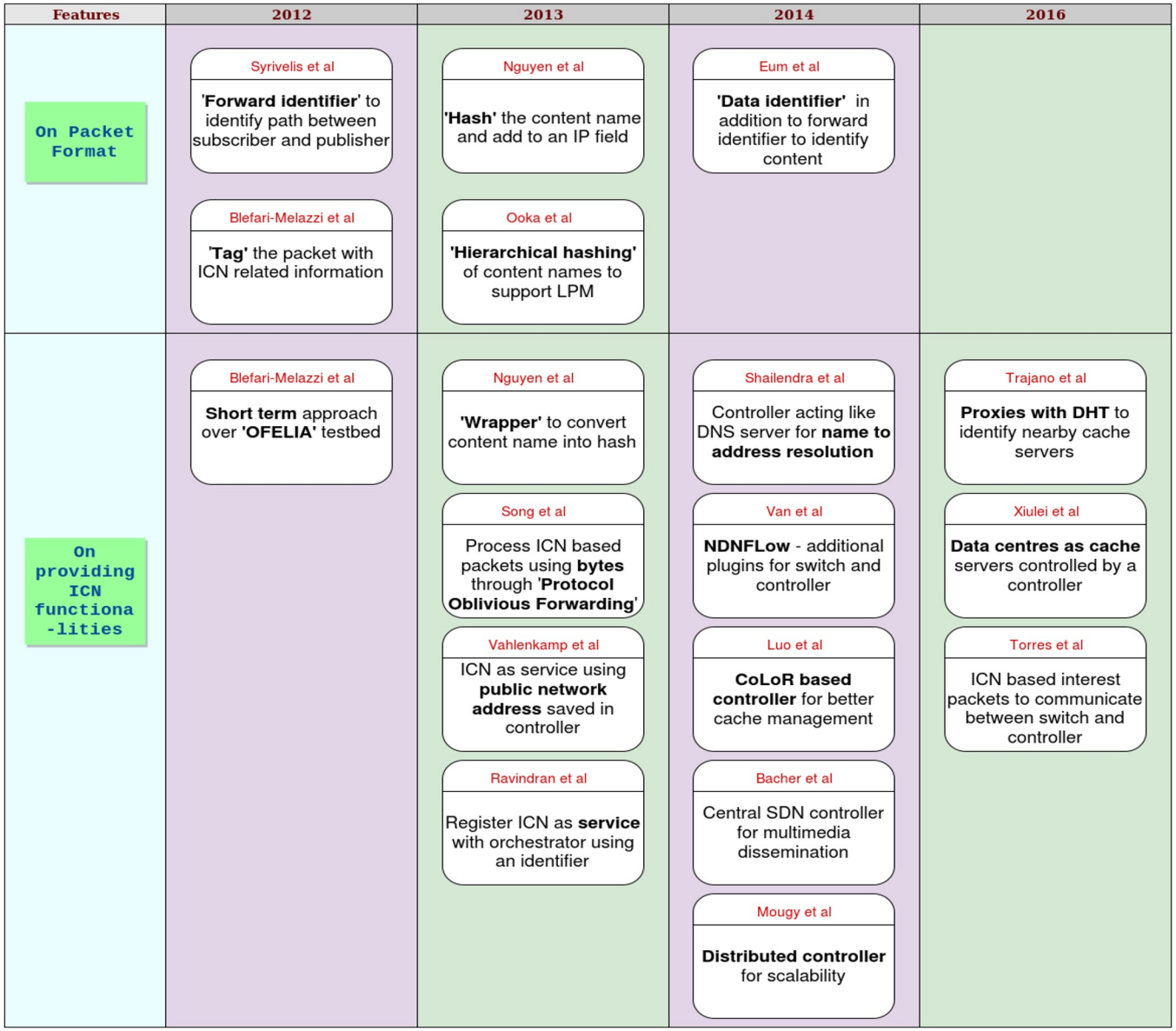
Motivation

- 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



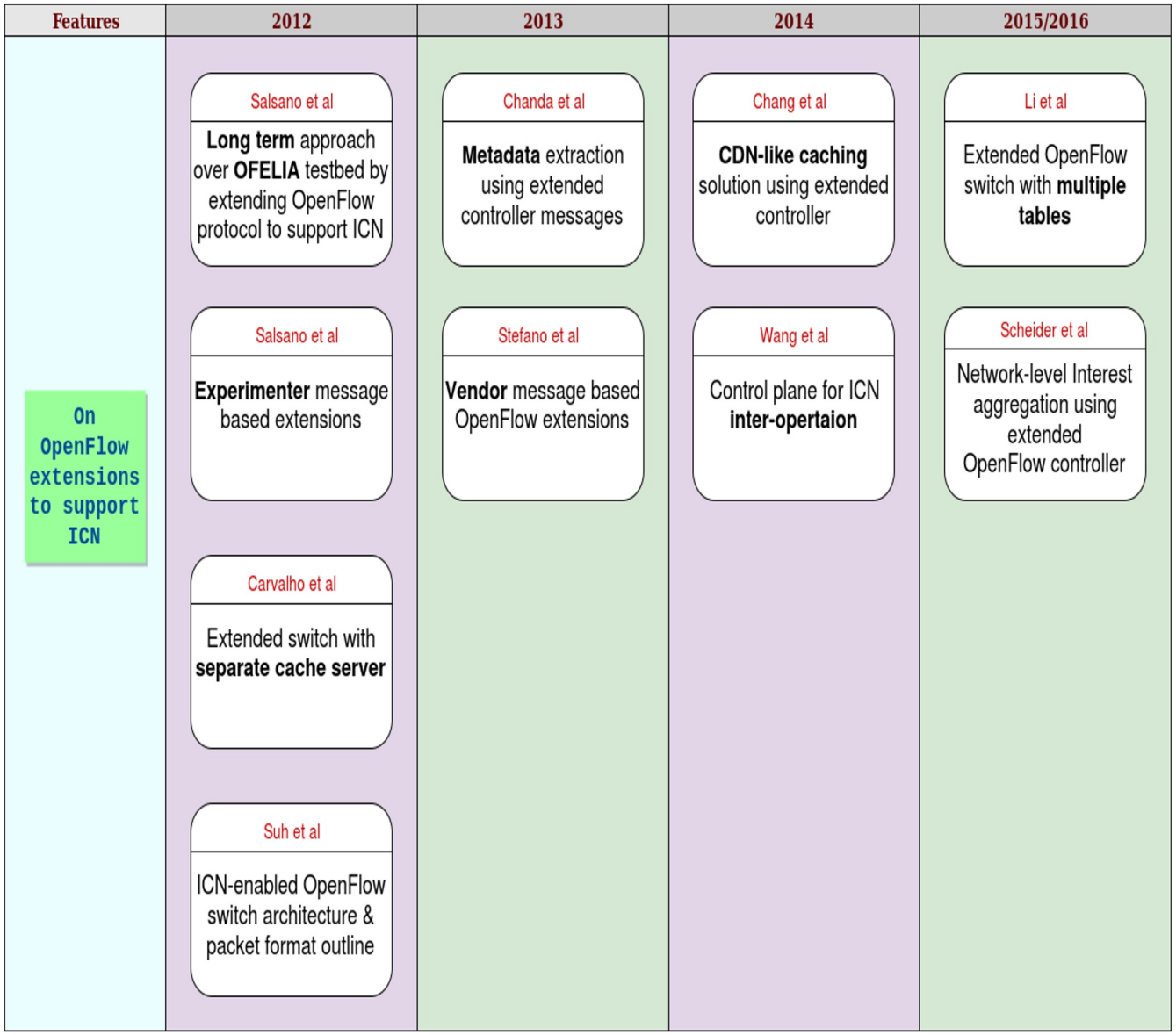
State of Art : What is the need for extension ?

- 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:

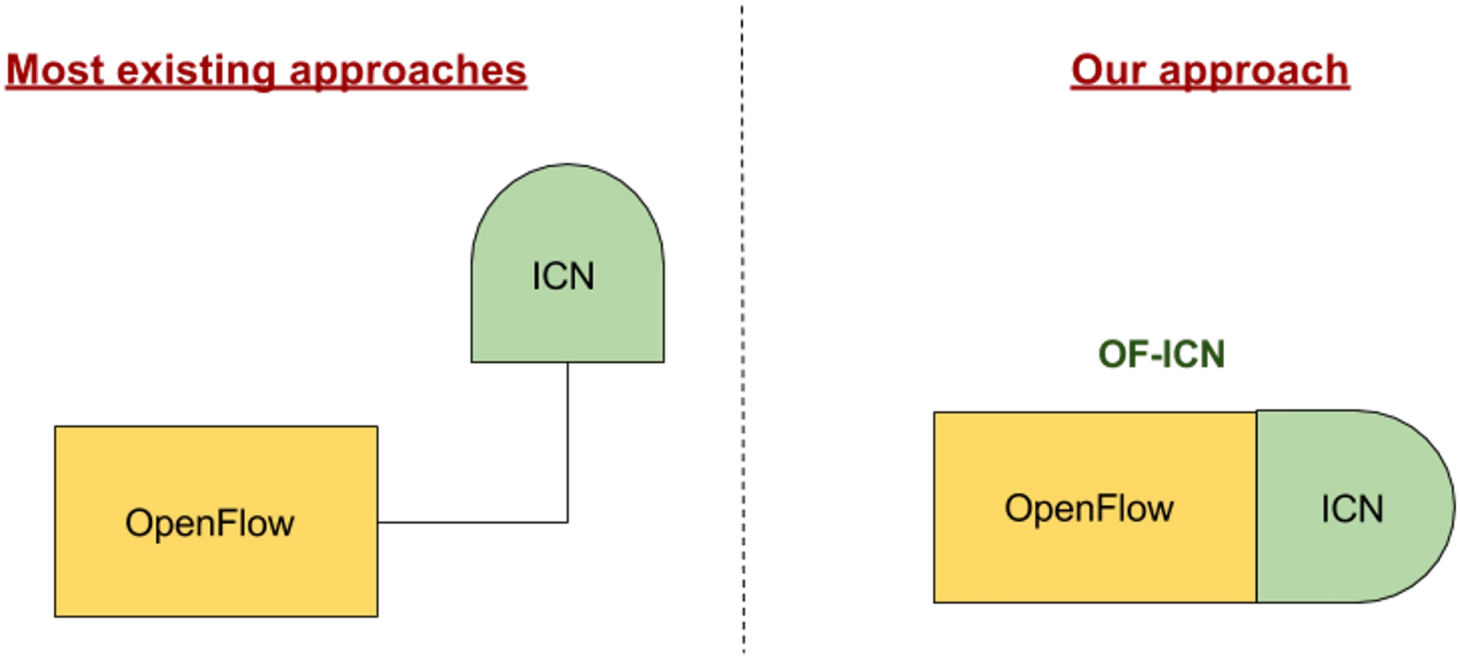
- Metadata extraction
- OFELIA long term
- Experimenter & Vendor messages
- Interest aggregation
- Switch architecture



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

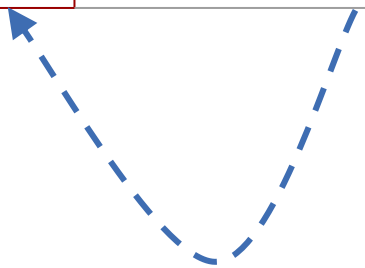
State of Art : How we are different ?



Problem Formulation - Abstracted view

- Abstracted comparison between OpenFlow and ICN

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



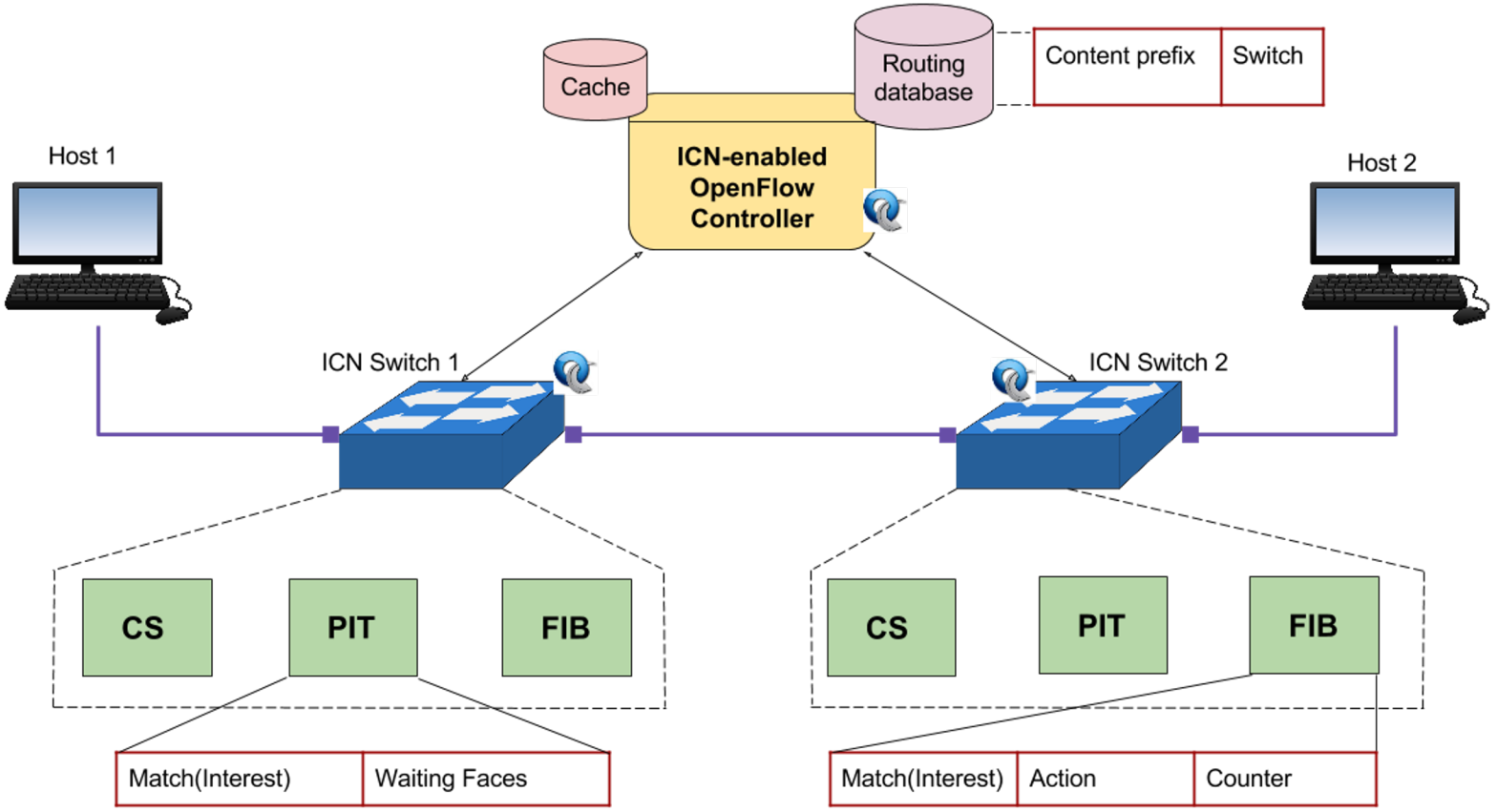
Problem Formulation - Challenges

- 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

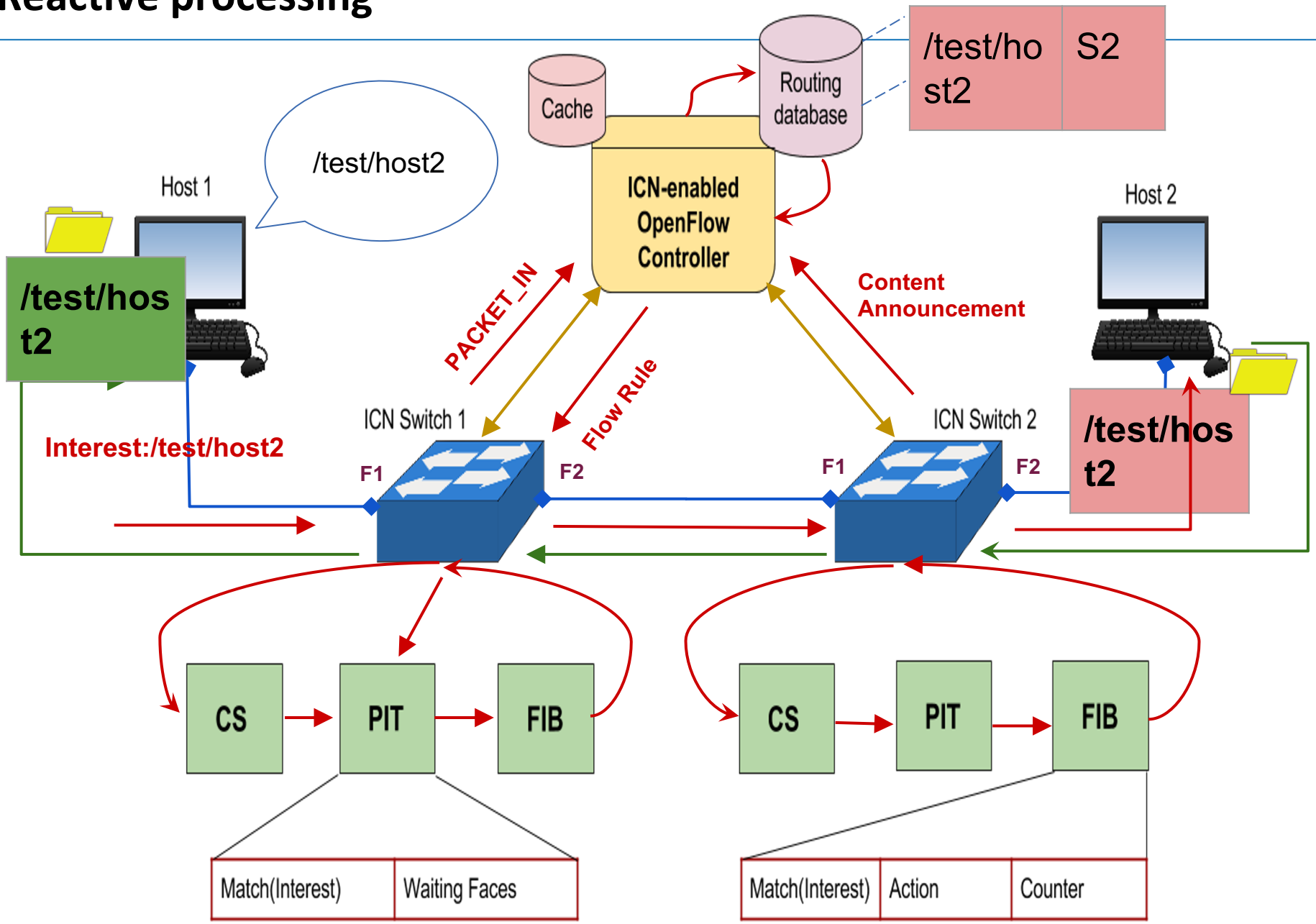
Problem Formulation - Targets

- 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

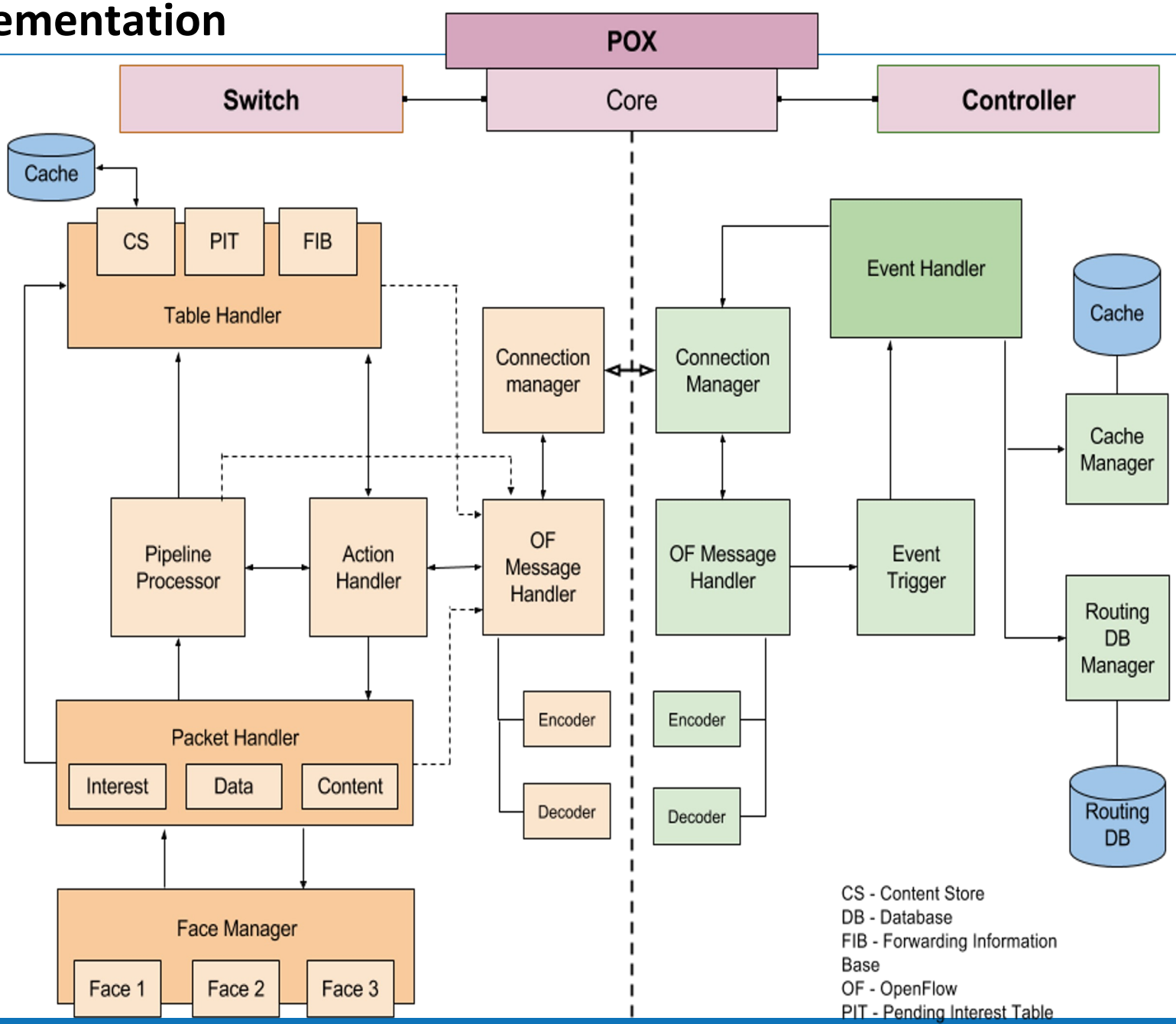
OF-ICN Design



Reactive processing



Implementation



Evaluation and Discussion

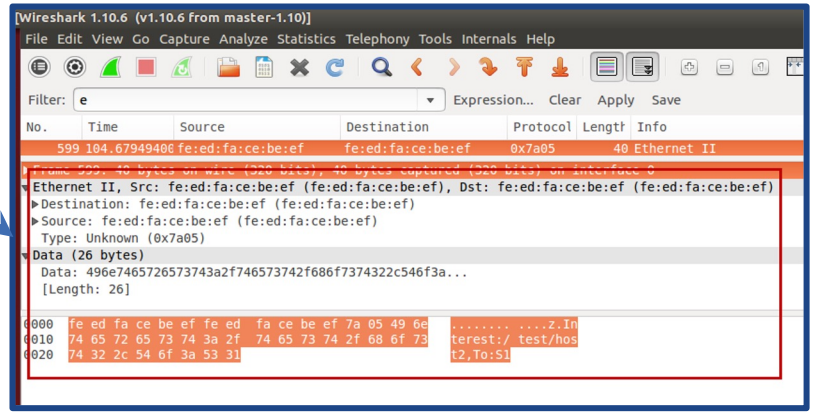
Evaluation Categories

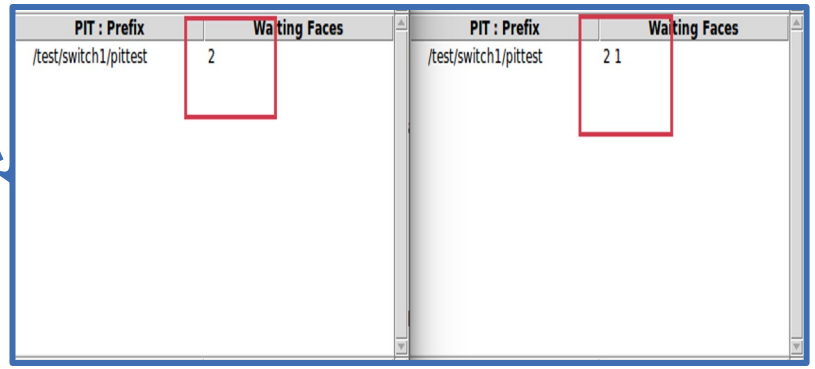
To ensure:

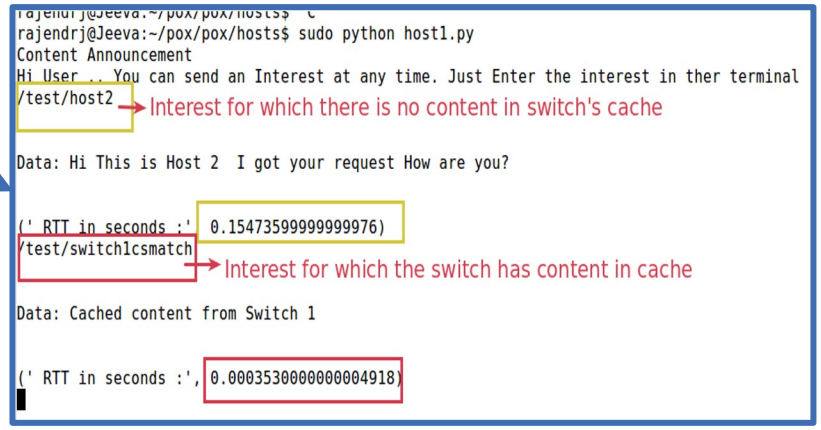
- Modified OpenFlow affords ICN functionalities
- Basic features of OpenFlow are still preserved
- Easy deployment
- Benefits of OpenFlow over ICN implementations

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

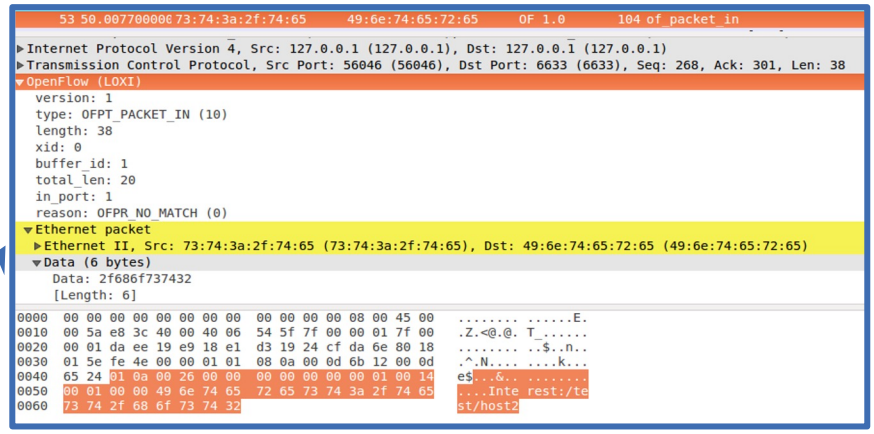


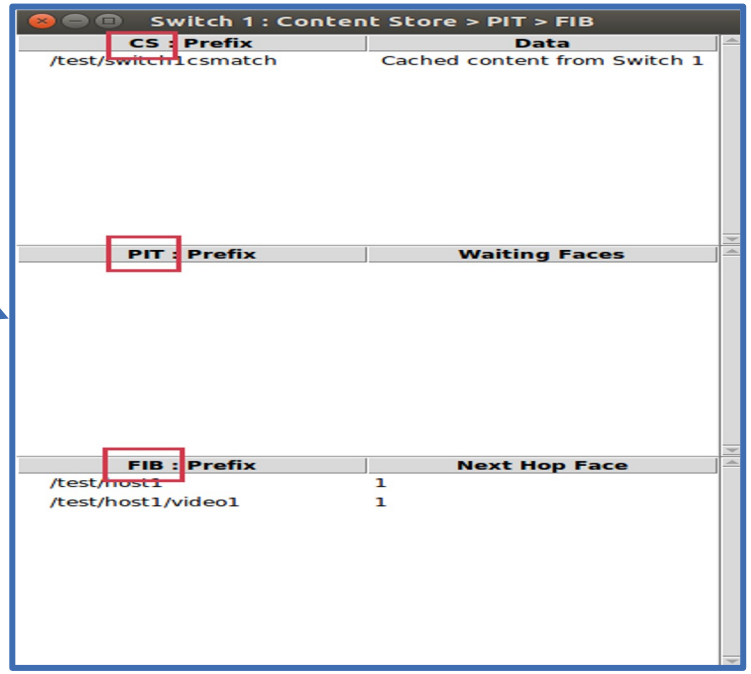




ICN Functionalities in OF-ICN

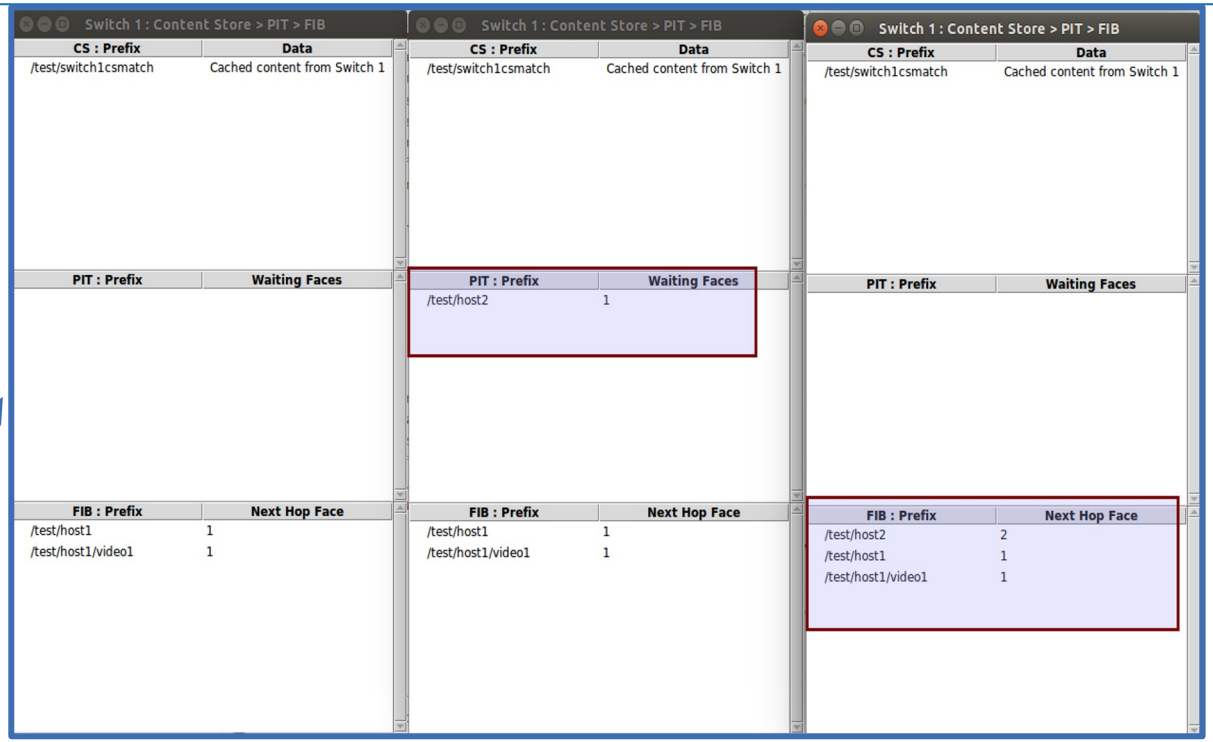
- Labeling information instead of hosts
- Requests aggregation
- Multicasting
- Scalability
- Compatibility with IP
- Flow rules
- In-network caching
- **Security**
- Flow balancing
- **Multiple tables**
- Forward-by-name
- FIB management
- Interest and data forwarding
- Caching

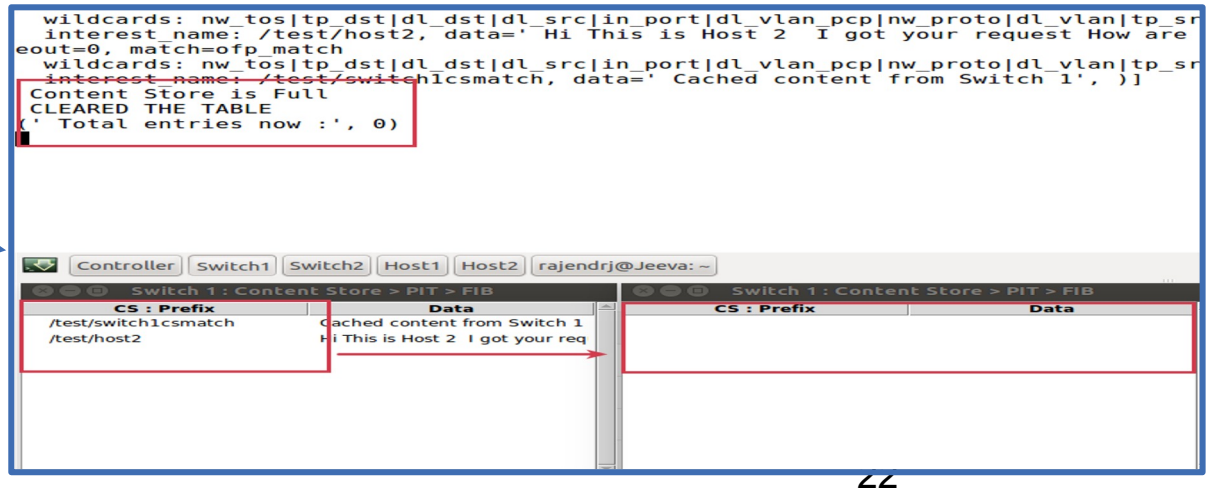




ICN Functionalities in OF-ICN

- 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**





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	OPFT_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

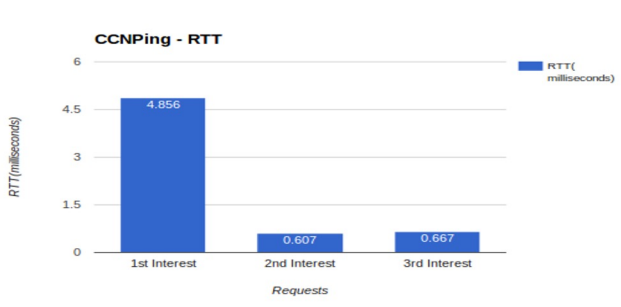
CCNPing

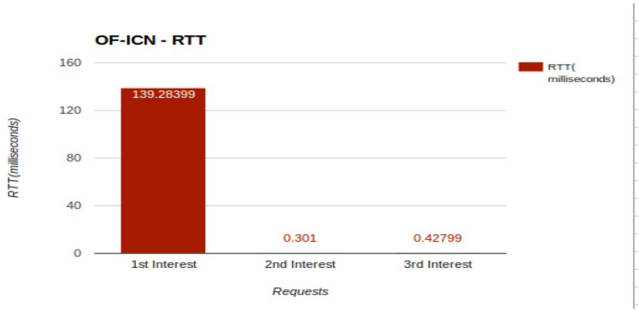
```
rajendrj@Jeeva:~/ccnping$ sudo ccnping ccnx:/test/video1
CCNPING ccnx:/test/video1
content from ccnx:/test/video1: number = 457521482
        rtt = 4.856 ms
content from ccnx:/test/video1: number = 785709862
        rtt = 0.607 ms
content from ccnx:/test/video1: number = 815833959
        rtt = 0.667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

```
rajendrj@Jeeva:~/ccnping$
rajendrj@Jeeva:~/ccnping$ sudo ccnpingserver ccnx:/test/video1
```

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 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 :', 0.00042799999999942884)
```



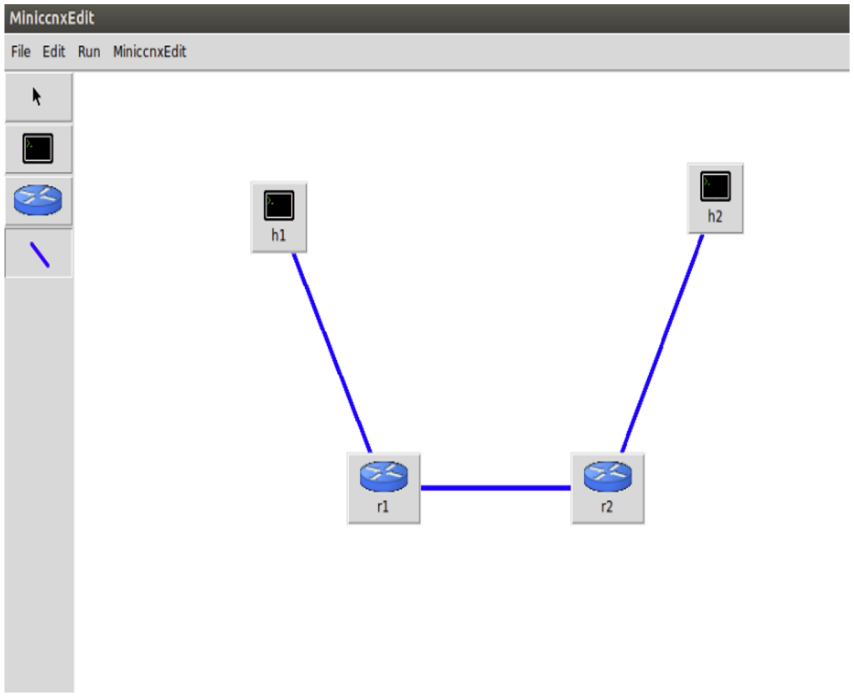


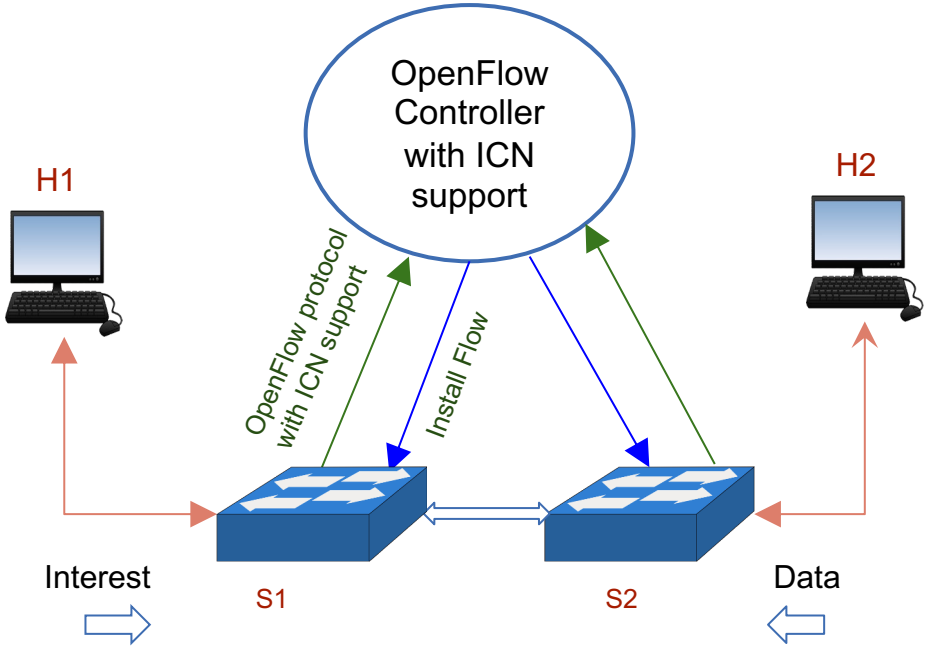
Comparison with existing ICN implementations

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

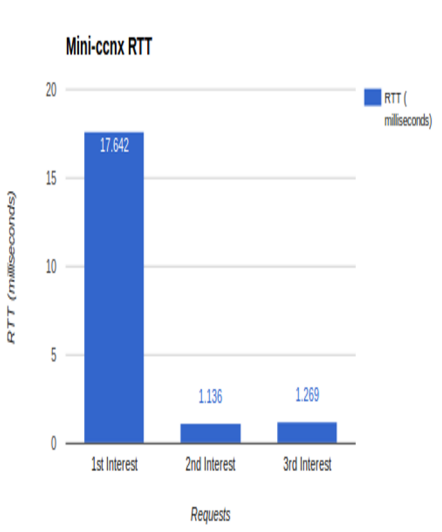
```
XTerm
rajendrj@Jeeva:~$ sudo miniccnx
Parse of miniccnx.conf done.
*** Creating network
*** Adding hosts:
h1 h2 r1 r2
*** Adding links:
(h1, r1) (h2, r2) (r1, r2)
*** Configuring hosts
h1 h2 r1 r2
*** Adding metrics collectors:
Host h1: database miniccnx_data already exists: appendir
Host h2: database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

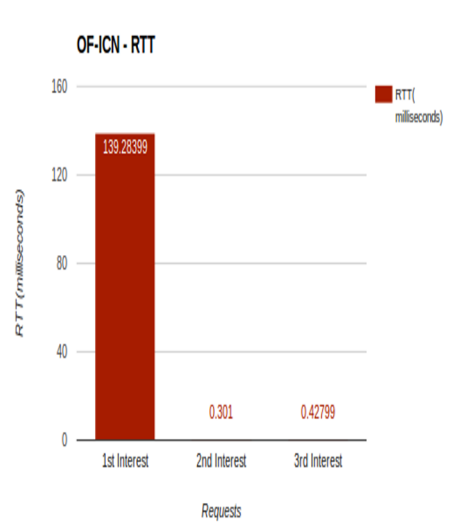
```
Node: h1
root@Jeeva:~# ccnping ccnx:/h2/video1
CCNPING ccnx:/h2/video1
content from ccnx:/h2/video1: number = 1582460534    rtt = 17.642 ms
content from ccnx:/h2/video1: number = 1700474847    rtt = 1.136 ms
content from ccnx:/h2/video1: number = 1400108873    rtt = 1.269 ms
content from ccnx:/h2/video1: number = 1689636578    rtt = 1.170 ms
content from ccnx:/h2/video1: number = 598189840     rtt = 1.107 ms
content from ccnx:/h2/video1: number = 1116217660    rtt = 1.154 ms
content from ccnx:/h2/video1: number = 516055224     rtt = 1.153 ms
^C
--- ccnx:/h2/video1 ccnping statistics ---
7 Interests transmitted, 7 Data received, 0.0% packet loss, time 6878 ms
rtt min/avg/max/mdev = 1.107/3.519/17.642/5.766 ms
root@Jeeva:~#
```

```
Node: h2
root@Jeeva:~# ccnpingserver ccnx:/h2/video1
```

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 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 :', 0.00042799999999942884)
```



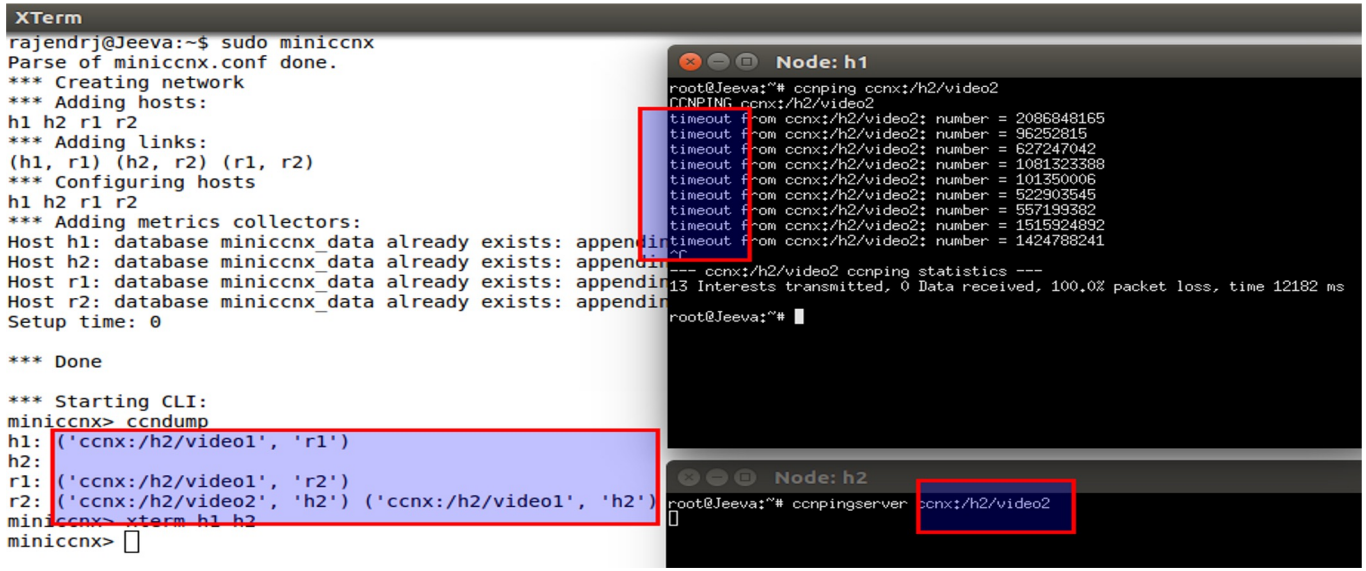


Comparison with existing ICN implementations

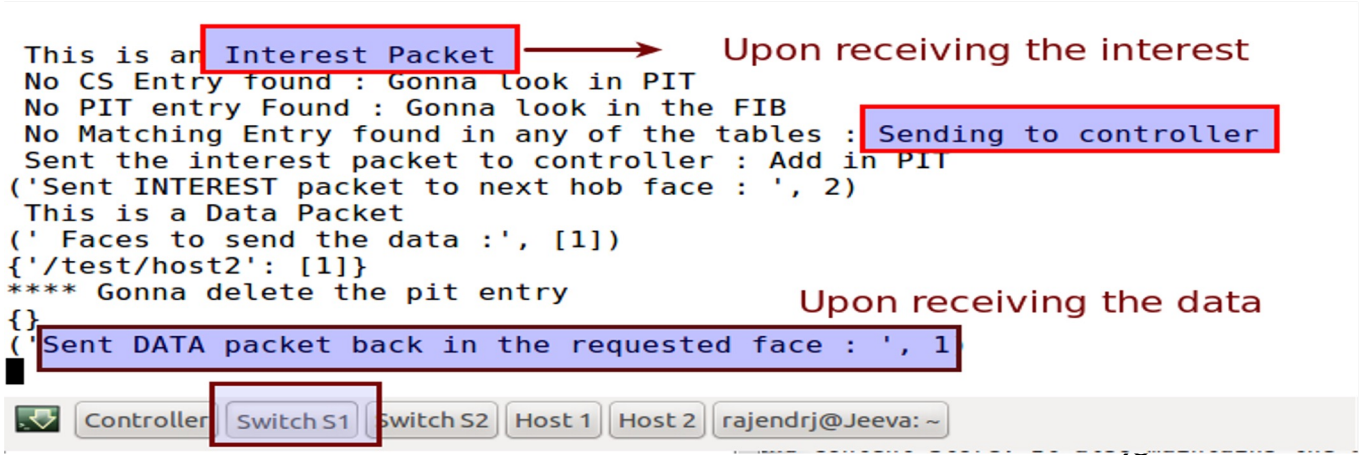
Test 2

Scenario where some routers have less forwarding details

Mini-ccnx



OF-ICN



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

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