Exploiting ICN for Flexible Management of Software-Defined Networks

Mayutan Arumaithurai[†], Jiachen Chen[†], Edo Monticelli[†], Xiaoming Fu[†] and K. K. Ramakrishnan[*]

[†] *University of Goettingen, Germany*

[*] *University of California, Riverside, U.S.A.*

Problem Statement and Terminology

Long term vision: Enhance SDN with Information Centricity to improve network management

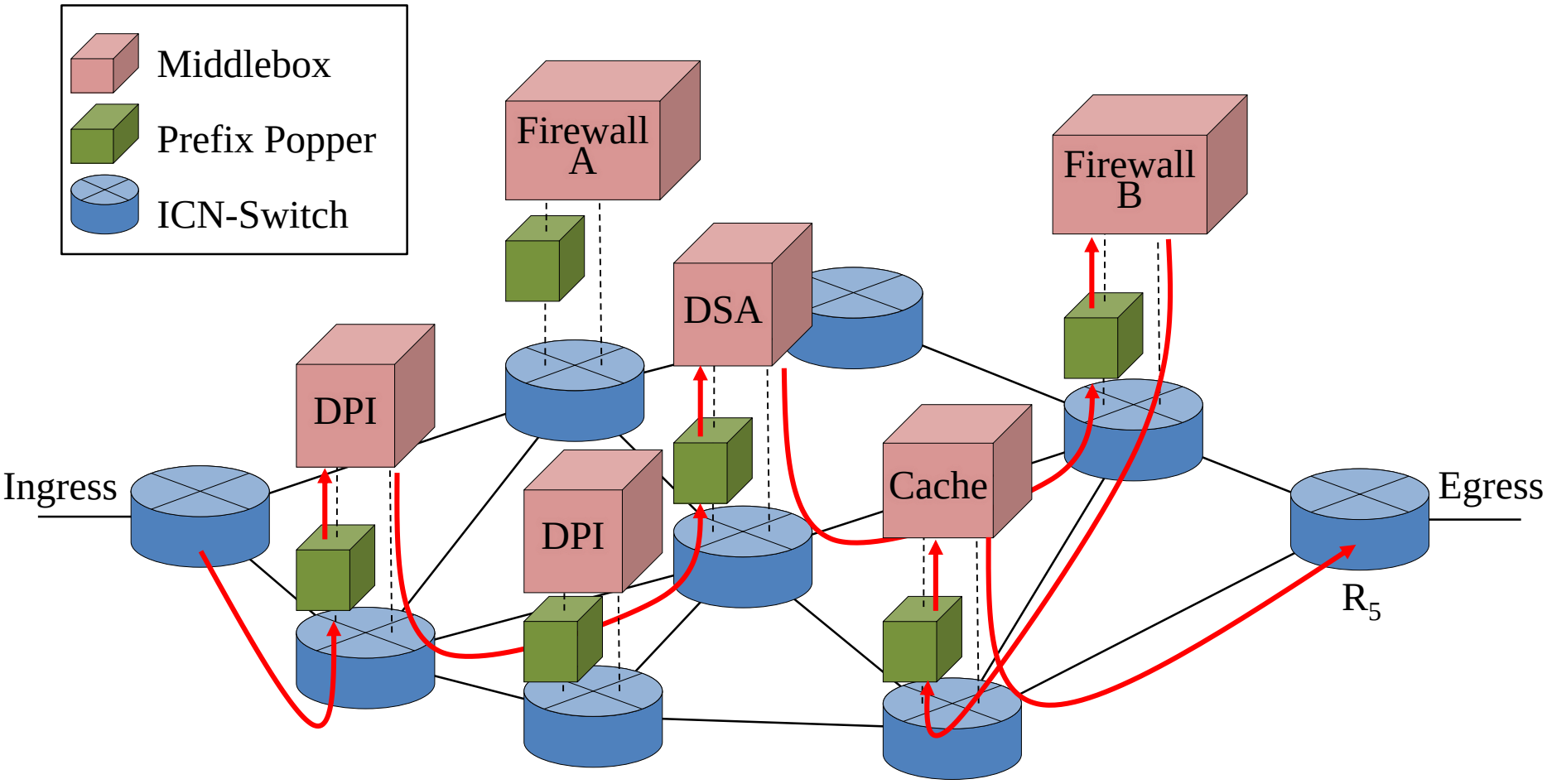
Objective of this paper: Deal with an important and common problem, i.e. Service Chaining

Service Chaining: The steering of flows through the different network functions/services needed, before it is delivered to the destination.

Middleboxes usually provide these services for reasons of

- Policy control, security, performance optimization
- They have to be resident on the path of a flow =>
  - traffic might have to deviate from its natural "IP" shortest path and forced through middleboxes

Base Topology



Motivation - I

- Middleboxes performing additional processing of packets before it is delivered has become an integral need of the Internet
  - E.g. , Facebook, Twitter, FourSquare, Google Instant, MyYahoo require content to go through middleboxes in order to improve performance
  - Dynamic site accelerators
  - TCP optimization
  - NAT
  - Proxy
  - Firewall
  - DPI
  - CDNs

NFV based Middleboxes

- Advent of Network Function Virtualization (NFV)
  - makes it easier to deploy middleboxes in a flexible and dynamic manner
  - can be instantiated, removed and positioned on demand

Managing such an NFV based middlebox provisioning and service chaining can be challenging

SDN - I

- Software Defined Networking (SDN's)
  - attempt to perform such network management by making use of a logically centralized controller
    - Controller has a global view of the network and therefore network mgmt. is easier than in the case of the current distributed approach
  - Setup flow based forwarding rules on paths
  - Provides greater control for networks to steer packets flexibly without being constrained by traditional routing such as OSPF, BGP

However, the management logic ("what" is required) is intrinsically coupled with the node location ("where" it is available).

SDN - II

- Intrinsic coupling results in issues pertaining to =>
  - Flexibility: cannot make real time decisions since location is decided
  - Scalability: places flow based forwarding rules on every router
  - Reliability: when middleboxes fail, a new path has to be built
    - or a backup path for every flow and every possibility (i.e. if router A fails, if router B fails and so on) exist

However, the management logic ("what" is required) is intrinsically coupled with the node location ("where" it is available).

Proposal

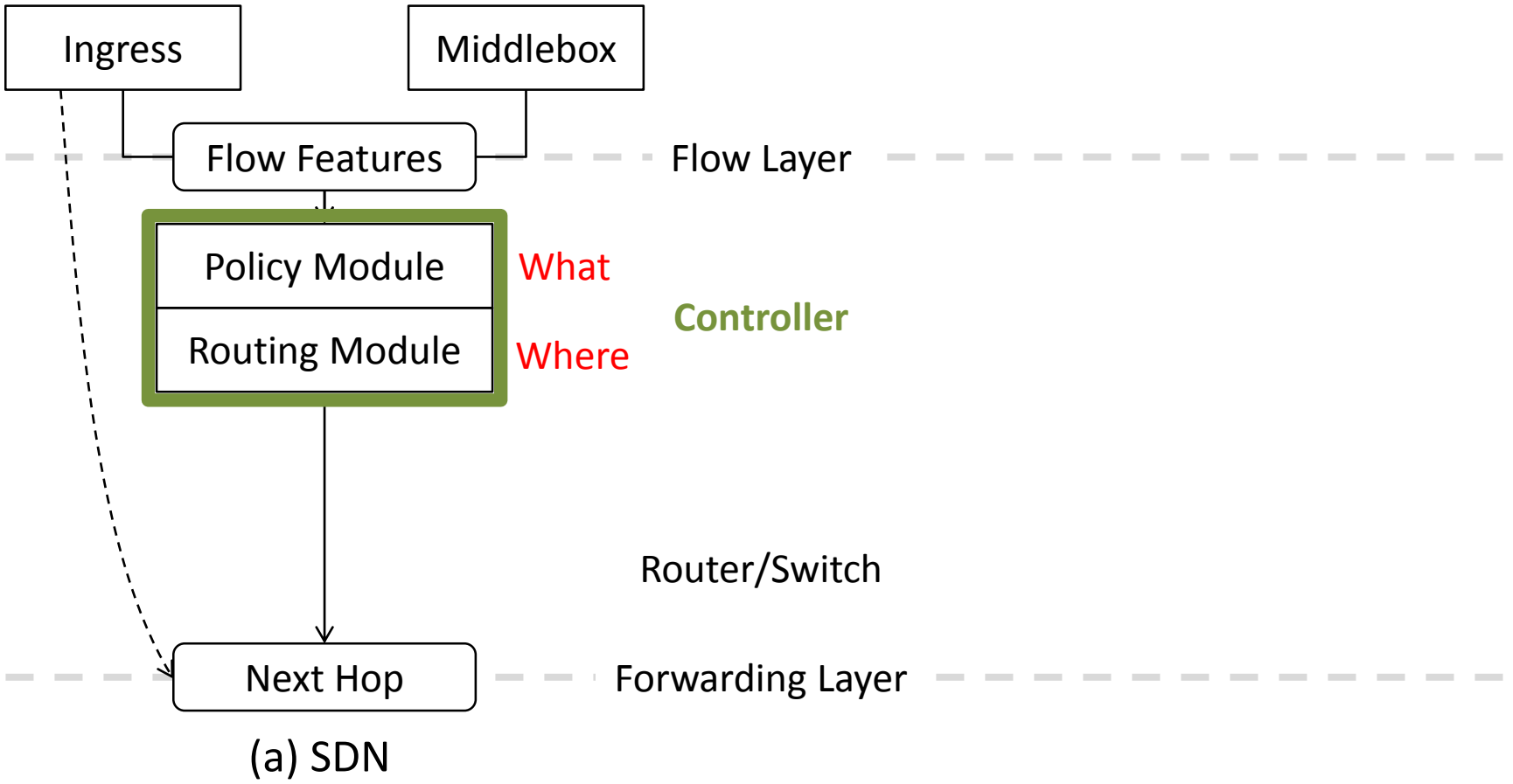
We Propose Function Centric Service Chaining (FCSC)

We argue that the performance of SDN can be further improved by using ideas of information centricity

- Decouple location from the identity (name) of the function
- Can make better use of
  - Flexibility offered by NFV

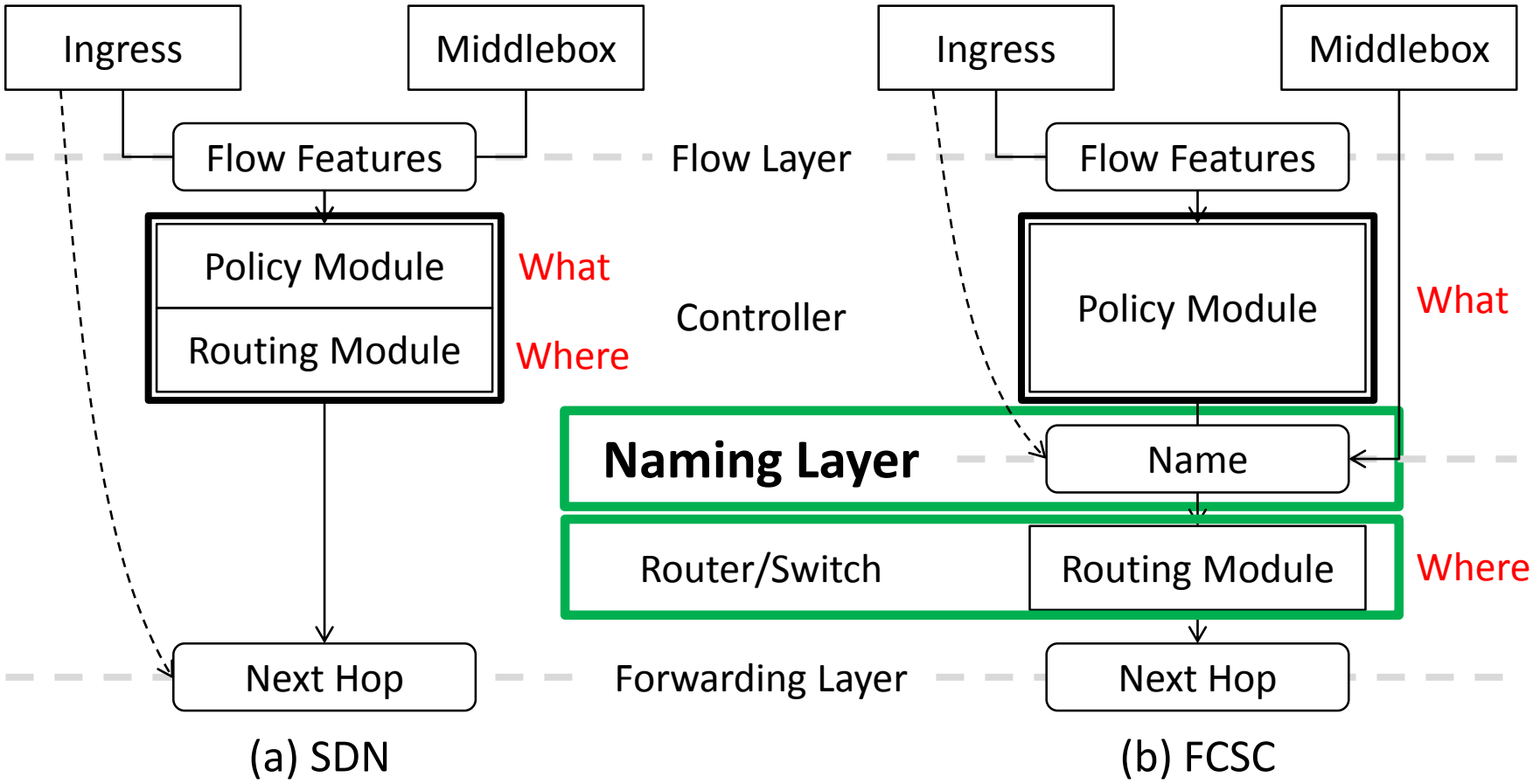
Key idea of information centricity =>
Name based forwarding
Caching is a service that it facilitates

FCSC Basic Design



(a) SDN

FCSC Basic Design



These layers are already available in NDN/CCN enabled nodes

FCSC: Forwarding Engine

Per flow solutions

Flow identifier	FACE
Flow-id	
5 Tuples	
Application	

Entries ∝ flows

FCSC

Function identifier	FACE
Function A	
Function B	

Entries ∝ functions

Scalability

FCSC: Forwarding Engine

Per flow solutions

Flow identifier	FACE
Flow-id	
5 Tuples	
Application	

FCSC

Function identifier	FACE
Function A	X
Function B	
Function A	Y

Entries ∝ flows

Entries ∝ functions — Scalability

Intrinsically supports the presence of multiple instances for the same functionality and can perform network layer load balancing among these nodes

Scalability — Flexibility — Reliability

FCSC Design Strategy

- Naming Strategy
  - **chain: DPI/cache/egress-R5**
    - Scheme identifiers could be chain, monitor, ctrl
  - Prefix popping (simple and stateless task) will be done at the node hosting that particular network function instance
- Routing Strategy
  - Middleboxes advertise the prefix they are serving
    - prefix (similar to advertising the data they serve)
  - Forwarding table is accordingly populated
    - Can use centralized/distributed routing schemes
- Stateful Middleboxes
  - In some cases, it is necessary to maintain states
  - Use **chain:firewall/_A/cache** instead of **firewall/cache**

FCSC Architecture Description

- Flow Initiation
  - Ingress (or via the controller) knows the list of functions a flow needs
  - The controller does not have to inform all the routers of the forwarding rules for the new flow, just ingress(es) is sufficient

- Proactive Rules
  - Can set at ingress the set of functions a flow might require
    - Not required to proactively set forwarding states in routers
    - No need to set paths from each ingress for all flows [O(N^2)]

Scalability

Flow identifier	Functions Needed
Flow-1	DPI, Cache
Flow-2	DSA, TCP Opt.
Flow-3	

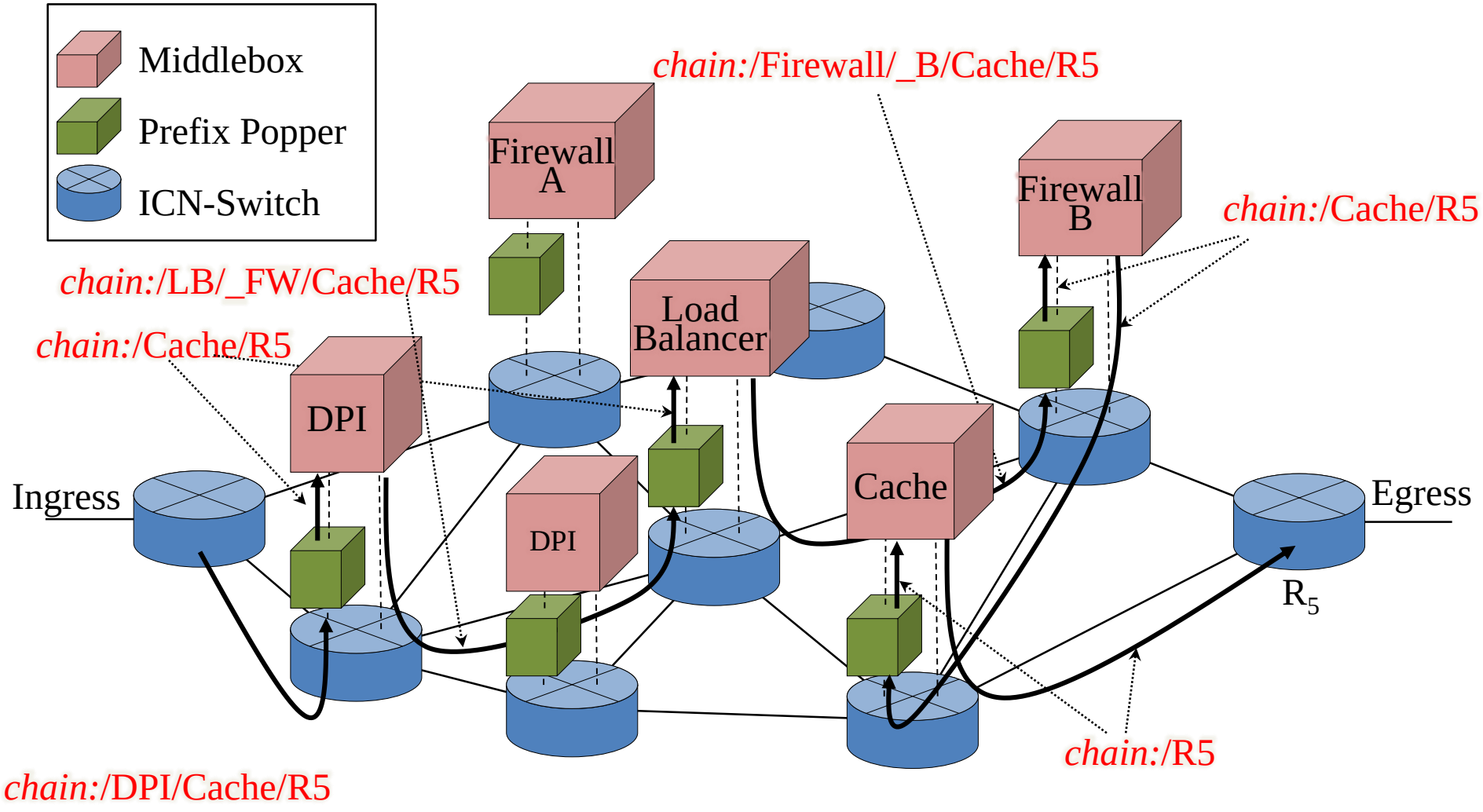
FCSC Architecture Description

- Policy change by middleboxes
  - Just change the name, i.e function list
  - No need to change forwarding rules in the routers

Flexibility

Scalability

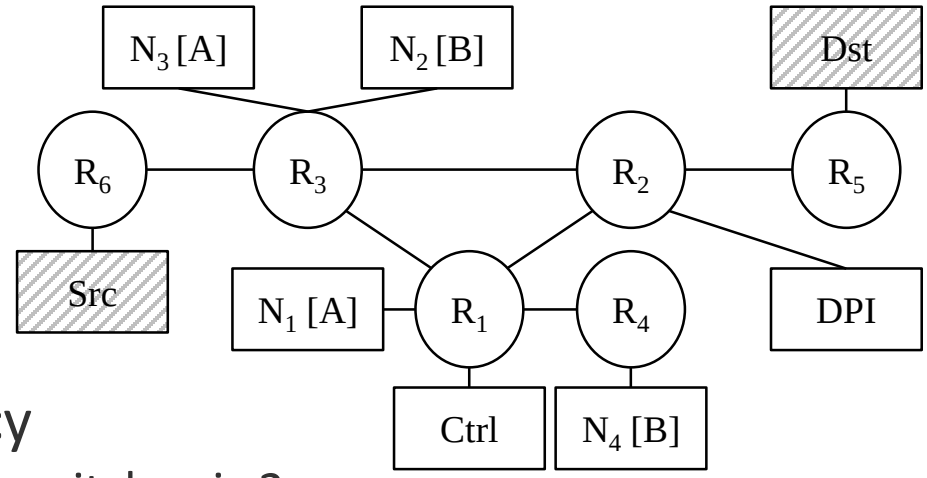
Lifetime of a Packet



Evaluations

- We use a custom simulator in Java
  - Used in previous works such as COPSS[1], G- COPSS[2], Coexist[3]
- Compare to a centralized controller based SDN
  - Decentralized approaches exist
    - Inconsistent state can affect performance
    - Communication overhead to keep them in sync is not known
  - Even if a decentralized approach exist, each controller will be responsible for a set of routers (a portion of the network)
- Topology (Demonstrate the benefits of FCSC)
  - Synthetic topology
  - Real world topology

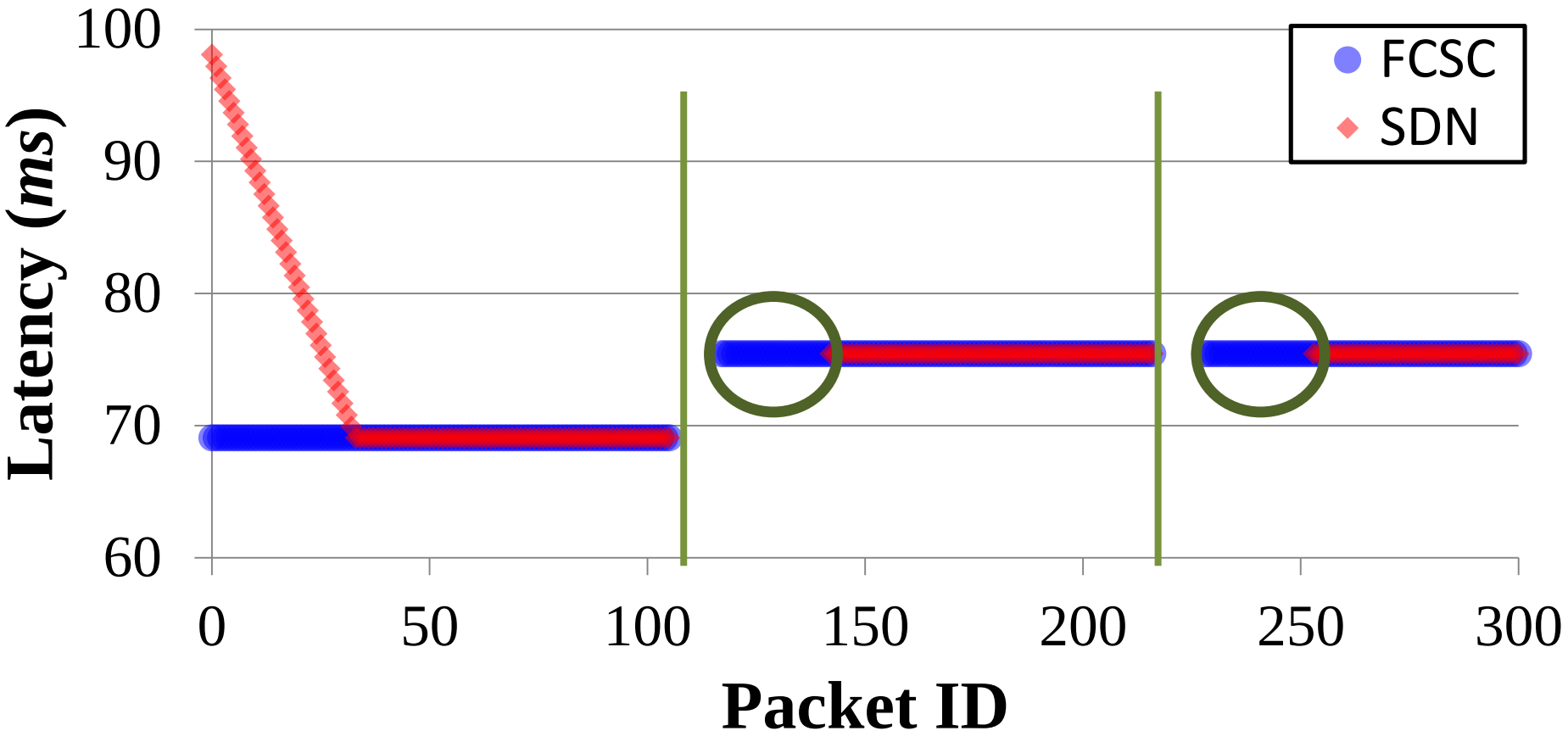
Synthetic Topology



- Link latency
  - between switches is 2ms
  - between switches and the end-systems (middlebox, src, dst, control) is 10ms.
- Bandwidth
  - 100Mbps
- The processing latency on all the middleboxes (including Ctrl) is 1ms, or 1000pps (packets per second).
- The sending rate at src is also 1000pps.

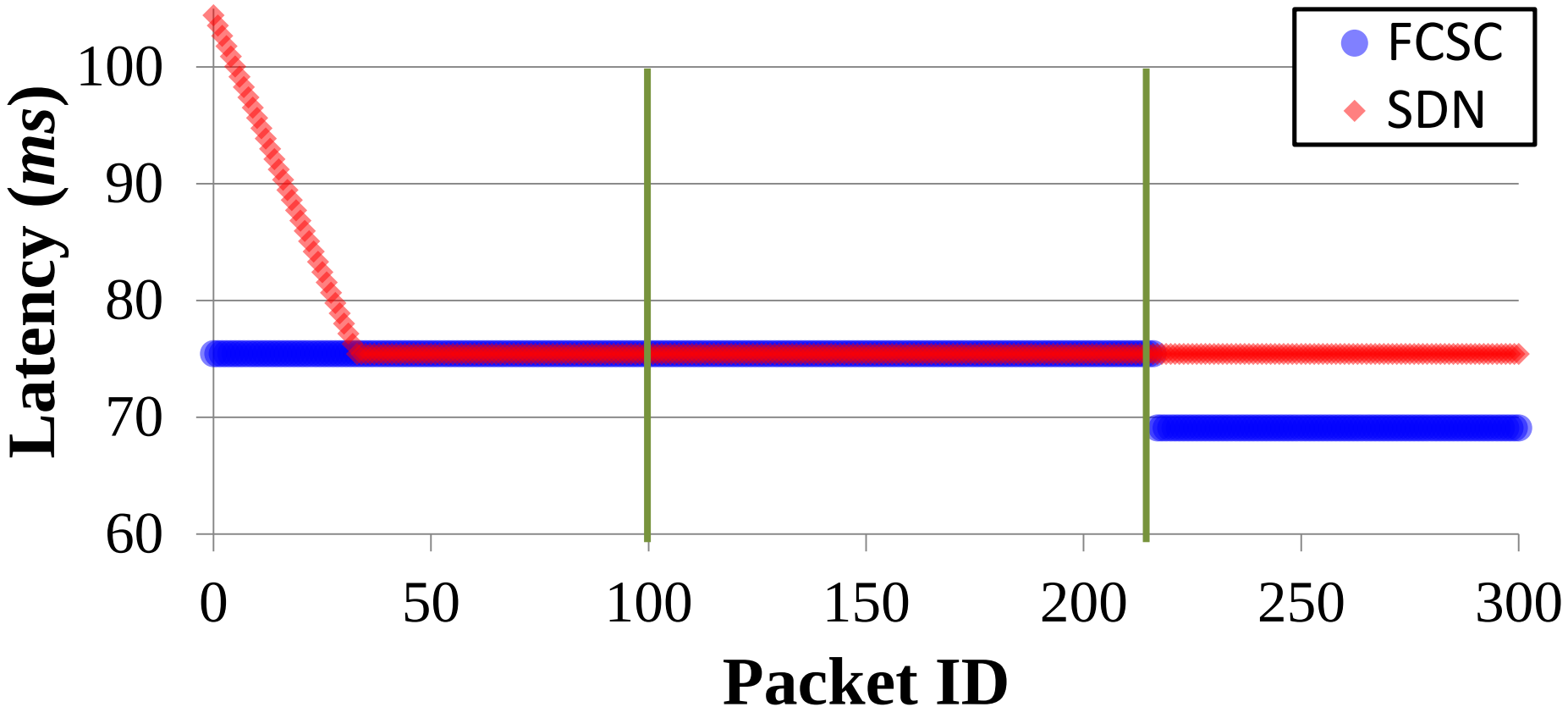
Focus: To understand how FCSC can compliment SDN

Dynamic Failure Recovery



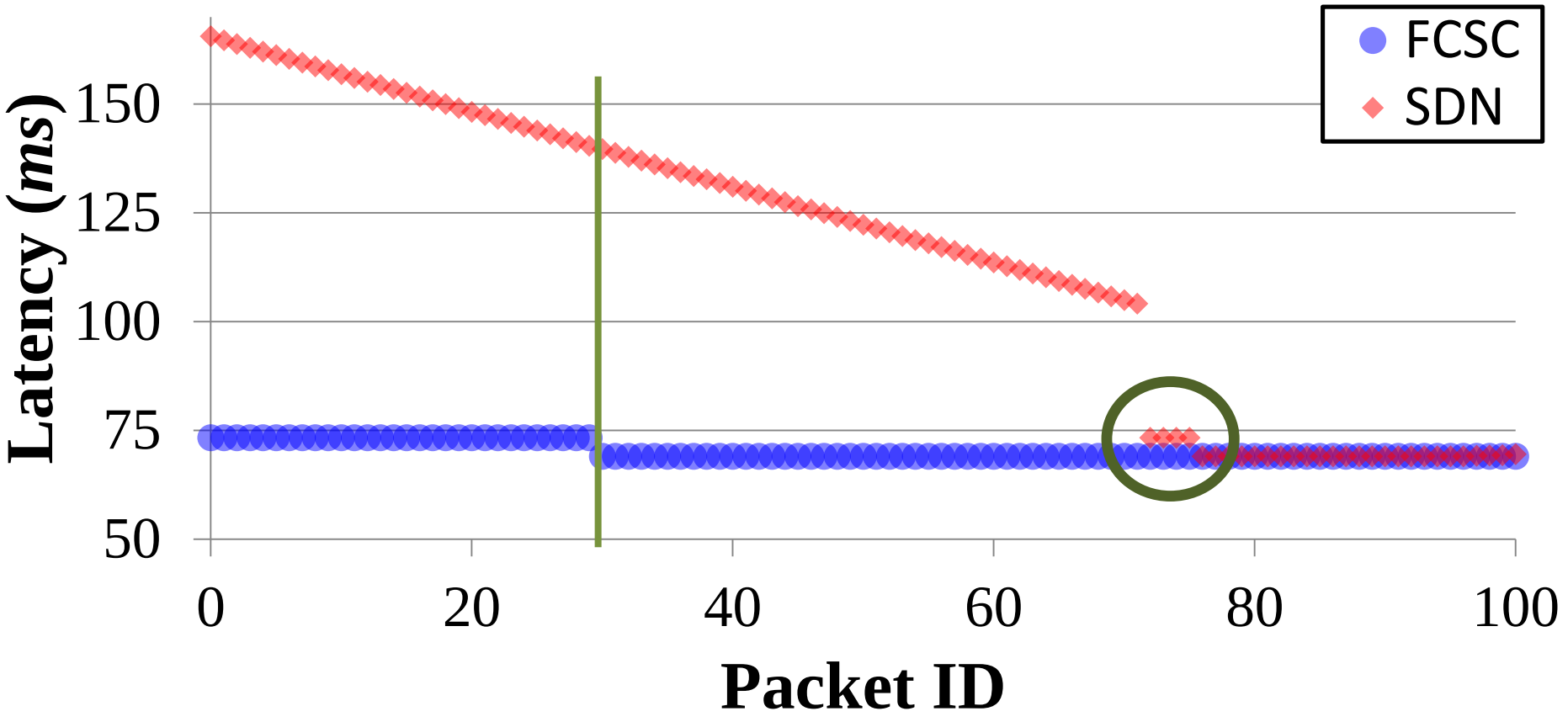
FCSC looses less packets while recovering from failure

Dynamic Adaption to New Instances



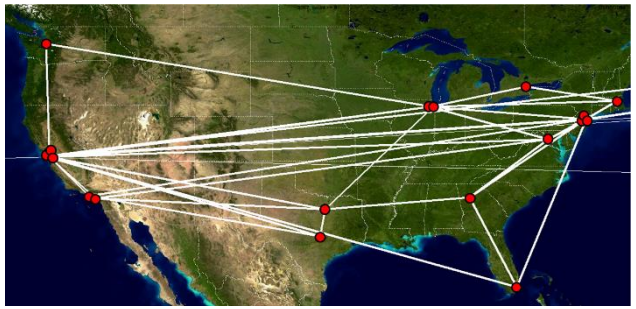
FCSC flows able to use new instances since routers are able to make forwarding decisions on a per packet basis

Dynamic Policy Change on Middleboxes



FCSC routes via the new function at 30th packet itself

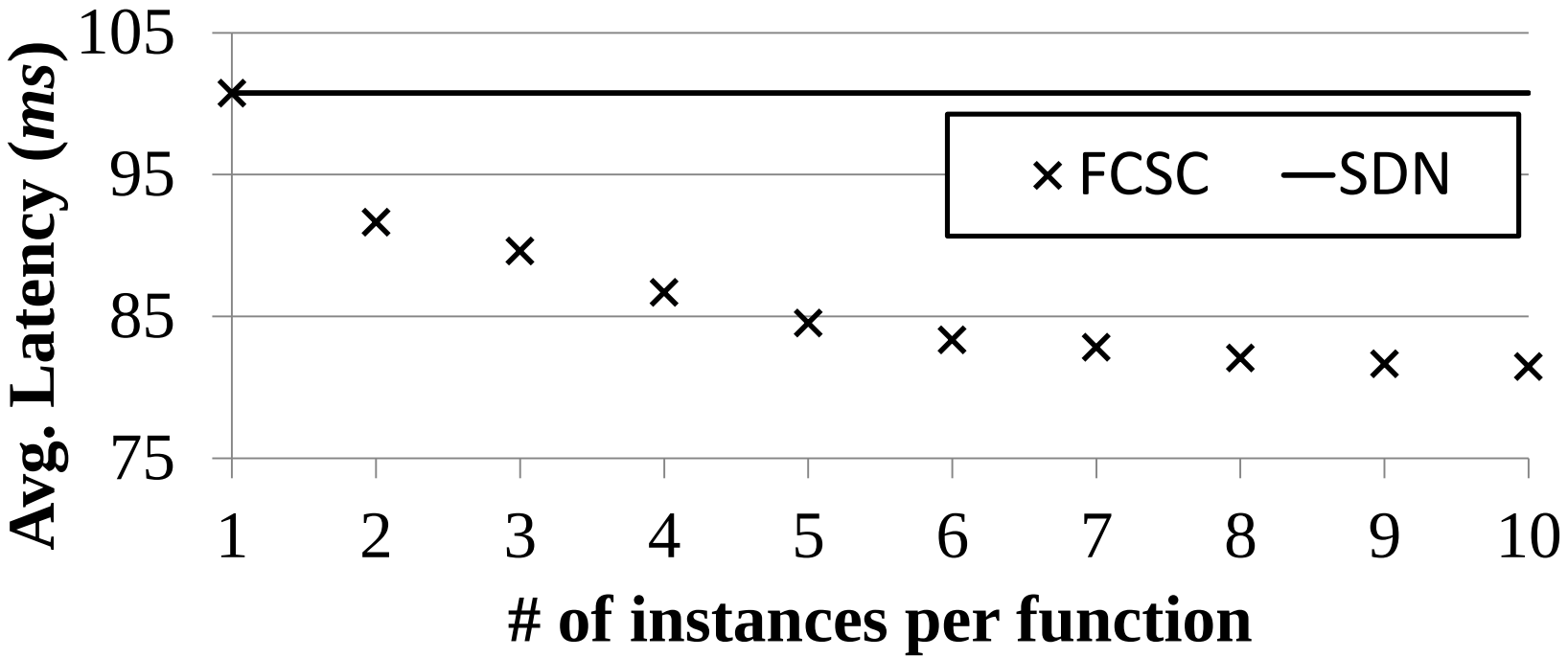
Large Scale Evaluation



- Rocketfuel Topology
  - Exodus AS-3967
- 18 cities used as core network
- Latency
  - 30 links
  - Latency ranging from 2ms – 21ms (avg 6.6ms)
  - Latency between end-hosts, controller and core routers set to 6ms
- Bandwidth is 100 Mbps
- We assume that 11 different functions are required
  - One is a DPI function that can rewrite the required function list
- Each flow belongs to one of 100 different applications
  - Each application requires a range of functions (1 to 4)

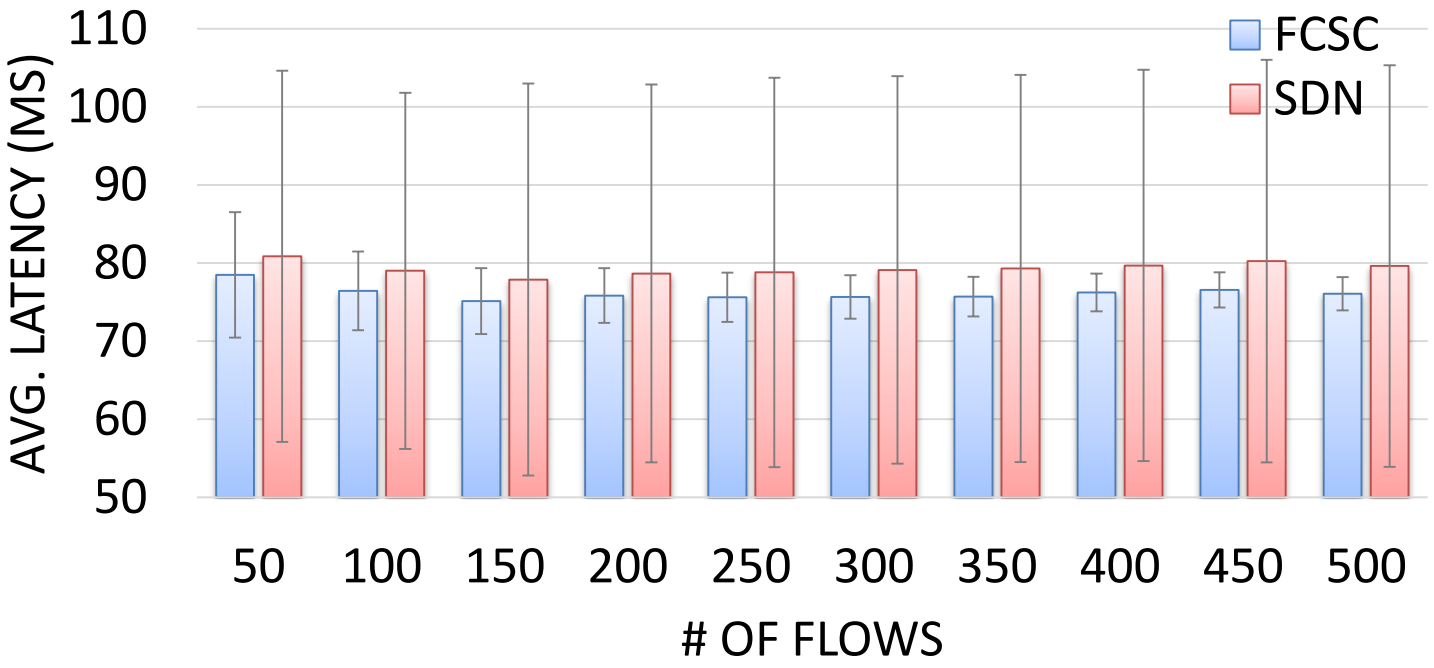
Focus: Verify if benefits of FCSC hold true in the presence of varying number of heterogenous flows and instances

Varying Number of Function Instances



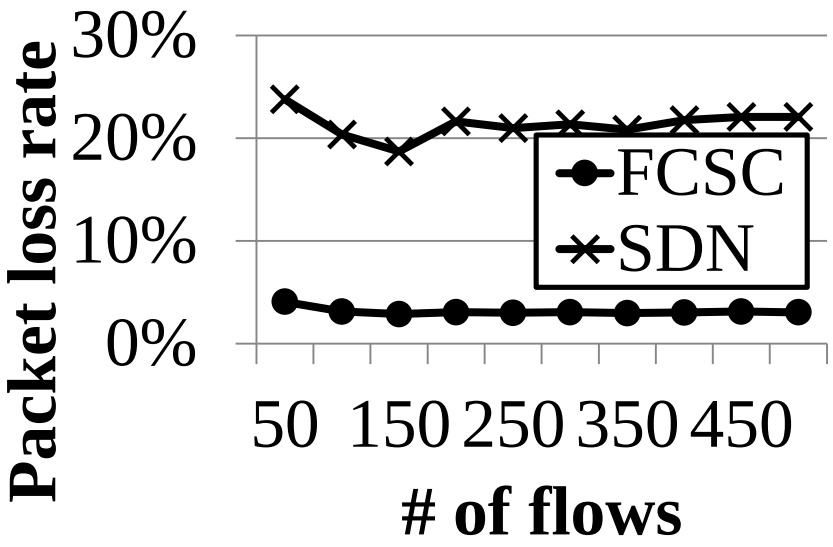
- 100 long lasting flows (5 min)
  - All flows start at 0s
  - Different sending rate (120Kbps to 1.05Mbps)
- The simulation starts with one instance
  - A new instance is launched at a random location every 30s

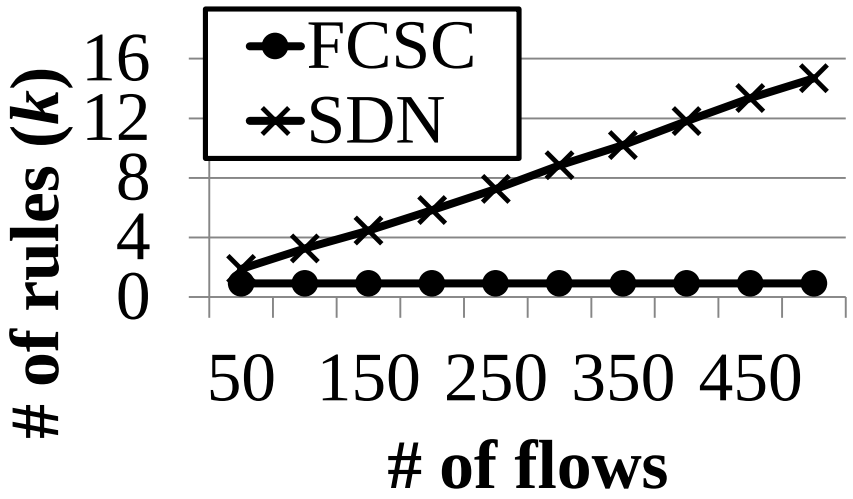
Varying Number of Flows - I



- # of flows: 50 – 500
- Arrival time: Random such that they start in the first 5 min
  - Duration: 0.05s to 91.24s
  - Sending rate: 1.2 Mbps to 11.09 Mbps.
- # of middlebox creation/failure events: 1,151
- 95% confidence Interval

Varying Number of Flows - II





- # of flows: 50 – 500
- Arrival time: Random such that they start in the first 5 min
  - Duration: 0.05s to 91.24s
  - Sending rate: 1.2 Mbps to 11.09 Mbps.
- # of middlebox creation/failure events: 1,151
- 95% confidence Interval

Conclusion

- First attempt of enhancing SDN with information centricity by dealing with a common and important problem of network mgmt.
  - Service Chaining
- Simple solution, but the potential could be huge

Future Work

As usual, open_issues/work_to_be_done outweighs work done ☺

Work on the more detailed aspects to improve naming, routing, forwarding and etc.

But, also work to convince the non-ICN community: other SDN use-cases, prototyping, deployment in small scale, better performance evaluations.

Thank you for your attention

Contact: Arumaithurai@cs.uni-goettingen.de

Acknowledgements

Parts of this research was funded by the joint EU FP7/NICT GreenICN project, under EU grant agreement 608518 and NICT contract 167 and the Volkswagen Foundation Project "Simulation Science Center".

References

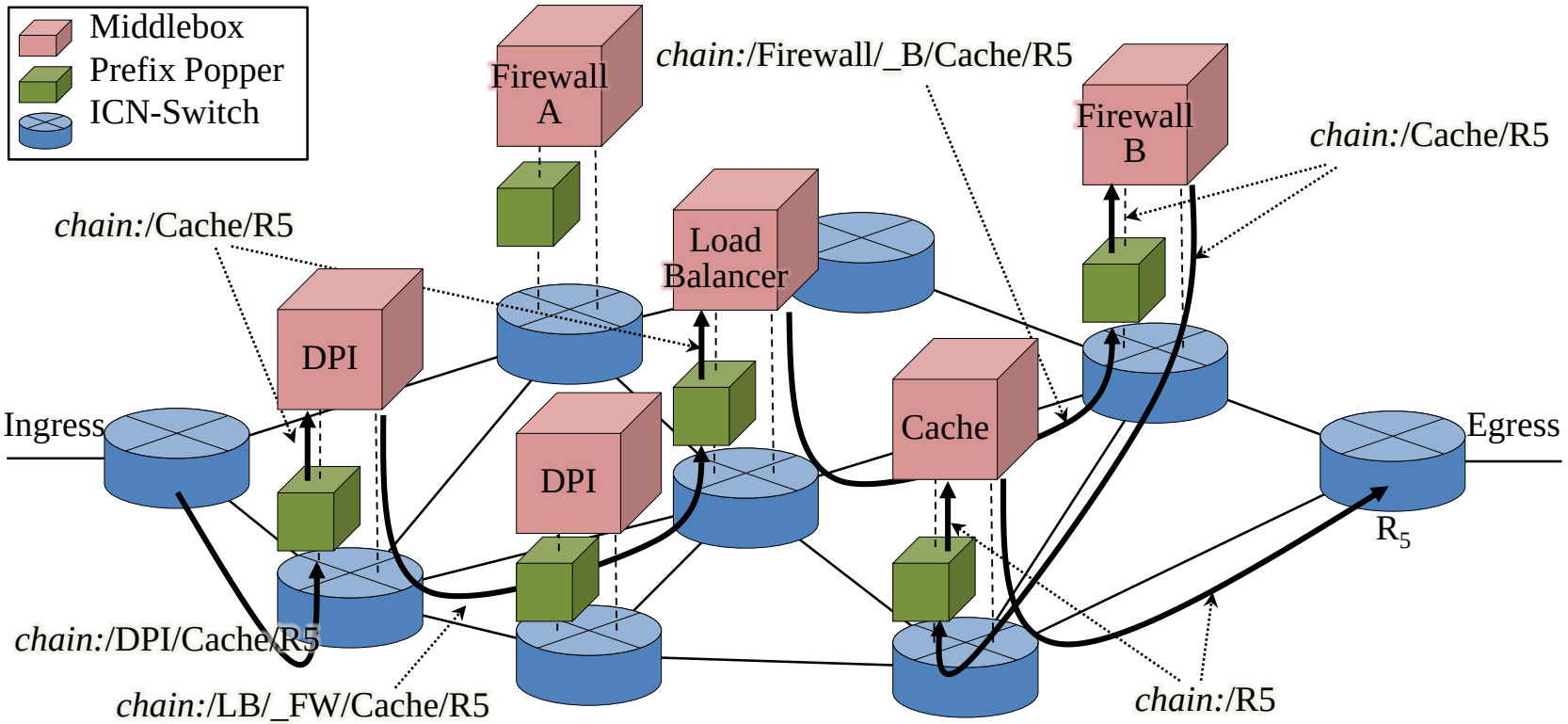
1. **COPSS: An Efficient Content Oriented Publish/Subscribe System**, Jiachen Chen, Mayutan Arumaithurai, Lei Jiao, Xiaoming Fu, K. K. Ramakrishnan, ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS 2011), Brooklyn, NY, USA, October 2011.
2. **G-COPSS: A Content Centric Communication Infrastructure for Gaming**, Jiachen Chen, Mayutan Arumaithurai, Xiaoming Fu, and K. K. Ramakrishnan, The 32nd IEEE International Conference on Distributed Computing Systems (ICDCS 2012), Macau, China, June 2012.
3. **Coexist: Integrating Content Oriented Publish/Subscribe Systems with IP**, Jiachen Chen, Mayutan Arumaithurai, Xiaoming Fu, K. K. Ramakrishnan, ACM/IEEE ANCS 2012, ACM, October 2012.

Extra Slides

Related Work - I

- Indirection based service chaining
  - They rely on predetermined nodes that provide the service
  - Con: Poor routes; Cannot make good use of the flexibility of NFVs
- Policy based routing
  - Based on traffic characteristics (5-Tuple-IP; rate)
  - Con: Policy decision needs to be communicated early to the routers, i.e., flow based forwarding rules (issues of scalability, flexibility)

Lifetime of a packet



FCSC special cases

- What if we want all the packets of a flow to go to the same instance?
  - Use name associated with that particular instance
    - E.g, chain:firewall/_A/cache
- What if we want the packets in both directions to go through the same instance?
  - Again, use name associated with that particular instance

FCSC Routing

- What is the Impact of the choice of a routing scheme?
  - Decision affects the flexibility and reliability
    - Centralized controller based scheme
      - » Global knowledge
    - Distributed routing scheme
      - » Need to synchronize better
      - » But information can propagate to neighbours quickly
  - Possibility to use a load-balancer middlebox instance if required

FCSC Advantages

- FCSC therefore
  - Flexibility
    - Enables switches to dynamically detect load and accordingly instantiate/dispose Network functions
    - Allows flows to use newly instantiated NFVs and faster recovery from node/link failures
    - Allows dynamic modification of the functions needed by a flow on the controller or middleboxes
  - Scalability
    - By placing the flow state in the packet header, the number of states stored in the network is reduced
      - Therefore more scalable than the per-flow state solutions
    - Intrinsically supports the presence of multiple instances for the same functionality and can perform network layer load balancing among these nodes
  - Reliability
    - Allows faster recovery from node/link failures by using the forwarding table to find another instance

Proactive rule for flow initiation