

FAR: A Fault-avoidance Routing Method for Data Center Networks with Regular Topology

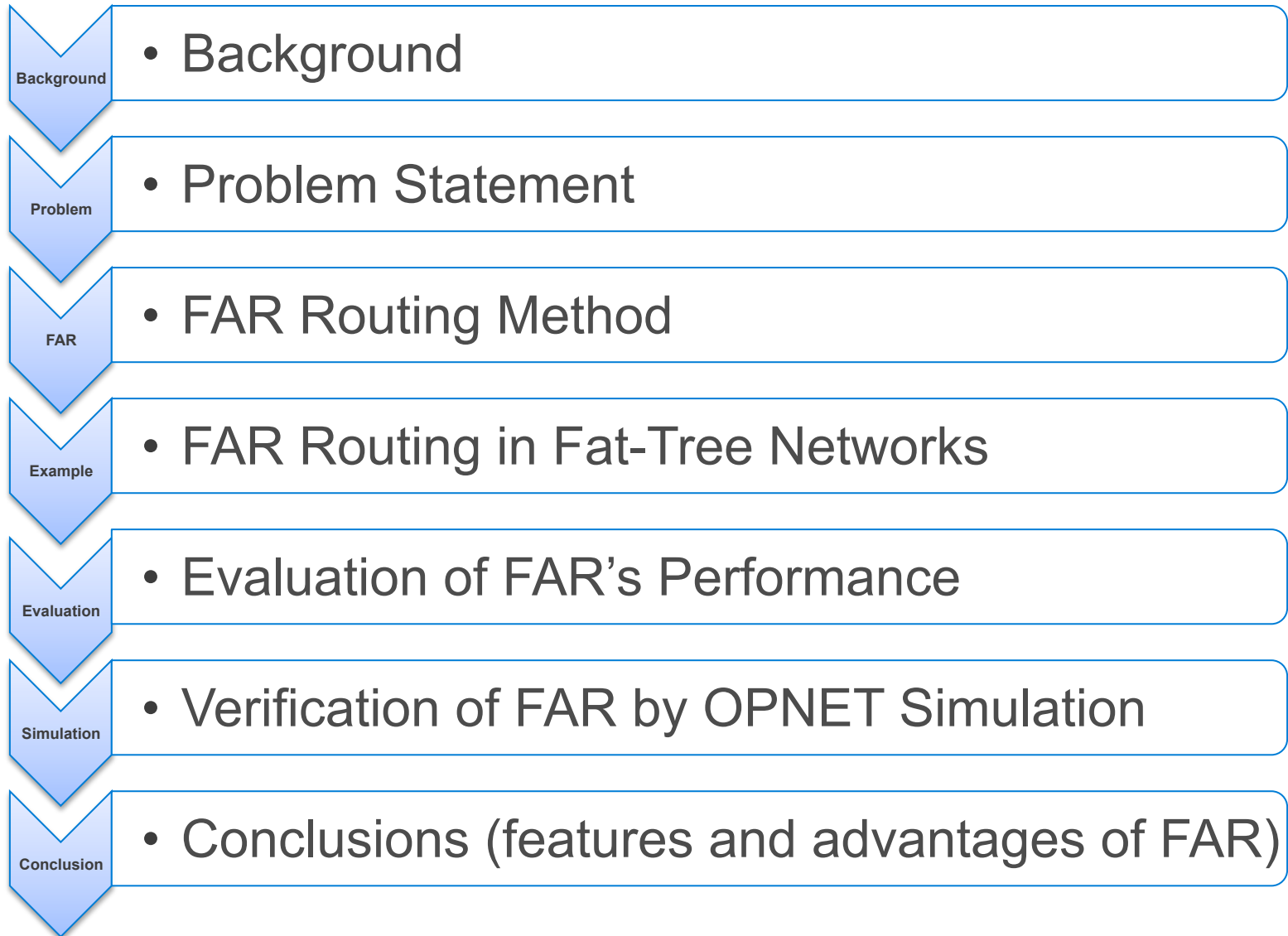
<http://datatracker.ietf.org/doc/draft-sl-rtgwg-far-dcn/>

Bin Liu, ZTE
Yantao Sun, Jing Cheng Yichen Zhang
Beijing Jiaotong University

Pease send comments to rtgwg@ietf.org

IETF89, Friday, 07 March 2014
Rm [Blenheim](#), Hilton London Metropole
London, UK W2 1JU

Outline



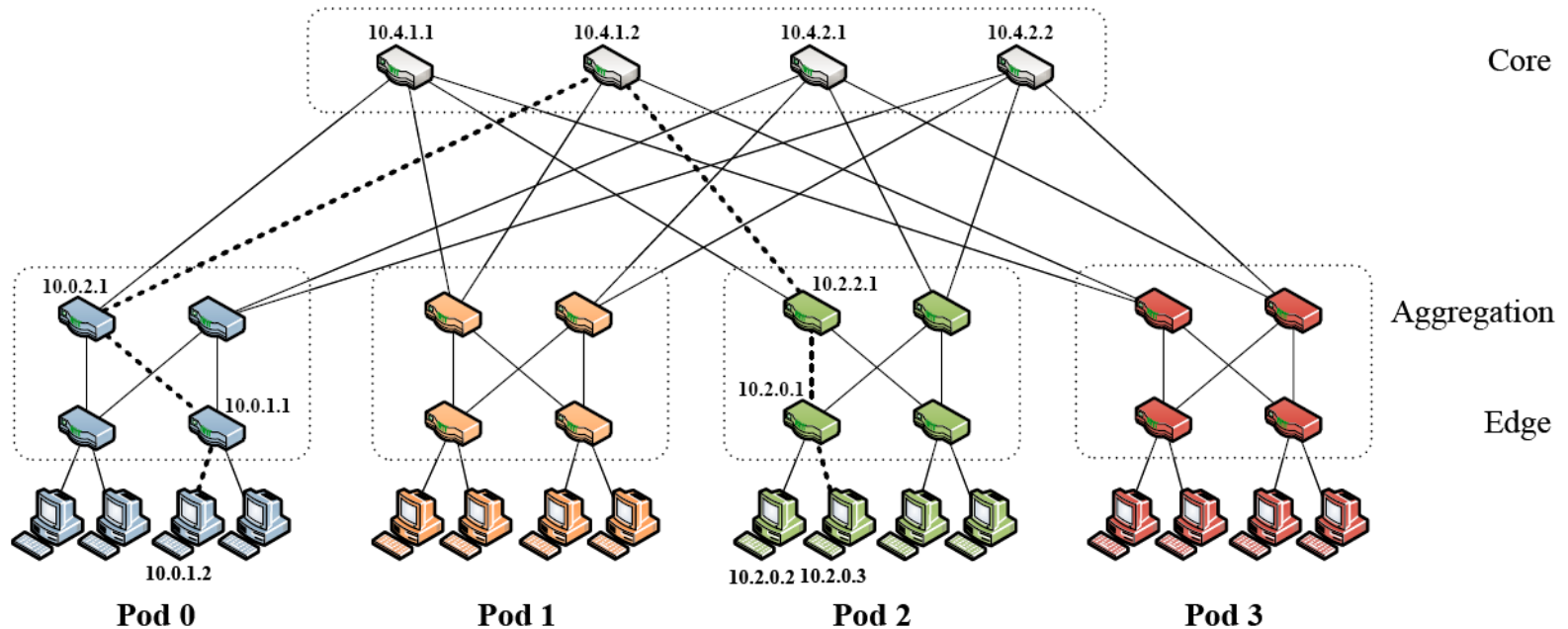
1. Background

With rapid development of cloud computing, the demand for Data Center scaling is increasing ...

As set of network architectures have been proposed to support extra-large-scale Data Centers with more than 100K servers

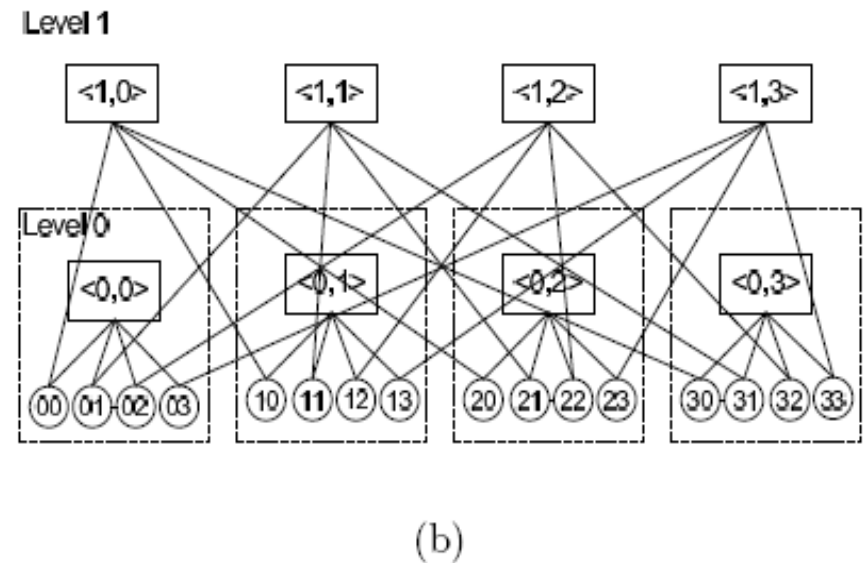
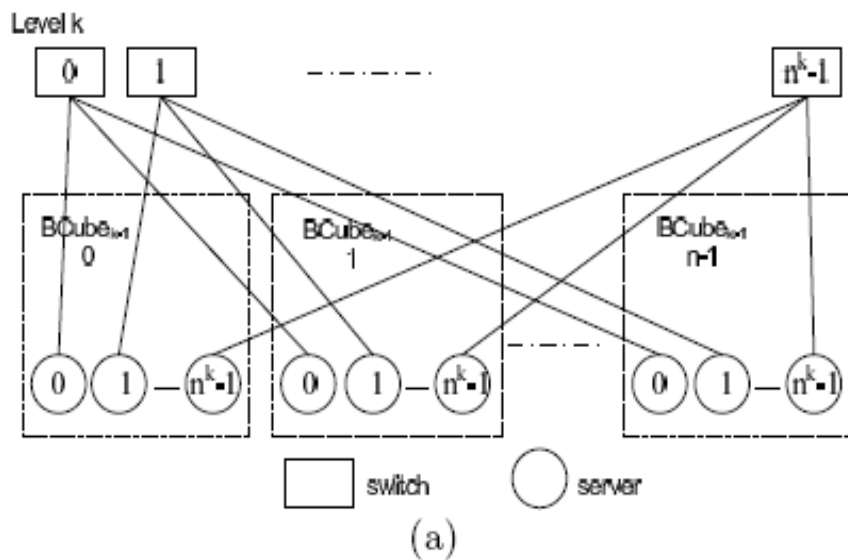
New Network architectures—Fat-tree

A Fat-tree network can support 27,648 hosts using 2,880 48-port switches.



New Network architectures—BCube

A BCube2 network can support 110,592 hosts using 6,912 48-port switches



2. Problem Statement

Large-scale data center networks and new architectures bring challenges to conventional routing methods

Challenge 1 — The Impact of Large-scale Networks on Route Calculation

- OSPF and other conventional routing methods **do not work well** in a large-scale network with several thousands of routers.
 - The time of network **convergence** would be **too long**, which will cause a longer time to elapse for creating and updating the routes.
 - a **large number** of routing protocol **packets** need to be sent, which will consume a lot of network **bandwidth** and **CPU** resources.
- In FAR, Routing tables including a **Basic Routing Table (BRT)** and a **Negative Routing Table (NRT)** are built based on local network and partial learnt link failures by leveraging the regularity of the network topologies.
 - So FAR does **not need to wait for the completion** of the network convergence in the process of building these tables.
 - FAR only needs to **exchange** a small amount of **link failure information** between routers, and consumes less network bandwidth.

Challenge 2 — Network Addressing Issues

- OSPF and other routing algorithm require each interface of a router must be configured with an IP address. Each router has dozens of network interfaces.
- Tens of thousands of IP addresses may be needed to configure for thousands routers in a DCN.
- In FAR, the device location information is encoded in the IP address of the router. Each router only needs to be assigned a unique **IP address for data plane according its location**. All controller card share one on IP address in a FAR router.

Challenge 3 — Big Routing Table Issues

- Tens of thousands route entries are required for a router in a large-scale data center network. It will increase equipment cost and reduce the querying speed of a **routing table**.
- FAR uses two measures to reduce the size of the routing tables
 - **Builds a BRT on the regularity** of the network topologies.
 - introduces a new routing table, i.e., **NRT**.
 - FAR can **reduce the size of routing tables** to only a few dozen routing entries.

Challenge 4 — Adaptability Issues for Routing Algorithms

- Besides FAR, some other routing methods are proposed for specific network architectures, such as Fat-tree and BCube. These routing methods are different (from both design and implementation viewpoints) and **not compatible** with conventional routing methods.
- **FAR** is a generic routing method. With slight modification, FAR method can be **applied** to most of **regular datacenter networks**.
- The **structure** of routing tables and **querying** a routing table in **FAR** are the **same as conventional routing methods**.

Challenge 5 — Virtual Machine Migration Issues

- Supporting VM migration is very important for a cloud datacenter. However, in order to support layer-3 routing, routing methods including OSPF and **FAR** require **limiting VM migration within a subnet**.
- To solve this paradox, one competitive method is to transmit packets by IPinIP or MACinIP **tunnels** passing through intermediate networks.

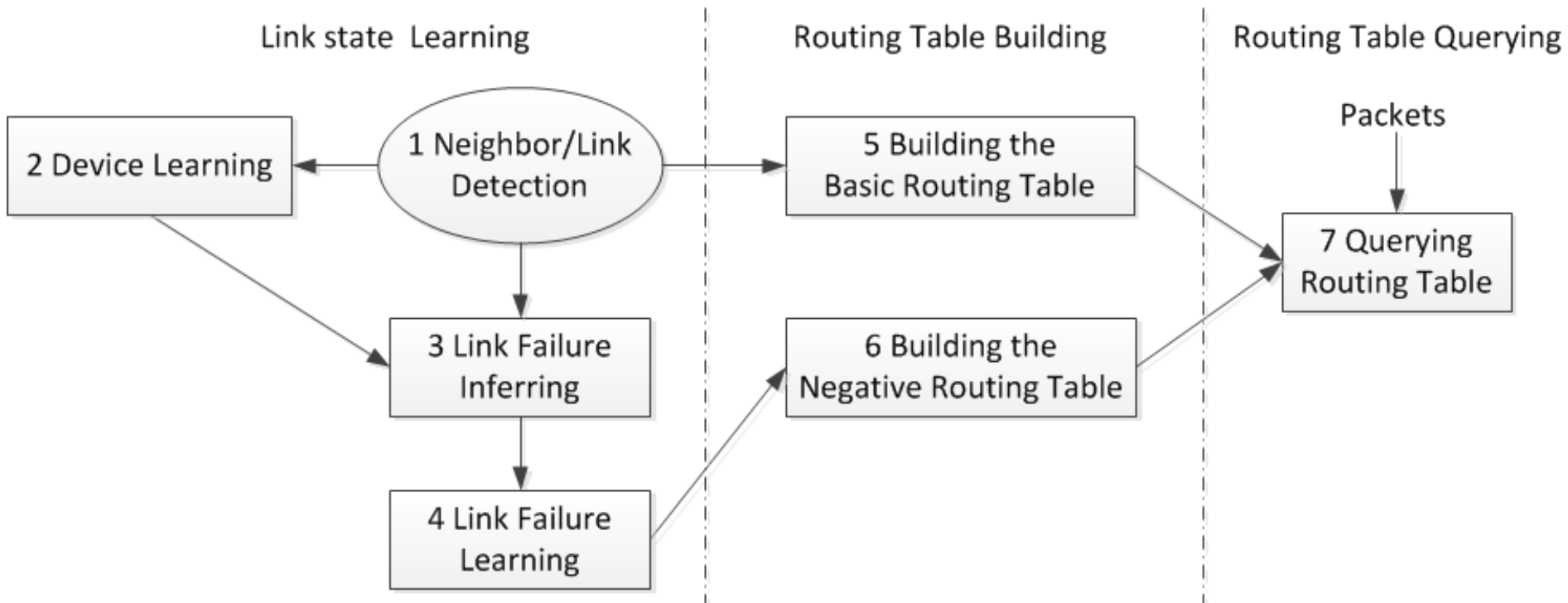
3

FAR Routing Method

The Principle of FAR

- FAR is a generic routing method designed for a data center network with regular topology. A regular topology means the structure of a network has a definite pattern, so a router in the network knows the entire network without a learning procedure.
- Network devices, including routers, switches, and servers, are assigned IP addresses according to their location in the network.
- A basic routing table(BRT) is built based on local topology.
- A negative routing table(NRT) is built based on link and device failures in the entire network.
- Look up both a BRT and a NRT to determine the final route in a routing procedure.

FAR Framework



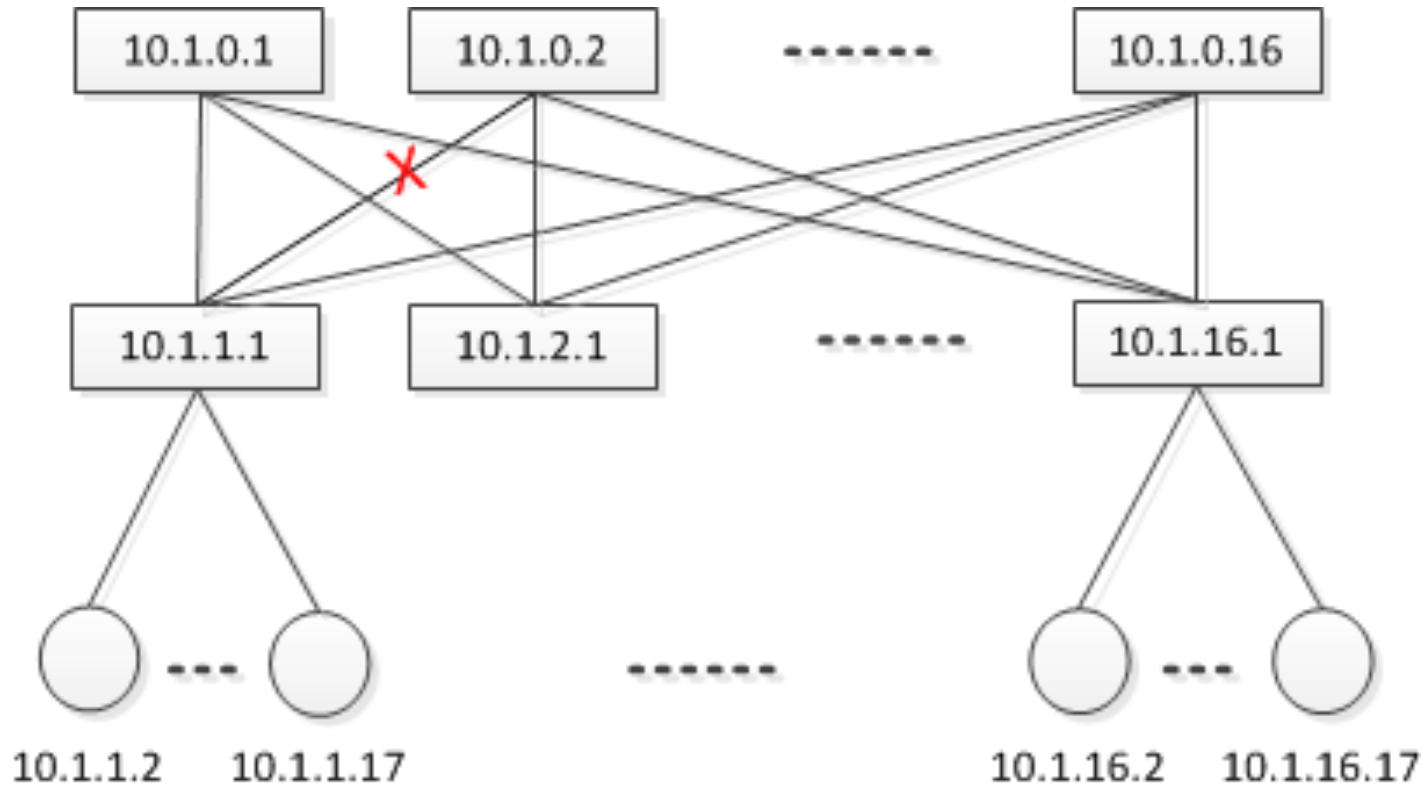
BRT (Basic Routing Table)

- A BRT performs like conventional routing tables.
- A BRT is stable and small. It almost doesn't change during a router's lifetime and contains only dozen of route entries.
- To build a BRT, a router only need to learn its neighbor routers by a heartbeat (every 100 ms) mechanism
- A router leverages the regularity in network topology when it builds its BRT
- Longest prefix match is applied in looking up a BRT entry

NRT (Negative Routing Table)

- An NRT is used to avoid failed links
- An NRT changes according to the change of links failures
- NRT is also very small. It contains several to hundreds of route entries varied according to the number of links failures
- Routers exchange information to learn the knowledge of link failures of the entire network
- Opposite to a BRT, if a route entry in an NRT is matched, the relevant next-hop should be avoided
- All the matched route entry are valid and their relevant next-hops should be avoided when looking up an NRT

NRT can decrease the size of a routing table remarkably in a multiple path networks



Contd../

If no failure, the routing table of node 10.1.16.1 has 16 entries.

<u>Destination/Mask</u>	<u>Next Hop</u>
10.1.0.0/255.255.0.0	10.1.0.1
10.1.0.0/255.255.0.0	10.1.0.2
...	
10.1.0.0/255.255.0.0	10.1.0.16

Contd../

- If the link between node 10.1.1.1 and 10.1.0.2 fails, 15 additional route entries should be added in conventional routing methods.

<u>Destination/Mask</u>	<u>Next Hop</u>
10.1.1.0/255.255.255.0	10.1.0.1
10.1.1.0/255.255.255.0	10.1.0.3
...	
10.1.1.0/255.255.255.0	10.1.0.16

- In FAR, only one route entries is added to a NRT.

<u>Destination/Mask</u>	<u>Next Hop</u>
10.1.1.0/255.255.255.0	10.1.0.2

Routing Procedure in FAR


1. Look up a BRT to obtain candidate next-hops



2. Look up a NRT to obtain avoiding next-hops



3. candidate next-hops - avoiding next-hops
= applicable next-hops



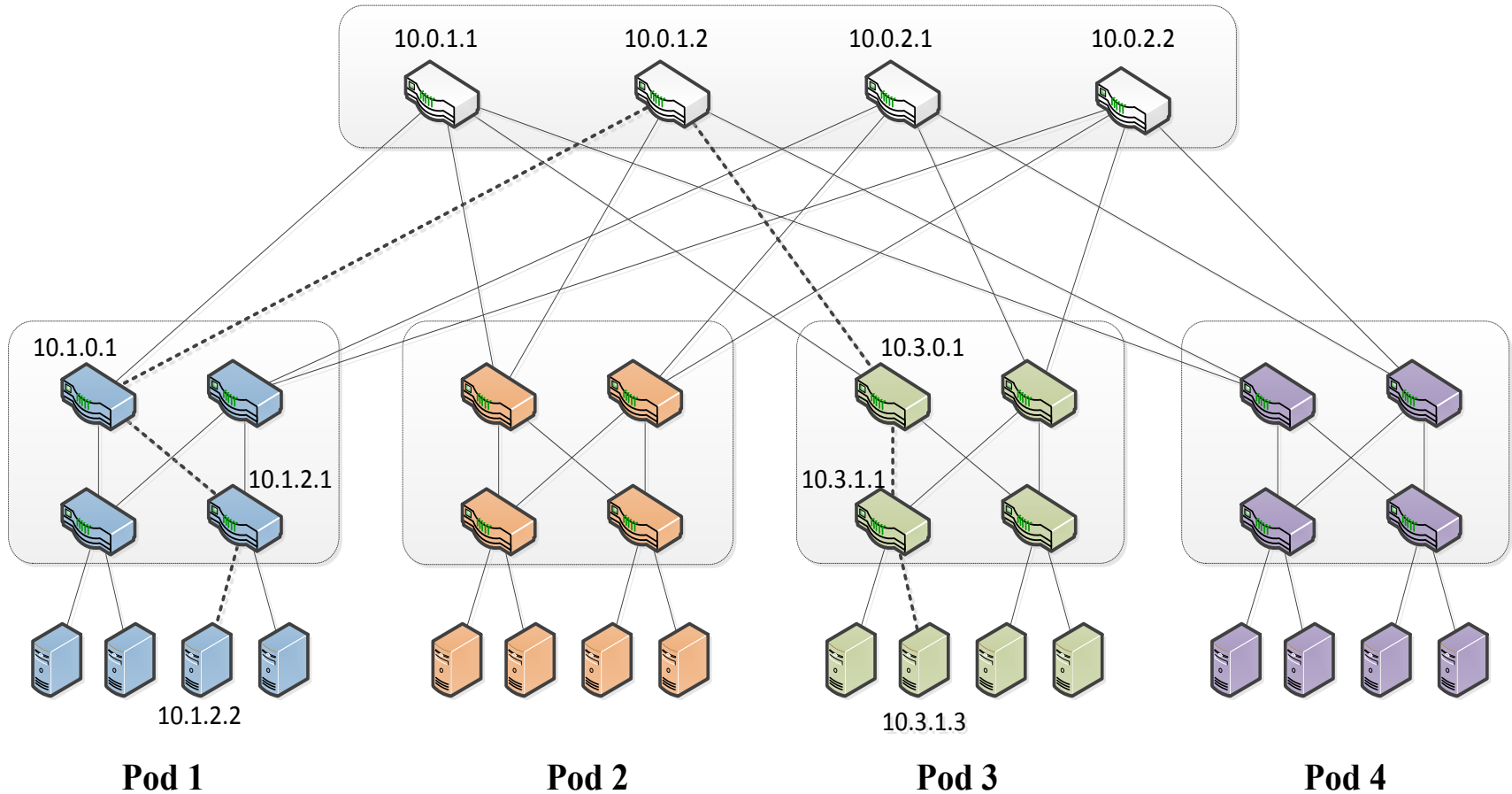
4. Forward packets to one of applicable next-hops,
according to source and destination MAC or randomly.

■

4

FAR Routing in Fat-Tree Networks

Example: Fat-Tree Network



The BRT of aggregation switch 10.1.0.1

- It is easy to build a BRT for a router according to its local topology
- We take 10.1.0.1 as an example. Its BRT is:

Destination/Mask	Next Hop
10.1.1.0/255.255.255.0	10.1.1.1
10.1.2.0/255.255.255.0	10.1.2.1
10.0.0.0/255.0.0.0	10.0.1.1
10.0.0.0/255.0.0.0	10.0.1.2

The NRT of aggregation switch 10.1.0.1

- A router's NRT is determined by locations of link or device failures in the network.
- There are several rules to calculate a router's NRT.
 - These rules are related to the regularity in topology.
 - Generally, single-link failures and some combination of link failures should be considered in the rules.
 - The draft presents the rules for Fat-tree Architecture.
- Suppose the link between 10.0.1.2 and 10.3.0.1 fails, The NRT of 10.1.0.1 is:

Destination/Mask	Next Hop
10.3.0.0/255.255.0.0	10.0.1.2

How does node 10.1.0.1 forward a packet to the destination 10.3.1.3

- 1) Calculate candidate hops. 10.1.0.1 looks up its BRT and obtains the following matched entries:

Destination/Mask	Next Hop
10.3.0.0/255.255.0.0	10.0.1.1
10.3.0.0/255.255.0.0	10.0.1.2

So the candidate hops = {10.0.1.1; 10.0.1.2}.

Contd../

- 2) Calculate avoiding hops. 10.1.0.1 looks up its NRT and obtains the following matched entries:

Destination/Mask	Next Hop
10.3.0.0/255.255.0.0	10.0.1.2

So the avoiding hops = {10.0.1.2}

- 3) Calculate applicable hops.

$$\begin{aligned}\text{applicable hops} &= \{10.0.1.1; 10.0.1.2\} - \{10.0.1.2\} \\ &= \{10.0.1.1\}\end{aligned}$$

- 4) Finally, forward the packet to the next hop 10.0.1.1.

5

Assessment of FAR's Performance

A Fat-tree network composed of 2,880 48-port switches and 27,648 servers is used to evaluate FAR's Performance

Required Messages

- 4 types of control messages (in-band) are required in FAR.
 - Hello
 - DLR: Device Link Request
 - DA: Device Announcement. The period of DA is typically 30 minutes.
 - LFA: Link failure Announcement

Message Type	Scope	Size	Rate	Bandwidth
Hello	Between adjacent switches	< 48 bytes	10 messages/sec	<4 kbps
DLR	Between adjacent switches	< 48 bytes	Produce one when a router starts	48 bytes
DA	In the entire network	< 48 bytes	The number of switches (2,880) in a period	1.106M
LFA	In the entire network	< 48 bytes	Produce one when a link fails or recovers	48 bytes

Routing Table Calculation Time

- The interval of sending Hello message is set to 100 ms, and a link failure will be detected in 200 ms.
- The spread time of a link failure between any pair of routers is less than 200 ms.
- FAR detects a link failure, spread it to all the routers, and calculates routing tables within 500 ms.

Size of the Routing Tables

- Suppose 1000 link failures occur
- FAR routing tables

Routing Table	Core Switch	Aggregation Switch	Edge Switch
BRT	48	48	24
NRT	0	14	333

- OSPF routing tables

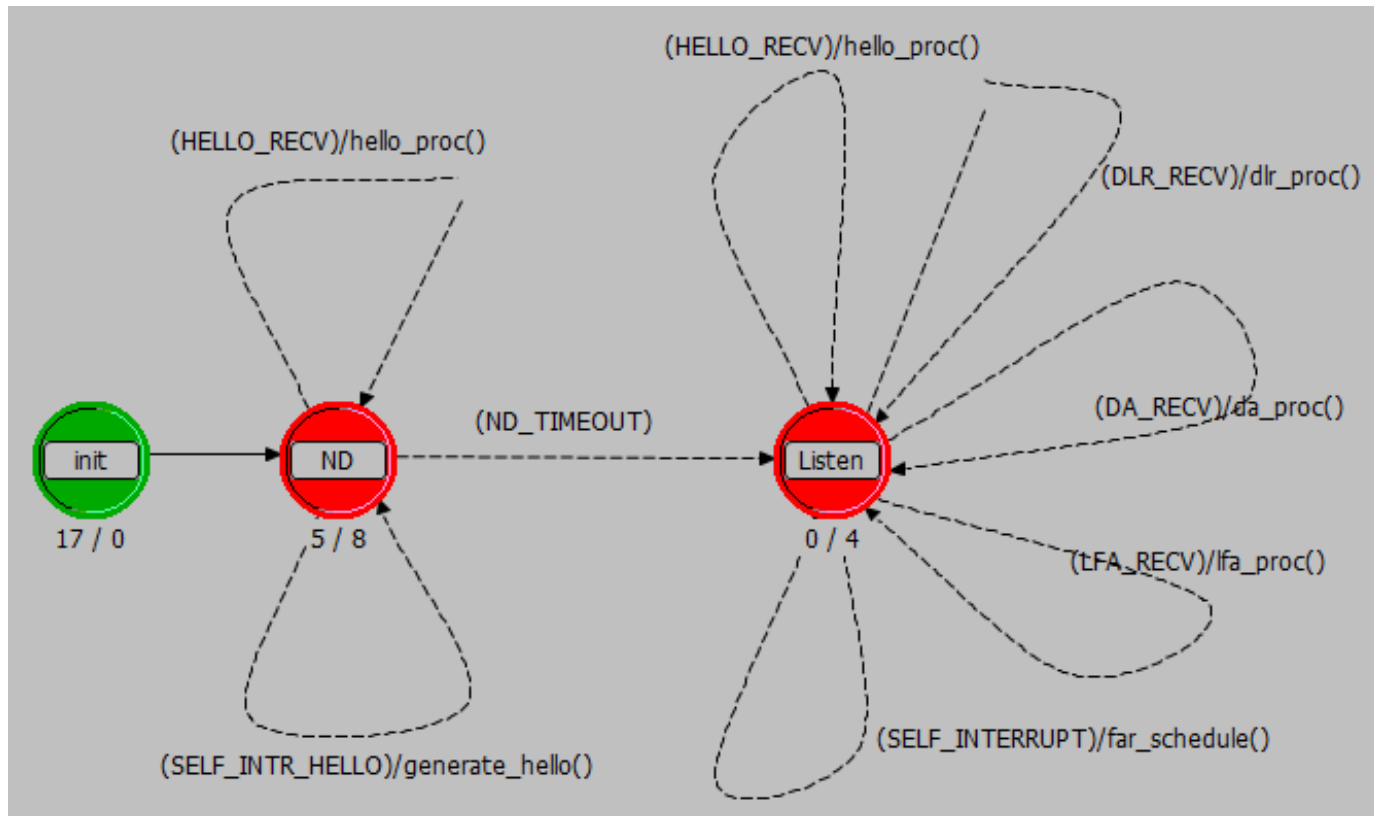
The Scale of Network	Core Switch	Aggregation Switch	Edge Switch
A Fat-tree network with 48-port switches	56,448	56,448	56,448

6

Verification of FAR by OPNET
Simulation

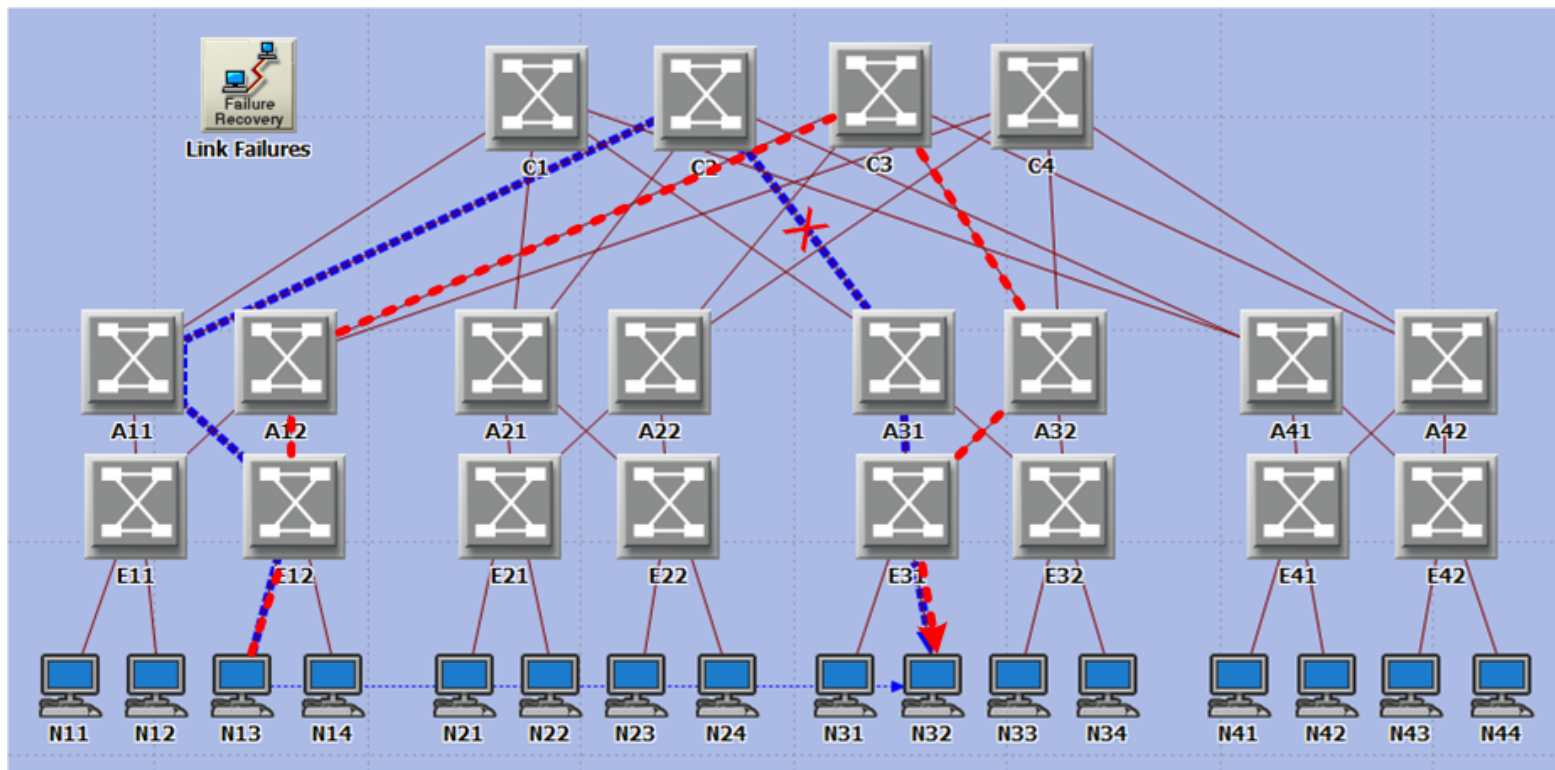
Simulation Model

- FAR switches are layer-3 switches, developed based on the standard layer-3 Ethernet switch model.
- FAR process is implemented as a process model in the standard layer-3 Ethernet switch model.
- FAR process model is placed over the ip_encap process model.

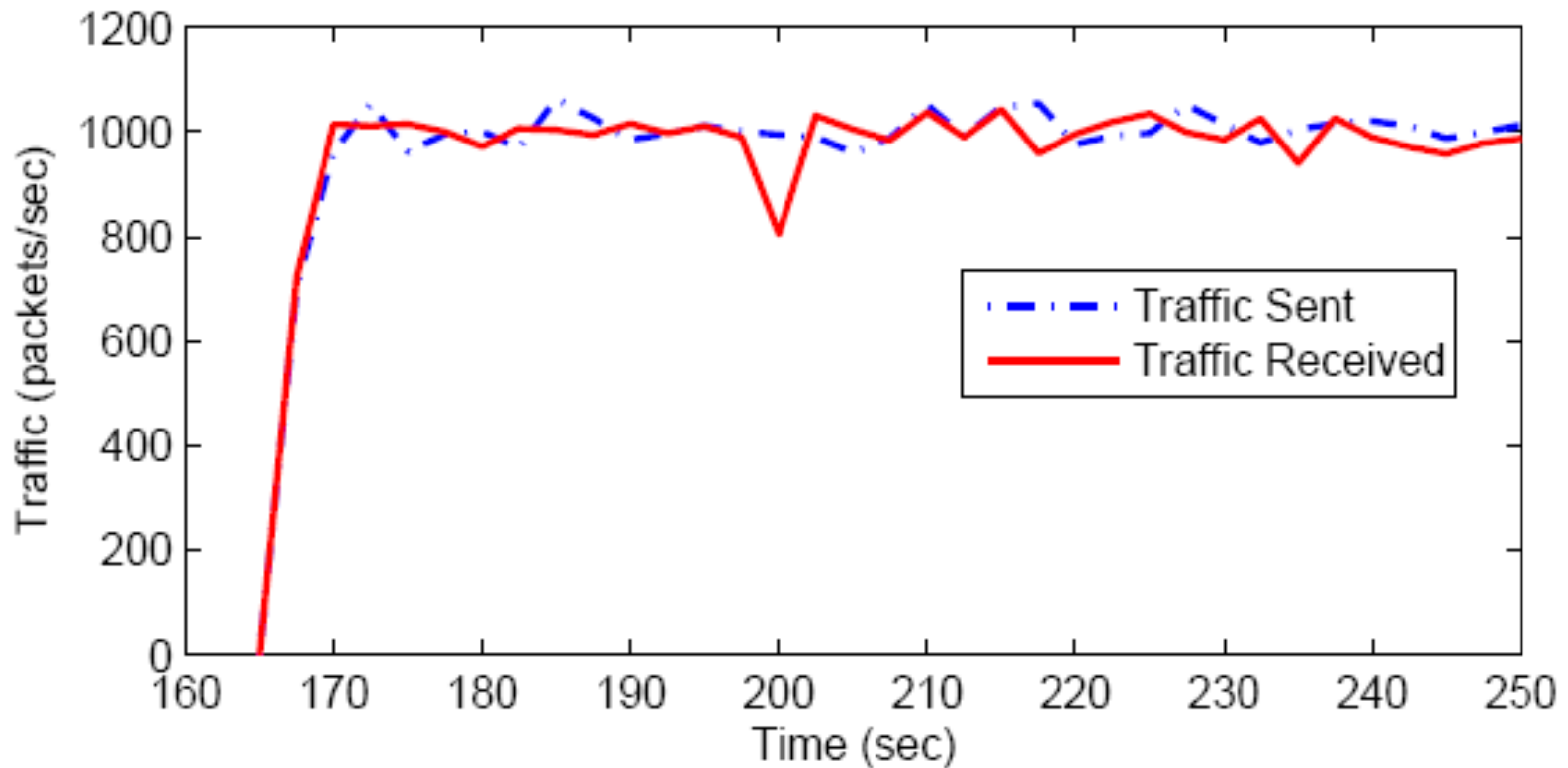


Reroute in FAR

- The traffic starts at second 165 and finishes at second 250.
- At first, the traffic is forwarded along the blue thick dotted line.
- At second 200, the link between C2 and A31 breaks, then FAR recalculates routing tables and the traffic is switched to the red thick dot line.



- The number of received packets in second 200 drops down



Traffic Sent & Received in FAR

7

Conclusions

Conclusions

- FAR doesn't require network convergence and calculating the shortest path tree
 - shortens the time of calculating routes
 - accelerates its response time to network changes
 - relieves the computing burdens of a router.
- In FAR, the calculating of a BRT and NRT is very simple and requires only a few computations.
 - it can be quickly completed in several milliseconds, even for very large scale data center networks.
- FAR requires less control messages.
 - FAR knows the topology information of a network, so link state exchanges are not required in FAR.

Contd../

- The size of routing tables in FAR is very small.
 - A BRT only has tens of entries
 - an NRT has no more than hundreds of entries.
 - It is very fast to look up routing tables in FAR.
- The configuration of a network is simpler in FAR.
 - Only one IP address is configured to a router. All controller card share one on IP address in a FAR router .
- FAR has very good adaptability.
 - It can be used in many kinds of data center network topologies with slight modifications.

Next Steps

- In the past, no draft has discussed routing problem in regular network topology in Data Centers
- Requesting IETF RtgWG to consider adoption of this draft

■

Thanks!