

Kathará

data centers' architecture

Version	3.1
Author(s)	L. Ariemma, G. Di Battista, M. Patrignani, M. Scazzariello, T. Caiazzì
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	Data Centers' Routing: Multipath, Fat-Trees



copyright notice

- all the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as “material”) are protected by copyright
- this material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide
- this material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes
- any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement
- this copyright notice must always be redistributed together with the material, or its portions



Multi-Path

in a nutshell



Multi-Path

- data center architectures allow to establish several paths between pairs of nodes
- Multi-Path routing is used to
 - distribute and balance the traffic among different paths
 - increase the fault tolerance



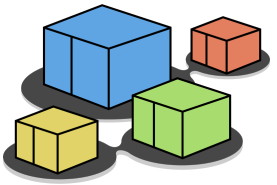
Multi-Path

- allows to have more than one next-hop for the same prefix in the FIB (Forwarding Information Base)
- needs kernel support
 - the main OSes and routers support it
- different usage policies can be applied
- packets of the same *flow* should use the same path
 - reordering packets is computationally heavy



Equal-Cost Multi-Path (ECMP)

- a specific approach to Multi-Path
- exploits paths to the same destination that have the same “cost”
 - e.g., same IGP cost, same number of hops
- allows to choose among the available paths in a uniform way
- often used in Fat-Tree data centers
 - we’ll get there in a couple of slides



how packets are forwarded

- for each packet, the kernel decides the FIB entry to be used
- different policies are allowed
 - hash-based policy
 - Layer-3 hash (src IP, dst IP)
 - Layer-4 hash (src IP, dst IP, src port, dst port, protocol)
 - round-robin



Hyper-scale Data Centers

main concepts

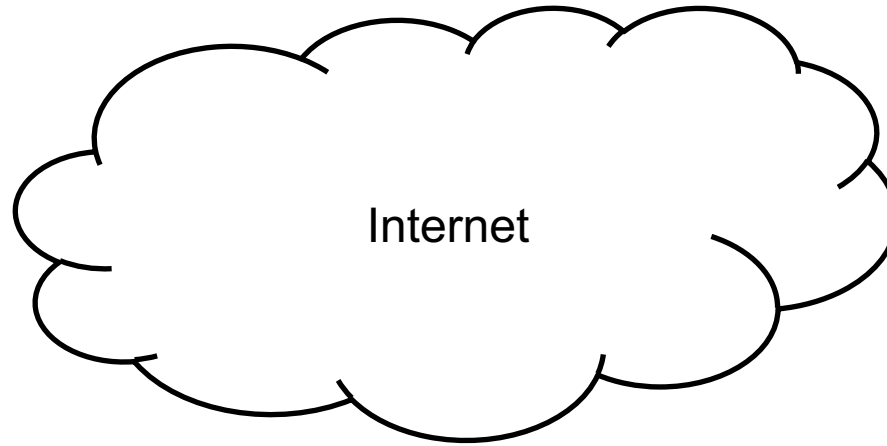


a high-level architecture

- Internet connections
 - multiple redundant high-speed fiber optic links
- fabric
 - infrastructure designed to transport packets between servers and between servers and Internet
- server farm
 - host applications and services



a high-level architecture





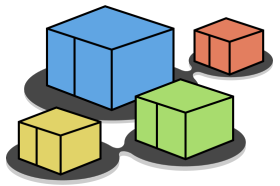
fabric architecture

why Fat-Trees?



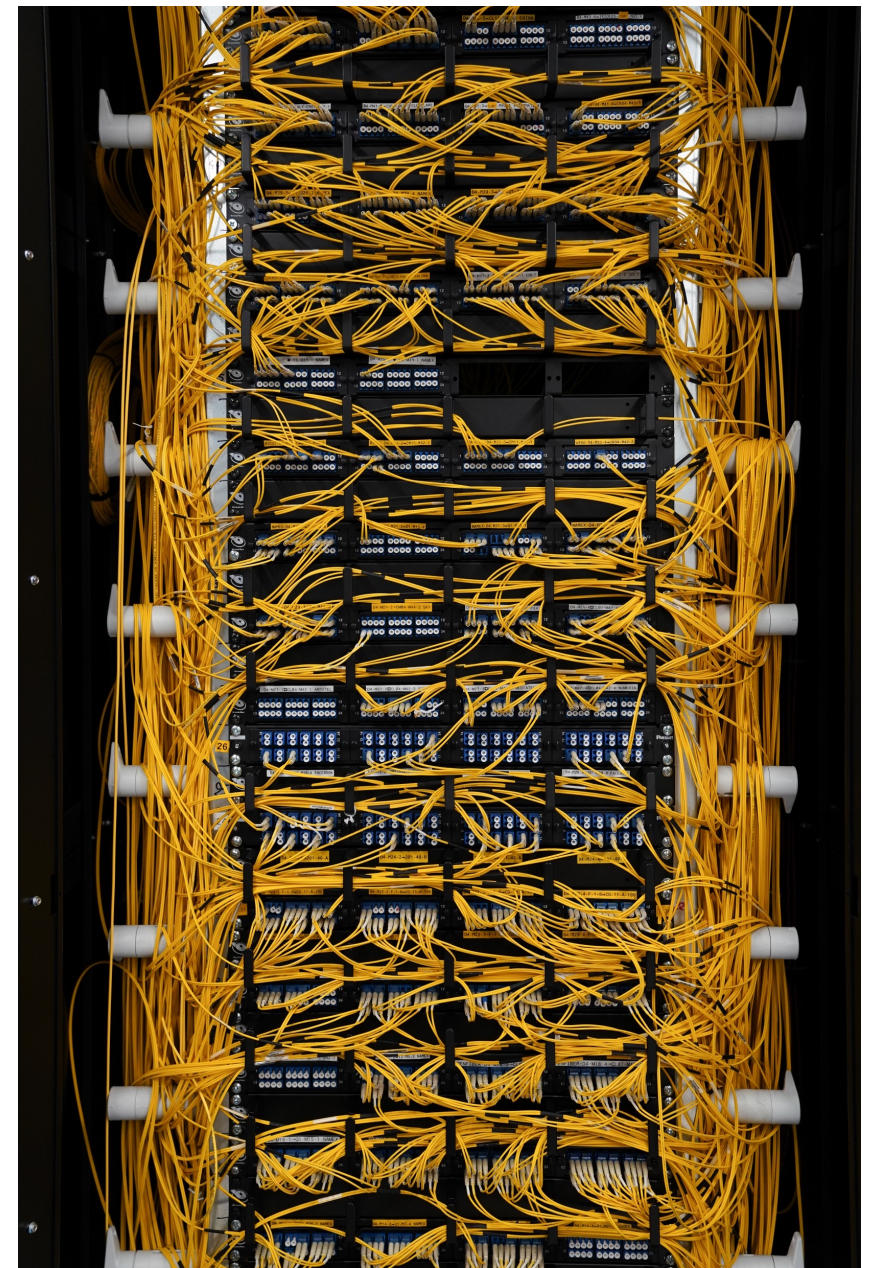
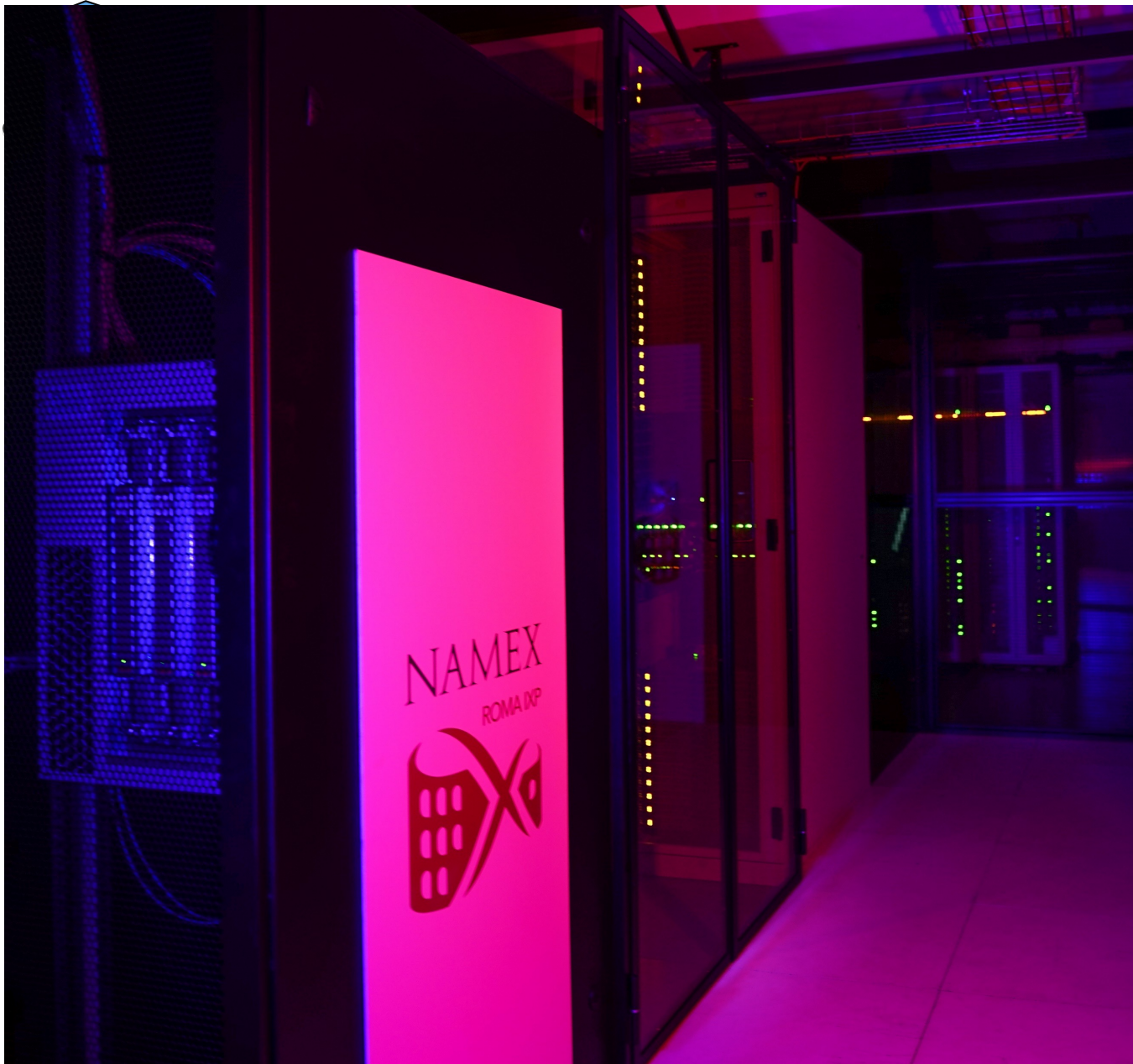
overview – fabric architecture

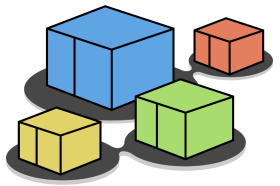
- real-life data centers
- service model
- data center network
 - traffic flow directions
 - requirements
 - components
 - topologies



real data center information – figures

- links are fiber optic links
- each link in the fabric is at least at 100Gb/s
- each switch has at least 64 ports
- about 5,000 switches/routers
- about 60,000 servers





data center failures – yearly report

statistics by Google (2008) in a *portion* of data center composed of 1,800 servers:

- 1,000 individual machine (switch/router or server) failures
- thousands of hard drive failure
- 1 power distribution unit failure
 - down for 500/1000 machines for 6 hours
- 1 cluster complete rewire (not simultaneously)



data center service models

■ three different service models

■ on-premise data centers

- built, owned and operated by a company
- often housed in a building of the organization

■ colocation data centers

- built and owned by a company that rents space within the data center to other companies
 - the hosting company manages the infrastructure (building, cooling, bandwidth, security etc.)
 - the hosted companies provides the components, including servers, storage, and firewalls

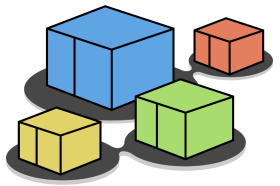
■ cloud data centers

- applications and data are hosted by a cloud service provider such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP)

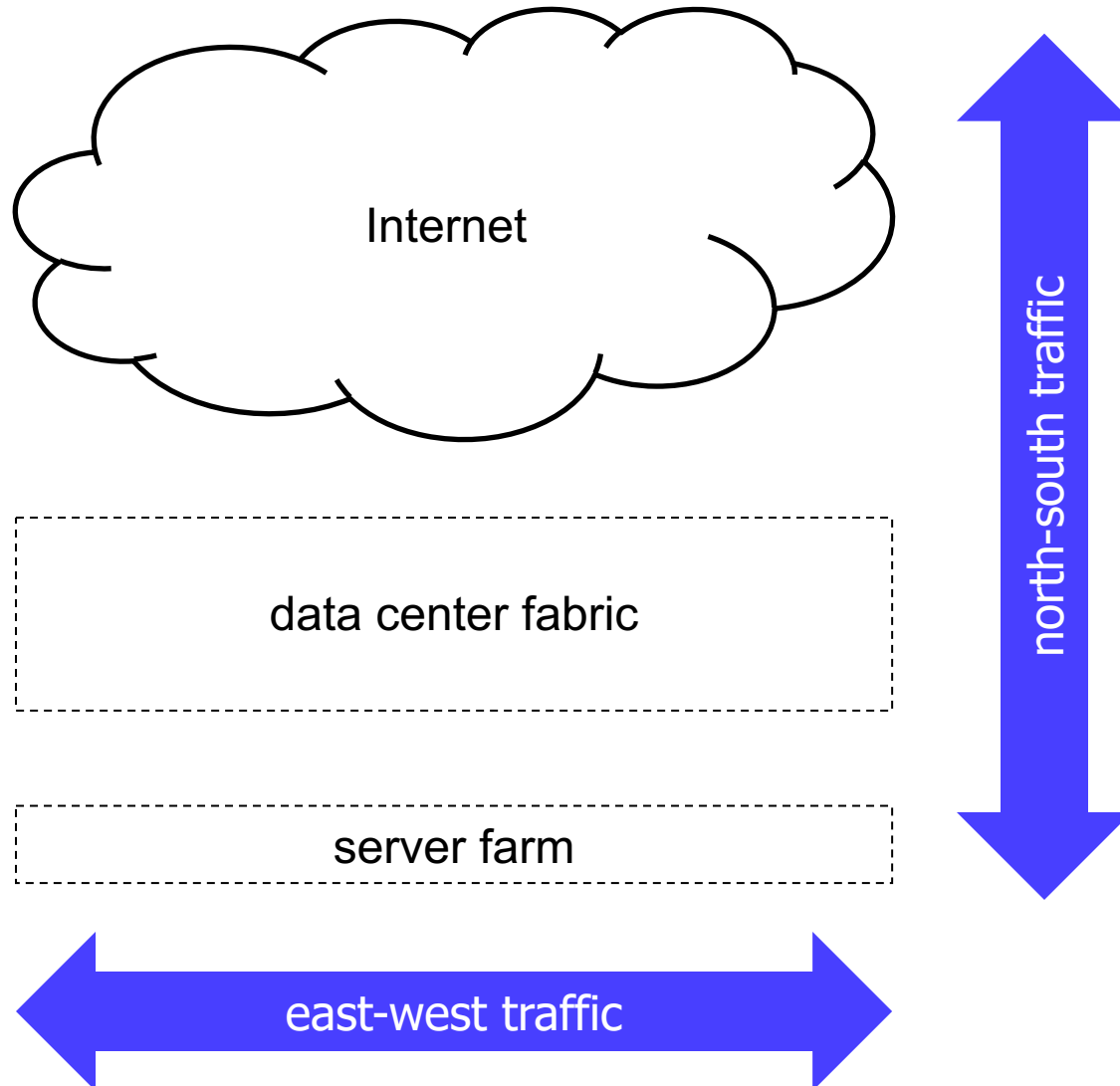


traffic flows inside a data center

- two types of flows can be identified
 - traffic exiting or entering the data center
 - also called “north-south traffic”
 - data sent/received via Internet
 - server-to-server communications
 - also called “east-west traffic”: in data center schemata servers are typically drawn side-by-side
 - primarily, supports micro-services and distributed architectures



traffic flows inside a data center





data center network requirements

- support high-bandwidth server-to-server communication
 - applications that rely on cluster computations, such as Hadoop or Spark, can involve hundreds or thousands of servers
 - customer's containers/virtual machines (VMs) are distributed across multiple servers but need to communicate seamlessly
 - microservice architectures heavily rely on server-to-server communication
- scale
 - data centers range from a few hundred to a hundred thousand servers in a single physical location
- resilience
 - data center applications are designed to work in presence of failures



data center network components

- data center nodes can be connected by using two network component types
 - specialized hardware
 - proprietary hardware that can scale to clusters of thousands of nodes with high bandwidth
 - e.g., Google Jupiter and InfiniBand switches
 - expensive option
 - commodity switches and routers
 - cheap option
 - widely adopted, this is the option we consider in the following



topology requirements

- scalability

- it should be possible to expand the data center
- in other words, it must be possible to buy and deploy hardware similar to the one already deployed

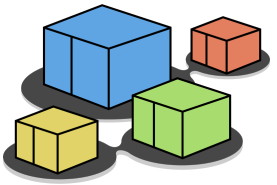
- bandwidth

- hosts in the fabric should communicate with each other using the full bandwidth of their NICs



data center network topologies

- several topologies have been proposed
 - Clos
 - Fat-Tree
 - VL2
 - Jellyfish
 - Xpander
 - DCell
 - ...



Clos topology

- invented by Edson Erwin in 1938 to address scalability issues in telephone networking
 - formalized by Charles Clos in 1953
- the original problem was allowing N contemporary connections from N input lines to N output lines without using a single crossbar switch



Fat-Tree topology

- Fat-Trees were originally introduced by Charles Leiserson in 1985
- the Fat-Tree is a special case of a Clos
- typical Fat-Tree architectures today consist of three level of nodes (switches/routers)
- high redundancy
- constant *bisection* of the available bandwidth
- typically, nodes have the same characteristics
 - easier to stock spare equipments



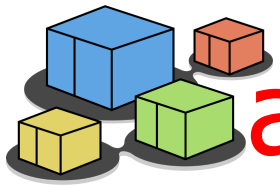
Fat-Tree parameters

- the Fat-Tree is a modular topology
- two parameters define a Fat-Tree
 - *radix* of the nodes (even number, denoted by $2K$)
 - number of available ports
 - *redundancy factor* (denoted by R)
- usually, if the radix of a node is $2K$ it has K connections towards the north and K connections towards the south
 - different choices are possible, but not considered here

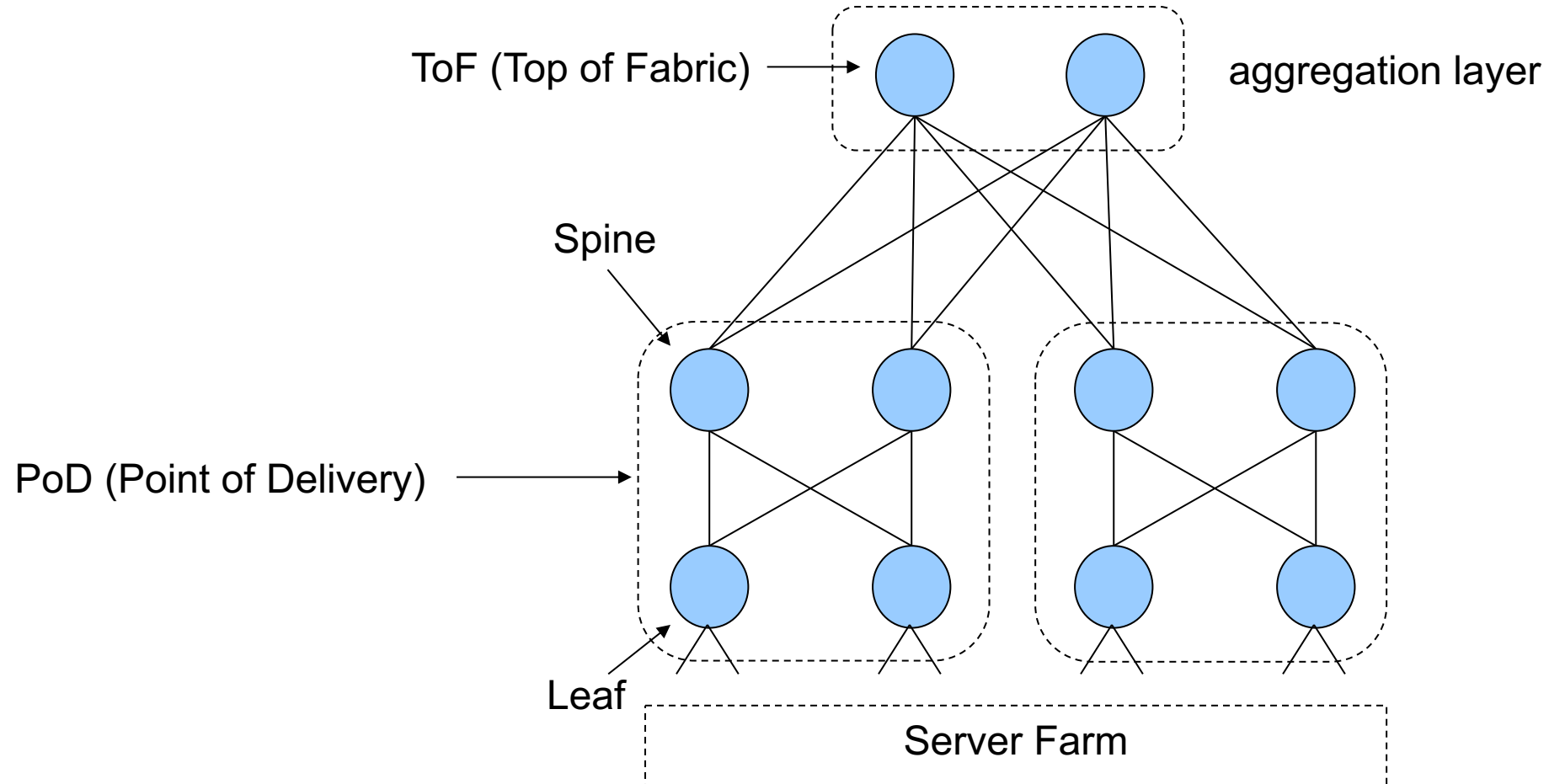


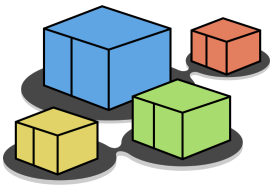
Fat-Tree nodes

- Leaf
 - node connected to the server farm
- Spine
 - node north of Leaves and south of ToF nodes
- Point of Delivery (PoD)
 - set of fully interconnected Leaves and Spines
- Top of Fabric (ToF)
 - set of top nodes that provide inter-PoD communication



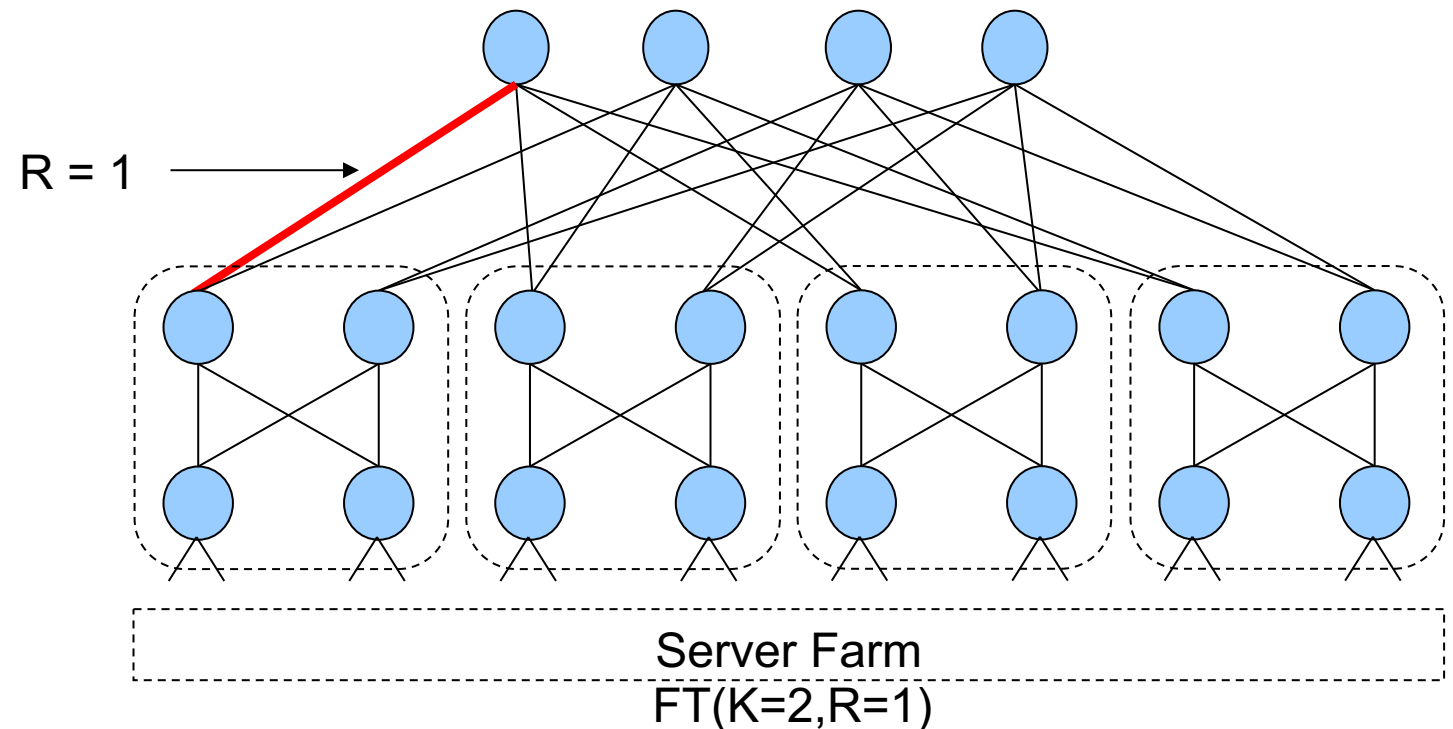
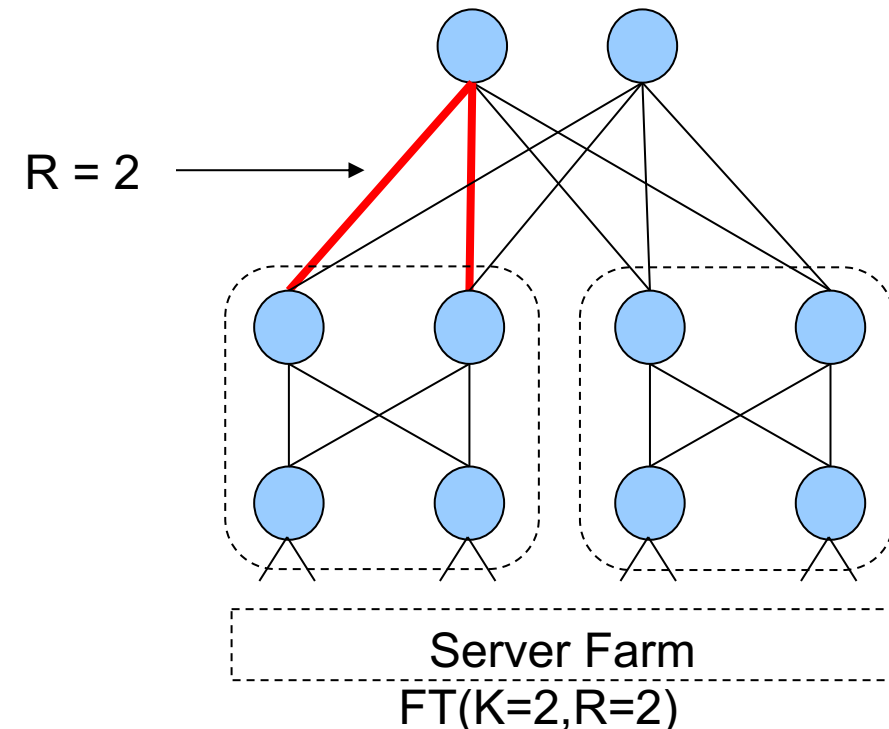
an example of Fat-Tree: $FT(K=2, R=2)$

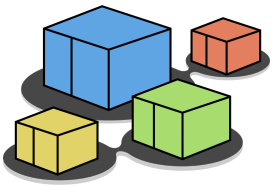




redundancy factor (R)

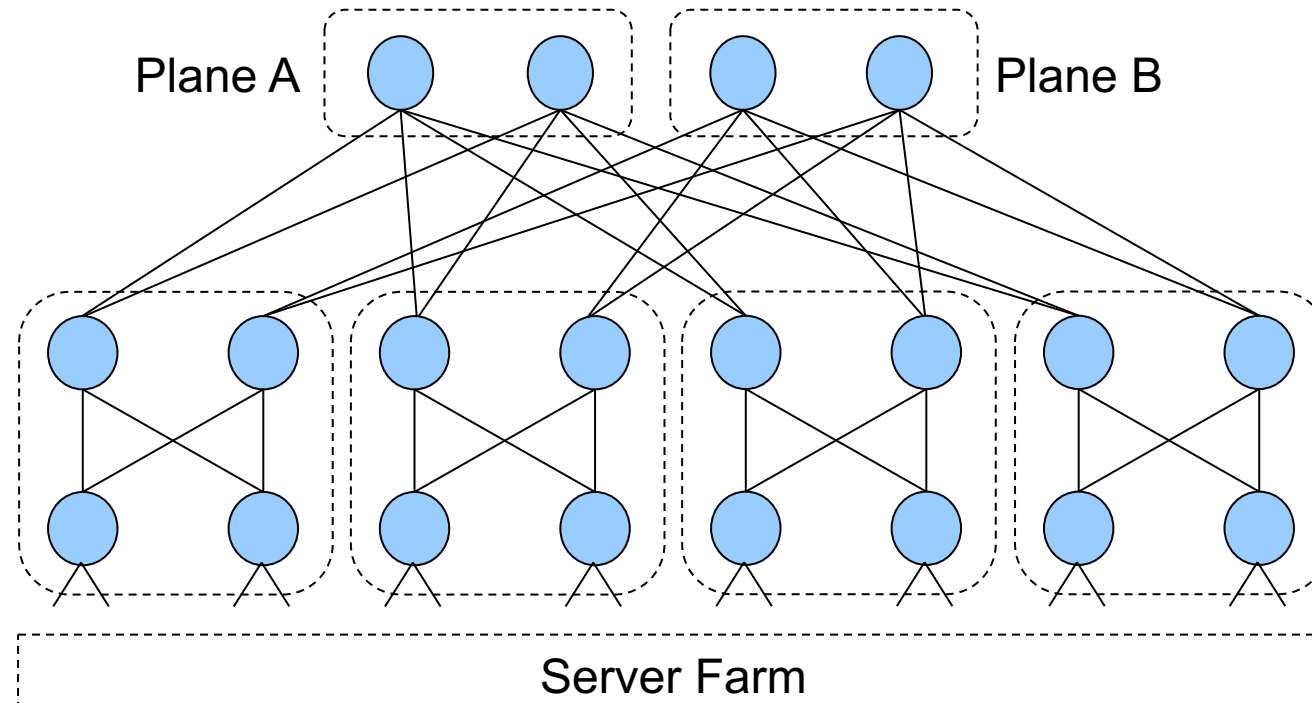
- number of links between a ToF node and a PoD
 - max = K , min = 1
- allow to connect more PoDs reducing the redundancy

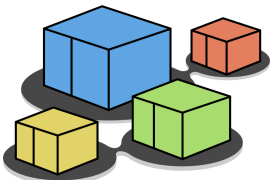




multi-plane Fat-Tree

- when $K \neq R$ the Fat-Tree is called *multi-plane*
- the ToF nodes are partitioned in sets called *planes*





example of a multi-plane Fat-Tree

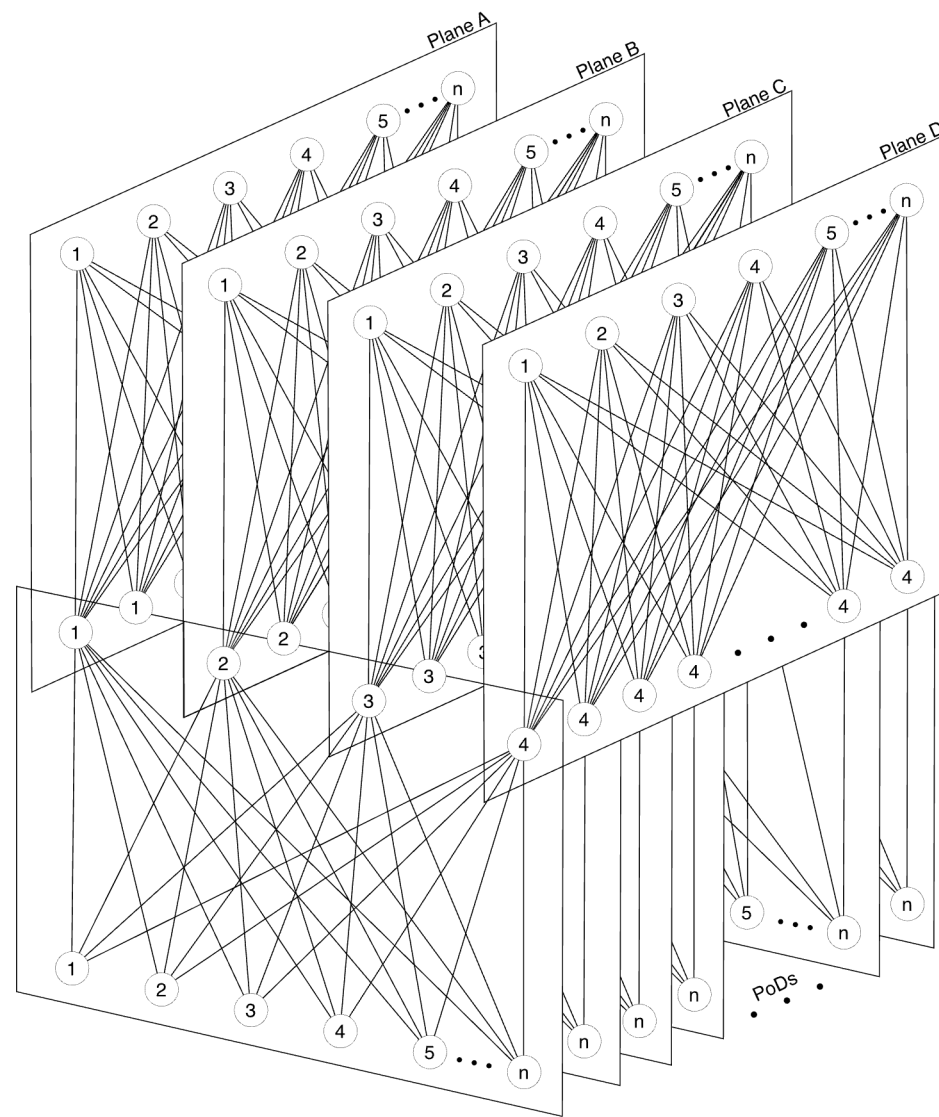
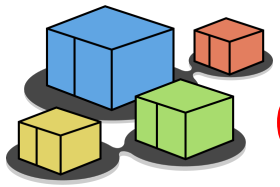
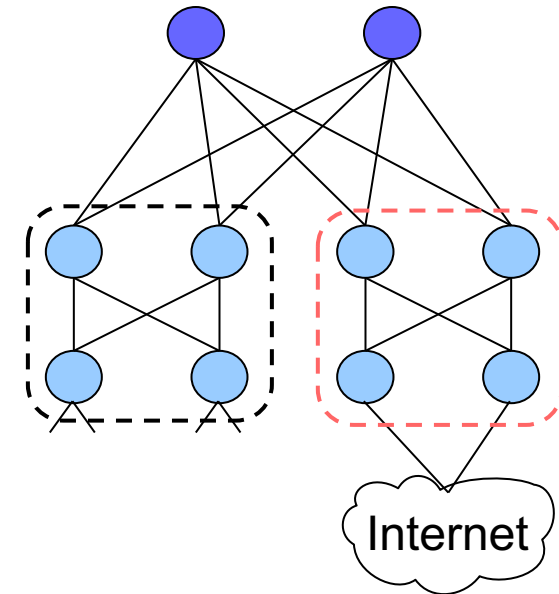
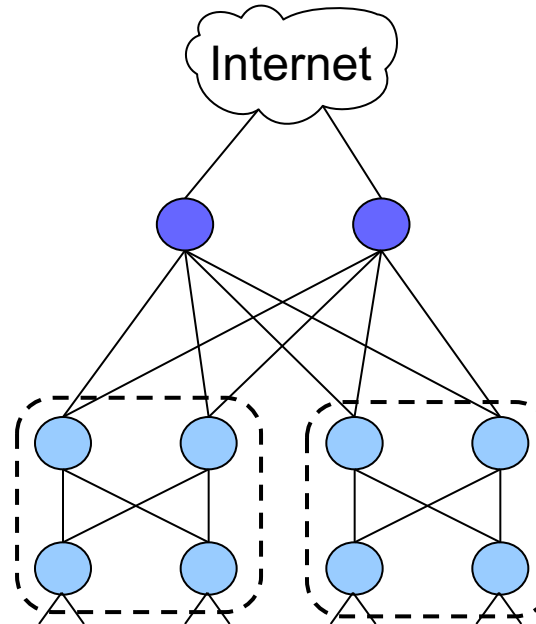


diagram from the
IETF RIFT Draft



connecting a Fat-Tree to the Internet

- two strategies
 - usage of a dedicated PoD
 - usage of ToFs





bibliography and further readings

- [Caiazzì '22] Caiazzì, Scazzariello, Alberro, Ariemma, Castro, Grampin, Di Battista, "Sibyl: a Framework for Evaluating the Implementation of Routing Protocols in Fat-Trees", NOMS 2022
- [Caiazzì '21] Caiazzì, Scazzariello, Ariemma, "VFTGen: a Tool to Perform Experiments in Virtual Fat Tree Topologies", IM 2021
- [Caiazzì '19] Caiazzì, "Software Defined Data Centers: methods and tools for routing protocol verification and comparison", Ms. Thesis, Roma Tre University, 2019
- [Dutt '17] Dutt, "BGP in the Data Center", O'Reilly, 2017
- [RFC-7938] Lapukhov, Premji, "Use of BGP for Routing in Large-Scale Data Centers" Internet Engineering Task Force (IETF) Request for Comments: 7938