b

UNIVERSITÄT
BERN

# Advanced Networking and Future Internet

## VI. Data Centre Networking

**Prof. Dr. Torsten Braun, Institut für Informatik**

Bern, 19.10.2020

# Advanced Networking and Future Internet: Data-Centre Networking

UNIVERSITÄT BERN

## Table of Contents

# 1. Cloud Computing

## 1. Definition

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.“

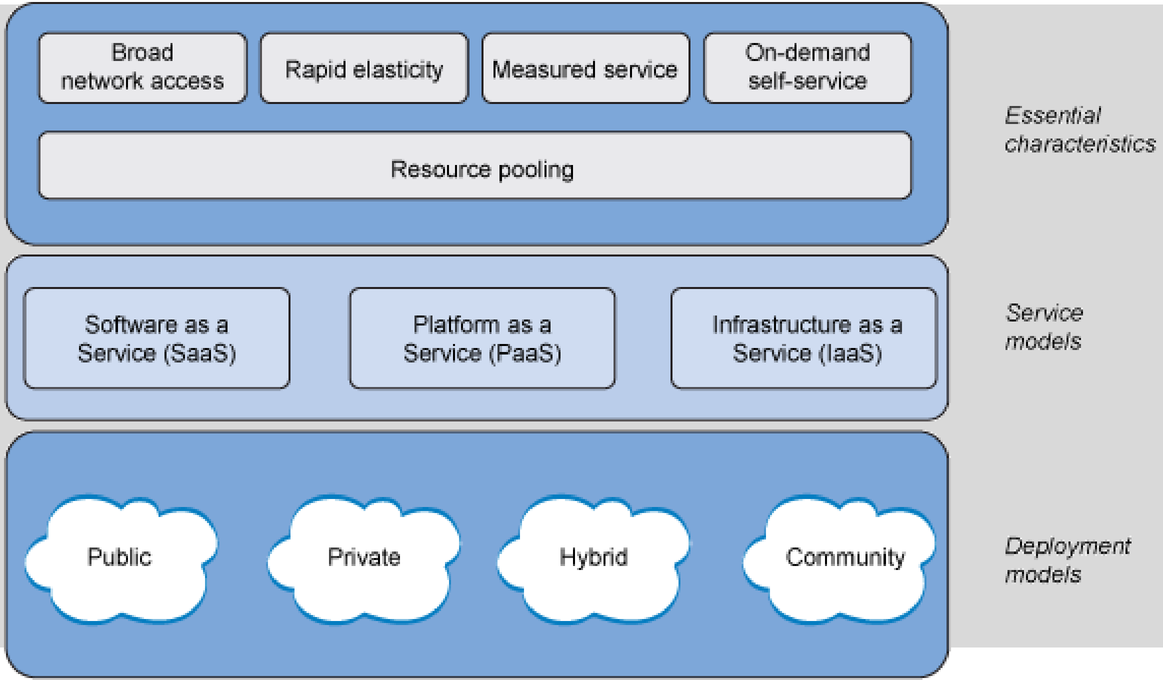
(NIST: National Institute of Standards and Technology) b UNIVERSITÄT BERN

# 1. Cloud Computing

# 2. Features

- Elastic resources
  - Expand and contract resources
  - Pay-per-use
  - Infrastructure on demand
- Multi-tenancy
  - Multiple independent users
  - Security and resource isolation
  - Amortize cost of the (shared) infrastructure
- Flexible service management
  - Resiliency: isolate failure of servers and storage
  - Workload movement: move work to other locations

# 1. Cloud Computing
## 3. NIST’s Cloud Computing Model



# 1. Cloud Computing

## 3.1 Deployment Models

**Public clouds**

– External or publicly (in the Internet) available cloud environments

**Private clouds**

– Typically tailored environment with dedicated virtualized resources for users of a particular organization

**Community clouds**

– Typically tailored for particular groups of customers

**Hybrid clouds**

– a composition of two or more clouds (private, community, or public) that remain unique entities offering the benefits of multiple deployment models

# 1. Cloud Computing

## 3.2 Service Models

**Infrastructure as a Service (IaaS)**

- Cloud provider supplies a set of virtualized infrastructural components such as virtualized machines (VMs), storage, network etc. on which customers can build and run applications.
- Avoidance of buying servers and estimating resource needs

**Platform as a Service (PaaS)**

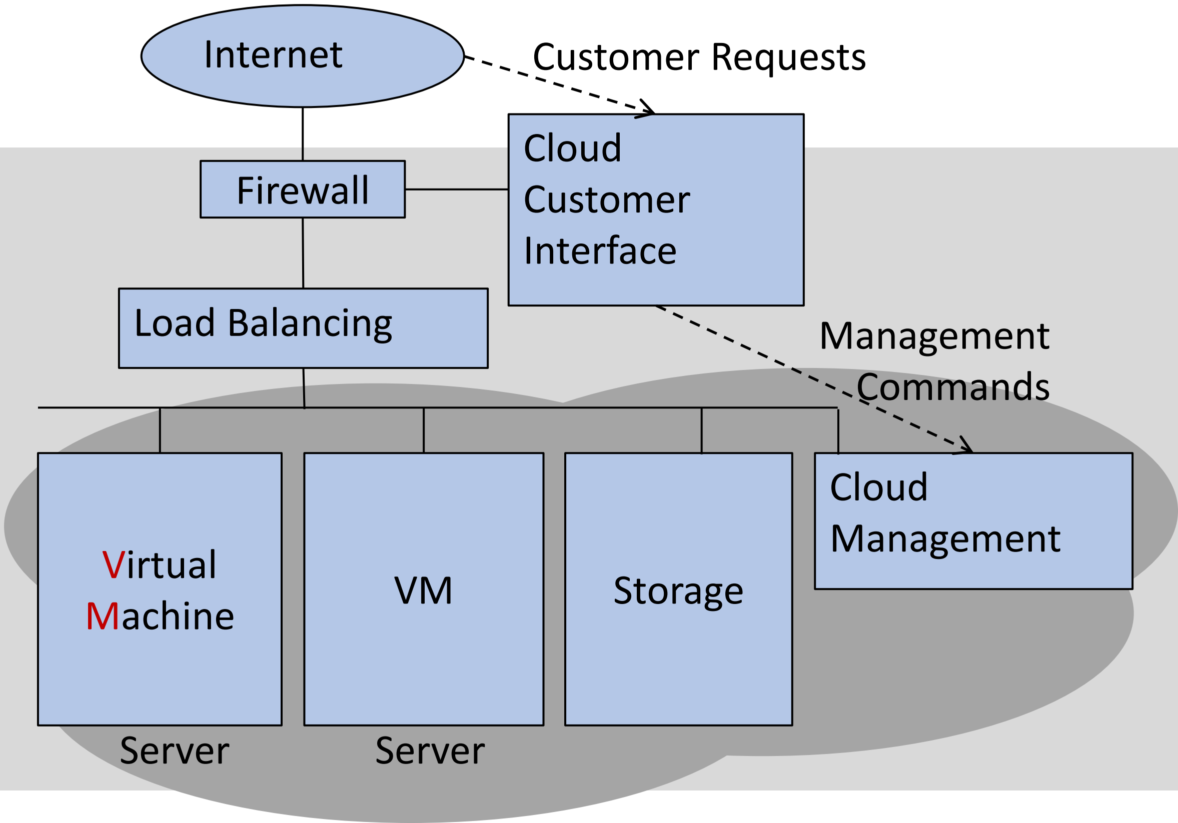
- Enables programming environments and middleware (e.g., databases) to access and utilize additional application building blocks
- Goal: enable developers to build their own application on top of the platform provided

**Software as a Service (SaaS)**

- Cloud providers enable and provide application software as on-demand services
- Avoidance of costs for installation, maintenance, etc.

# 1. Cloud Computing

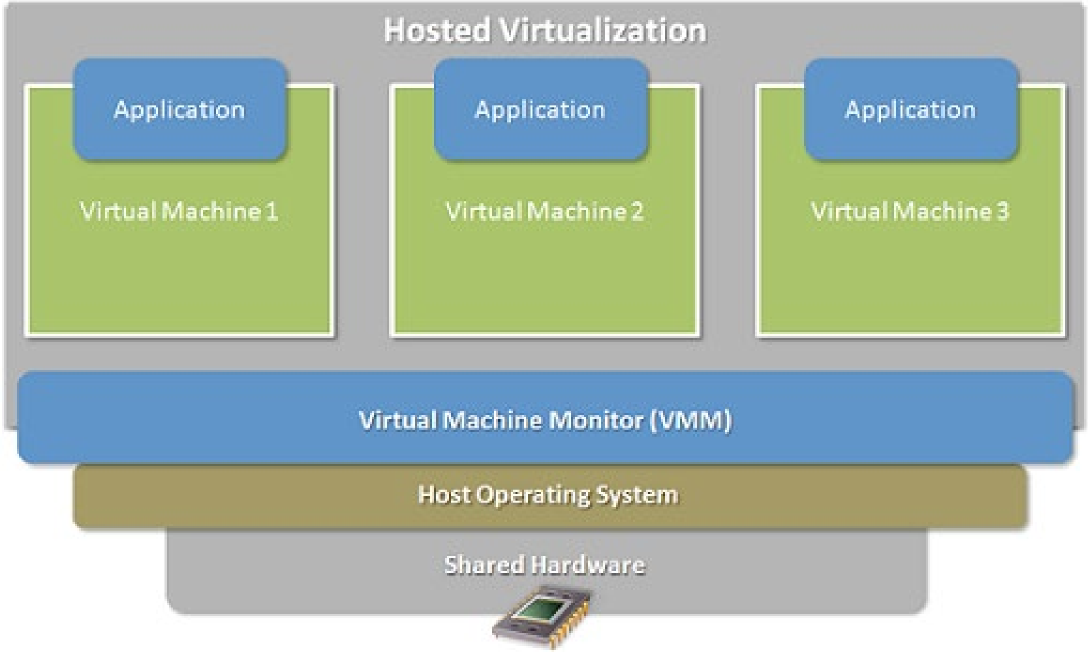
## 4. Infrastructure as a Service



u^b UNIVERSITÄT BERN

# 2. Virtualization

- Enabling technology
- Multiple VMs run on 1 physical machine.
- Applications run unmodified as on a real machine.
- VM can migrate from one physical machine to another.



b UNIVERSITÄT BERN

# 2. Virtualization

## 1. Goals

- High level goals
  - Efficiency
  - High availability
  - Lower cost
- Lower level goals
  - Multiple applications / operating systems on each server
  - Maximum server utilization with a minimum number of servers
  - Faster and easier application and resource provisioning

- "The art of virtualization is the art of sharing."
- Virtualization allows 1 computer to do the job of multiple computers by sharing the resources of a single computer across multiple environments.

# 2. Virtualization

## 2. Implementation

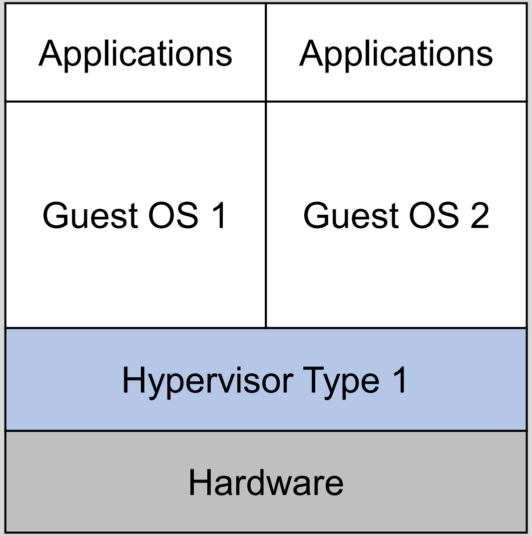
- Virtual Machines (VMs)
  - Isolated software container with operating system and applications
  - VMs are separate and independent.
  - Many of them can run simultaneously on a single computer.
- Hypervisor / Virtual Machine Manager (VMM)
  - decouples the virtual machines from the host(s).
  - dynamically allocates computing resources to each VM as needed.
- Computing, Storage, Network Resources

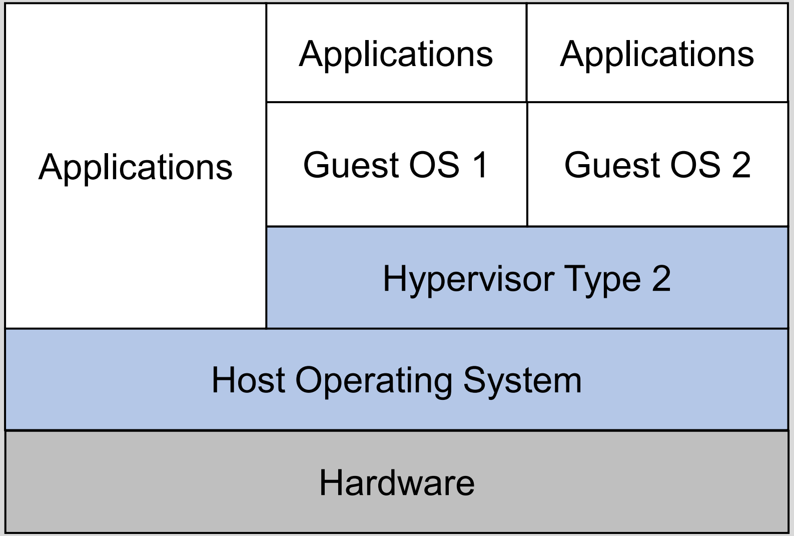
u^b b UNIVERSITÄT BERN

# 2. Virtualization

## 3. Hypervisor Types

0. Hardware solutions, e.g., Lynxsecure
1. Operating system (OS) like software (possibly extension of standard operating systems) to enable virtualization, e.g., Linux with KVM
2. Applications running on standard operating systems and offering Hypervisor functionality to guest operating systems



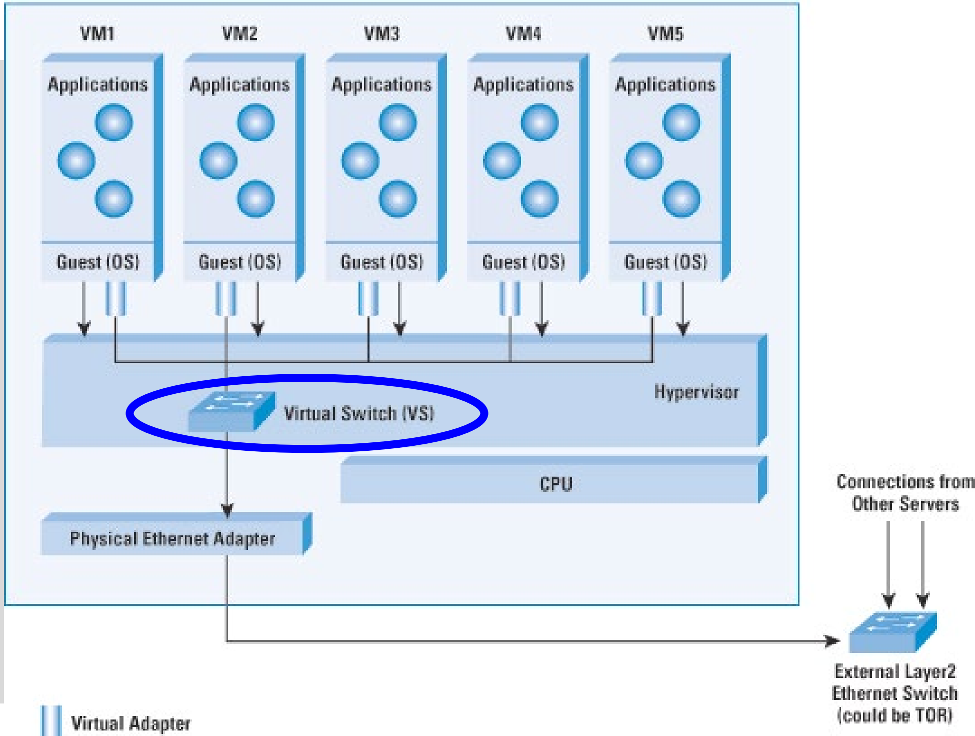



UNIVERSITÄT BERN

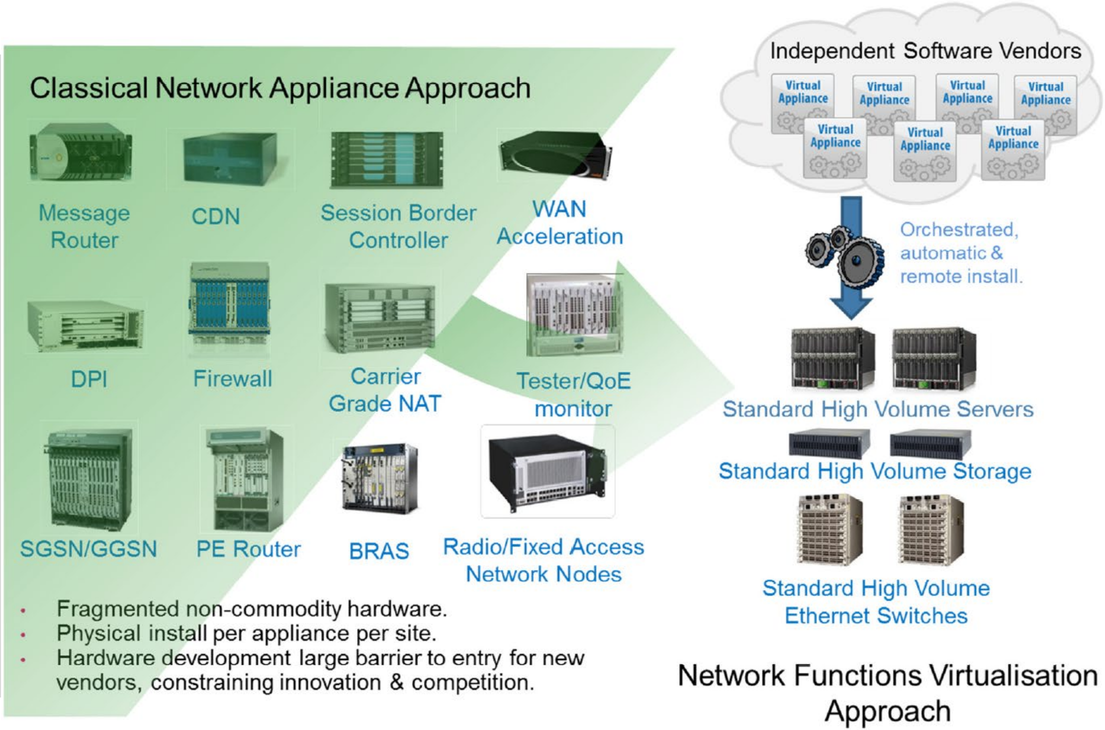
# 2. Virtualization

## 4. Virtual Switch

Switch based on shared memory passing pointers between VMs

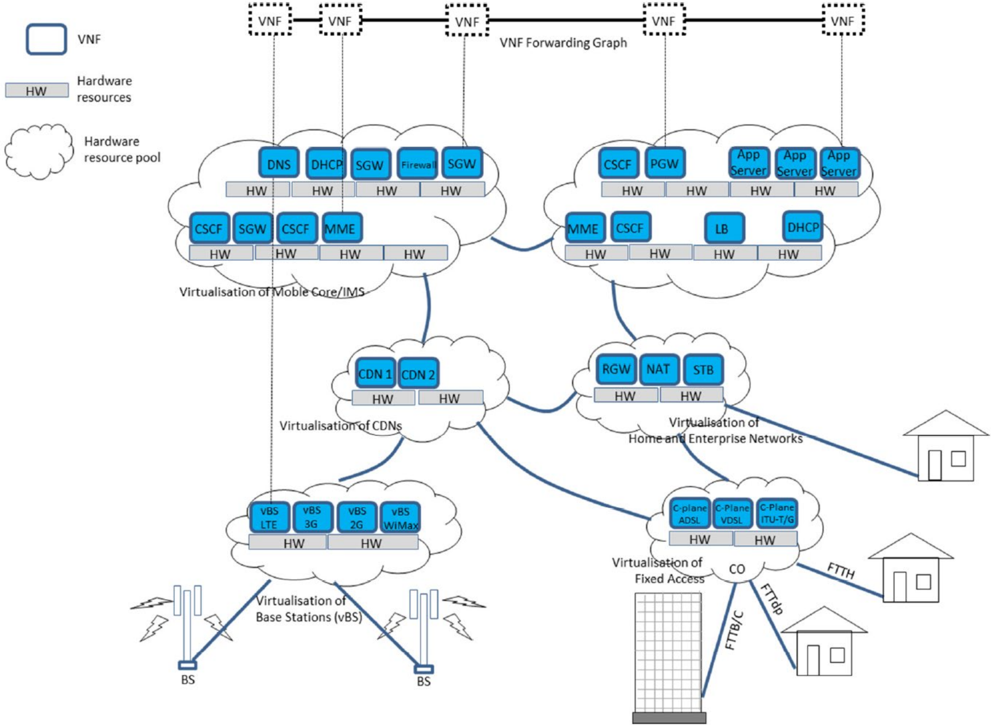


# 3. Network Function Virtualization



# 3. NFV

# 1. Use Cases

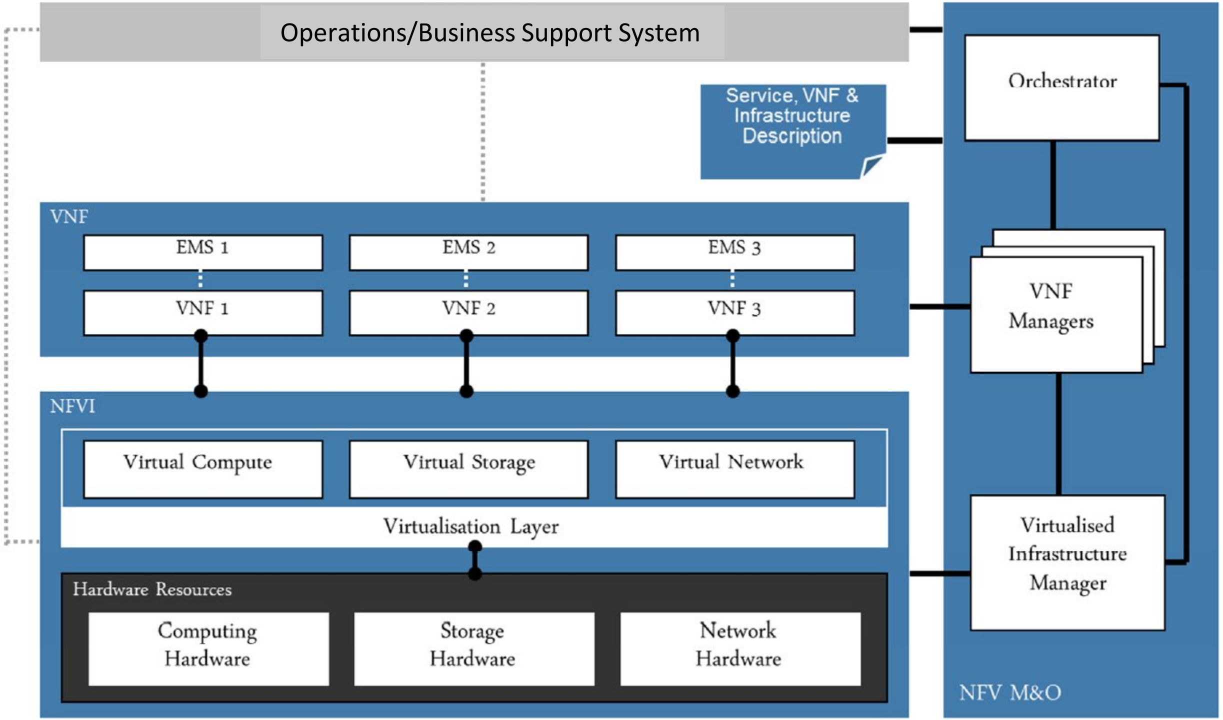


UNIVERSITÄT BERN

# 3. NFV

# 2. Architecture

Element Management System
Virtual Network Function




Management & Operation

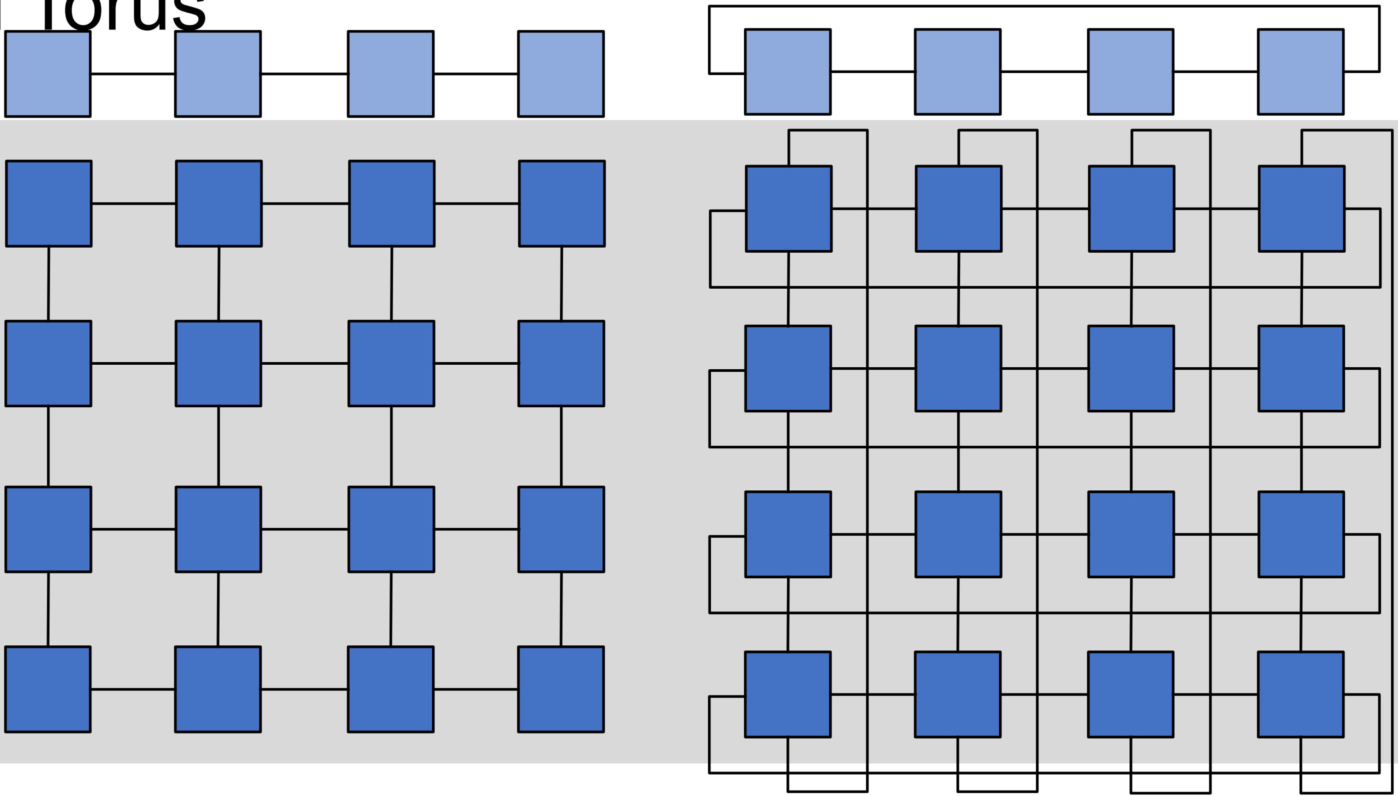
# 4. High-Radix Topologies

- Direct and indirect networks
  - Direct networks: processing nodes attached to switching fabric, i.e. switching fabric distributed among nodes
  - Indirect networks: network independent of nodes, i.e. dedicated switches

- High dimension mesh, torus, hypercube
  - Direct networks with high-radix routers (high number of ports)
  - High dimension topologies reduce network diameter, but significantly increase wire and cabling complexity.
- Indirect networks
  - better exploit high-radix routers while reducing network cost and cabling complexity
  - Examples
    - (flattened) butterfly networks
    - (folded) Clos networks

# 4. High-Radix Topologies

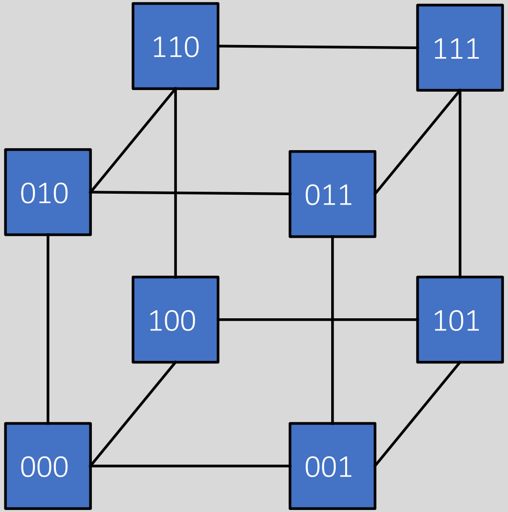
## 1. Mesh and Torus



# 4. High-Radix Topologies

## 2. Hypercubes

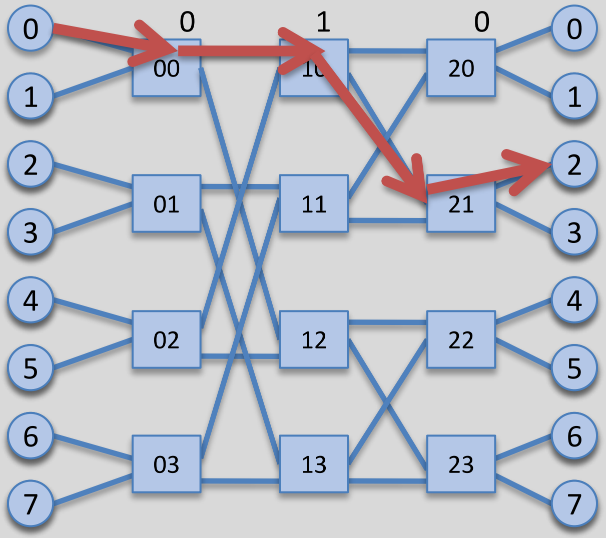
- Nodes that differ in only one digit are connected to each other.
- Routing in a hypercube will require up to n (= number of dimensions) hops, if the source and destination differ in every dimension.

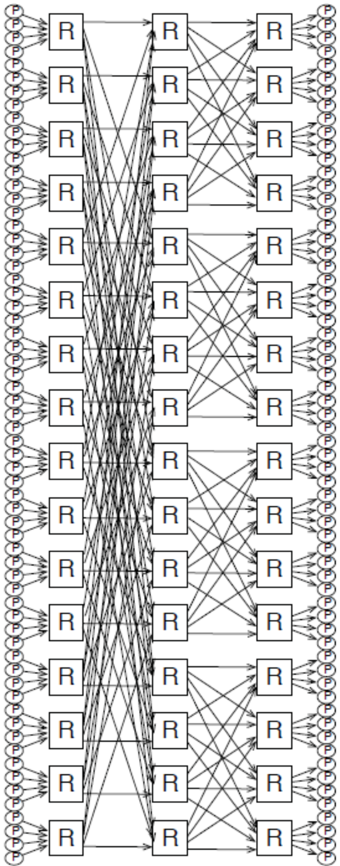


# 4. High-Radix Topologies

## 3. Butterfly Network

- k-ary n-fly: $N=k^n$ nodes,
  e.g., 2-ary 3-fly: N=8 / 4-ary 3-fly: N=64
- Routing from 000 to 010
  - Destination address used to directly route packet
  - Bit n used to select output port at stage n
- indirect network
- $\log_k(N) + 1$ hops
- N/k routers per stage
- Lack of path diversity
- No exploitation of traffic locality



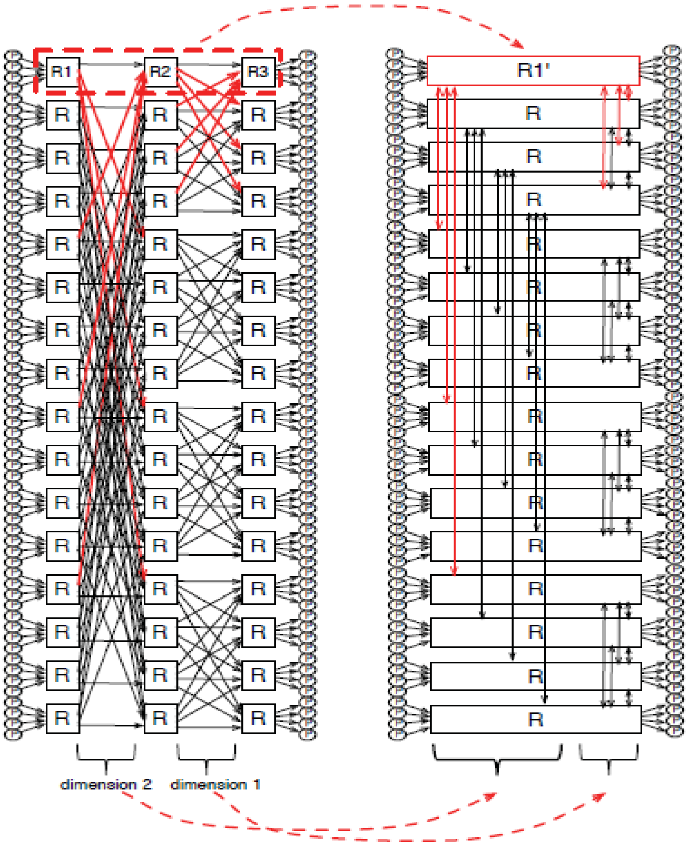



UNIVERSITÄT BERN

# 4. High-Radix Topologies

## 4. Flattened Butterfly Network

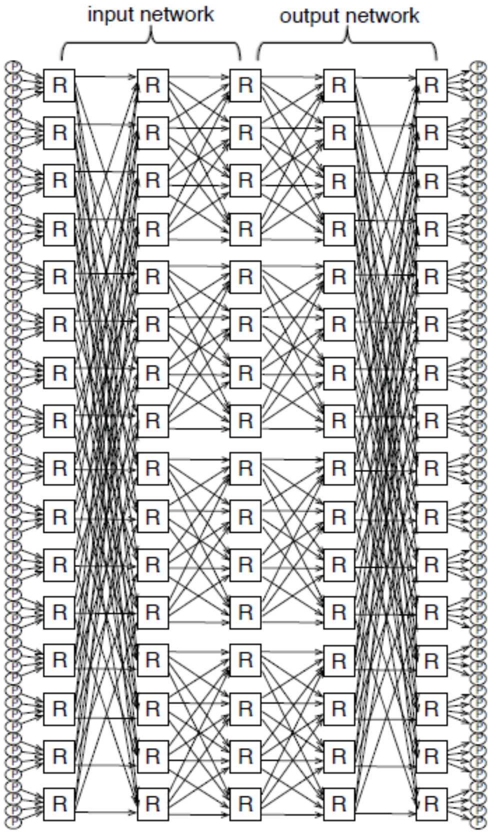
- Removal of intermediate stages
- Combining / flattening of routers in a row into a single router
- Path diversity
- Better performance

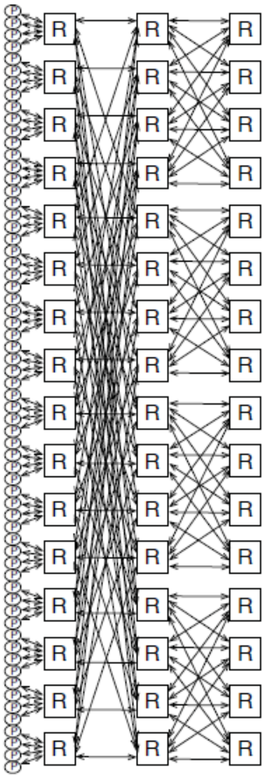


# 4. High-Radix Topologies

## 5. (Folded) Clos Network

- Clos network
  - Multi-stage network with an odd number of stages
  - Combination of two butterfly networks with two stages
    - Load balancing (input network)
    - Traffic routing (output network)
- Folded Clos network
  - Combination of input and output
  - can exploit traffic locality

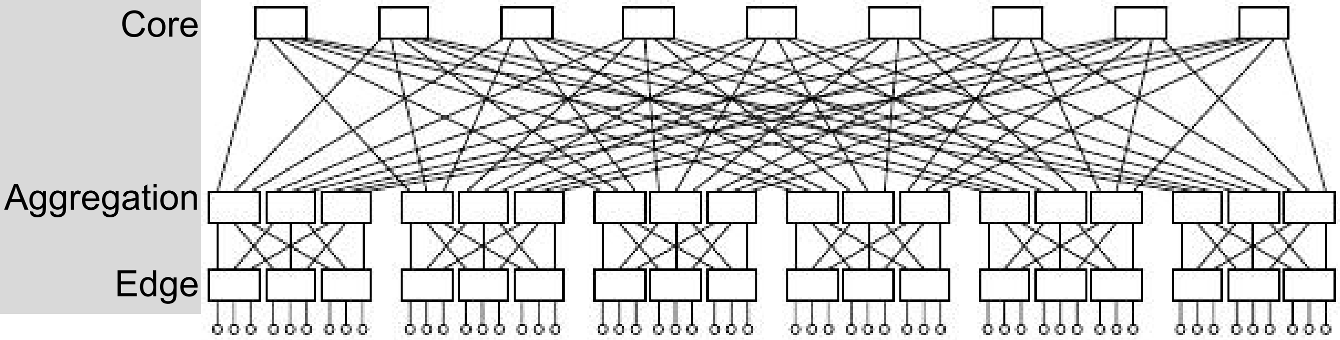


# 4. High-Radix Topologies

## 6. Fat-Tree Based Network

- special type of Clos networks
- can be built using cheap devices with uniform capacity
- Each port supports the same speed as the end host.
- All devices can transmit at line speed, if packets are distributed uniformly along available paths.
- good scalability



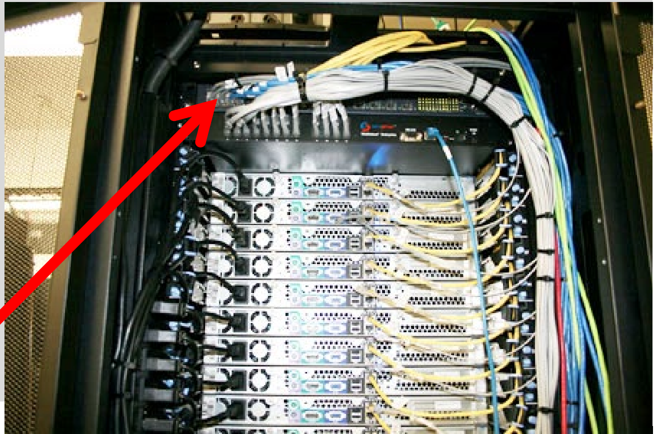
# 5. Data Centre Networks

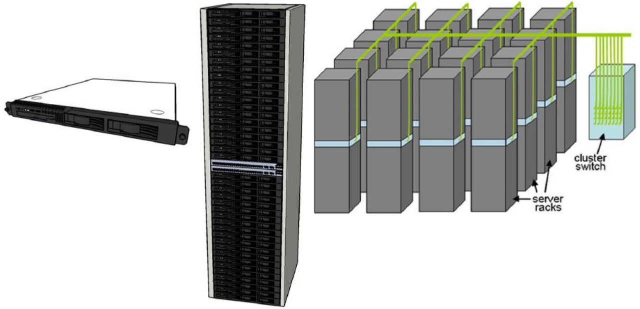
- 10’s - 100’s of thousands of hosts, often closely coupled, in close proximity:
  - E-business (e.g., Amazon)
  - Content servers (e.g., YouTube, Akamai, Apple, Microsoft)
  - Search engines, data mining (e.g., Google)
  - Social networks (e.g., Facebook)
- Challenges
  - Multiple applications, each serving massive numbers of clients
  - Managing / balancing load, avoiding processing, networking, data bottlenecks

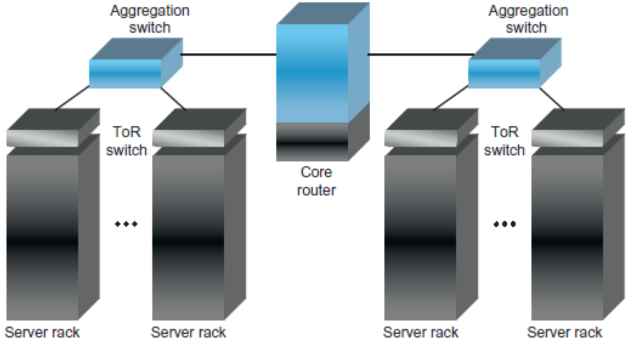
# 5. Data Centre Networks

## 1. Top-of-Rack (TOR) Architecture

- Clustering
  - Hosts are packaged into racks.
  - Racks are allocated and tightly connected to form clusters.
  - A cluster can contain 1000s of hosts.
  - Each cluster is homo-geneous in processor type and speed.
- Rack of servers
  - Commodity servers
  - Top-of-rack switch





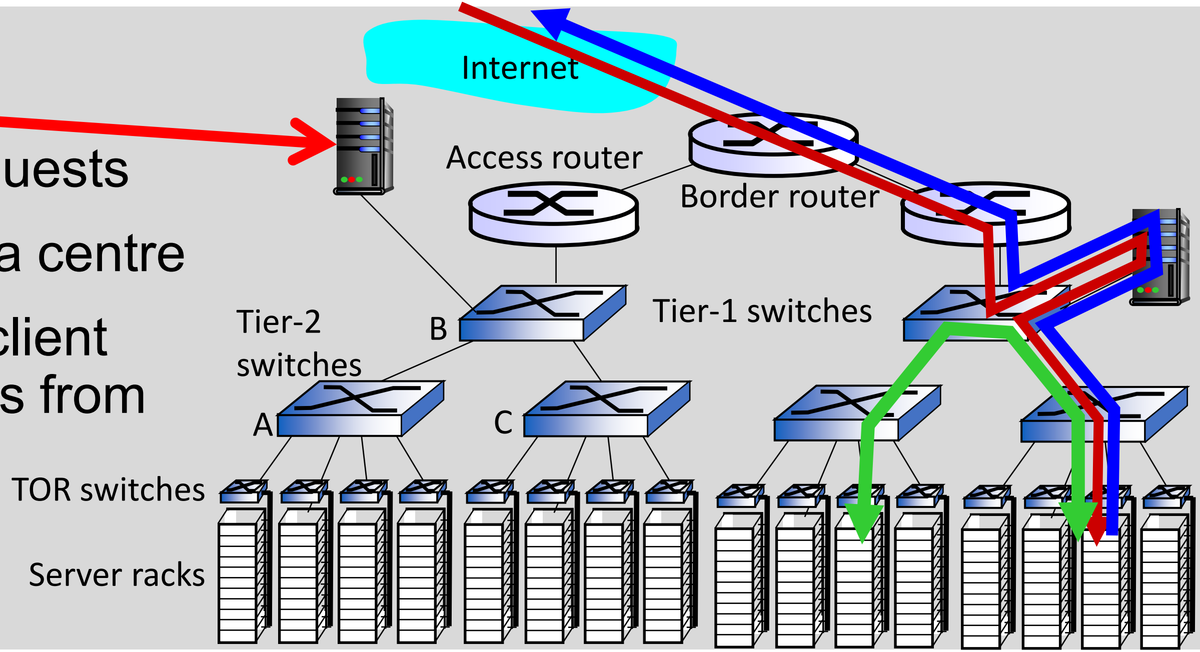



# 5. Data Centre Networks

## 2. Load Balancing

Load Balancer

- receives external client requests
- directs workload within data centre
- returns results to external client (hiding data centre internals from client)



UNIVERSITÄT BERN

# 5. Data Centre Networks

## 3. Data Centre Traffic

- Traffic within a data centre network is often characterized according to flows.
  - Sequences of packets from a source to destination host
- Asymmetric traffic in client-server model
  - Requests are abundant but small in size (client to server).
  - Server-to-client responses are generally big in size.
- Bandwidth provisioning in multi-tier data centres
- Inter-cluster and intra-cluster traffic
- Goal: predictable latency and bandwidth characteristics across varying traffic patterns

UNIVERSITÄT BERN

# 5. Data Centre Networks

## 4.1 Traffic Engineering Challenges

- Scale
  - Many switches, hosts, and VMs
- Churn
  - Component failures
  - VM migration
- Traffic characteristics
  - High traffic volume
  - Volatile, unpredictable traffic patterns
- Performance requirements
  - Delay-sensitive applications
  - Resource isolation between tenants

# 5. Data Centre Networks

## 4.2 Traffic Engineering Opportunities

- Efficient network
  - Low propagation delay and high capacity
- Specialized topology
  - Fat tree, Clos network, etc.
  - Opportunities for hierarchical addressing
- Control over both network and hosts
  - Joint optimization of routing and server placement
  - Can move network functionality into the end host
- Flexible movement of workload
  - Services replicated at multiple servers and data centres
  - VM migration

# 5. Data Centre Networks

## 5. Layer 2 vs. Layer 3 Forwarding

**Ethernet Switching (Layer 2)**

- Cheaper switch equipment
- Fixed addresses and auto-configuration
- Seamless mobility, migration, and failover

**IP Routing (Layer 3)**

- Scalability through hierarchical addressing
- Efficiency through shortest-path routing
- Multipath routing through equal-cost multipath

UNIVERSITÄT BERN

# 5. Data Centre Networks

## 6.1 Objectives of Routing Algorithms

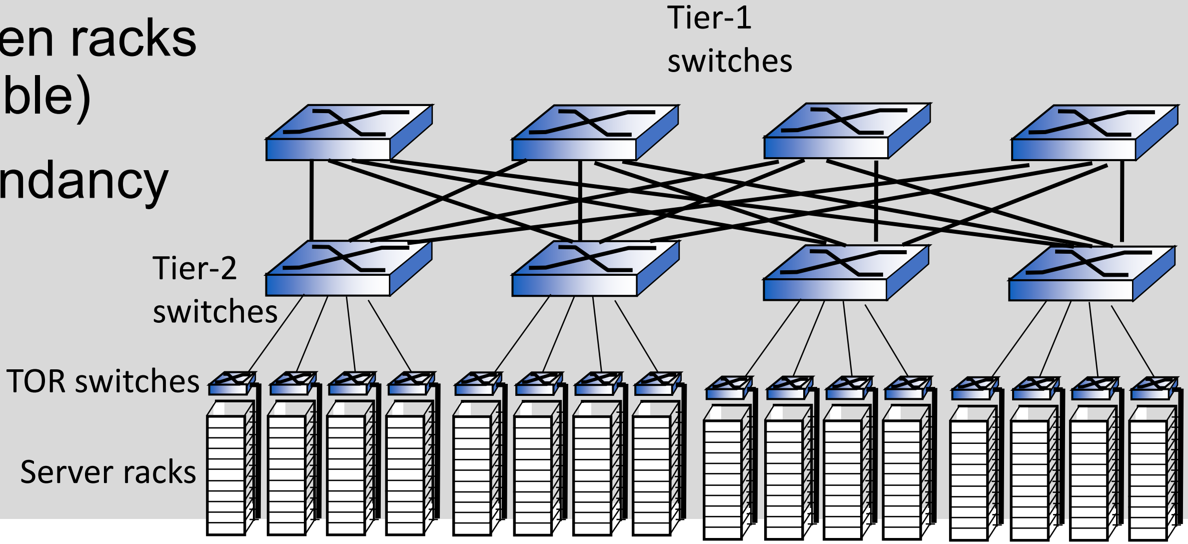
- Path diversity
  - exploits the topology, which can include both minimal and non-minimal paths
- Load balancing
  - Network traffic is routed across channels to achieve high overall throughput.
- Complexity
  - To minimize the impact on packet latency and load imbalance, which may result from a fault in a network, the routing algorithm must be able to be implemented efficiently.

# 5. Data Centre Networks

## 6.2 Data Centre Network Topologies

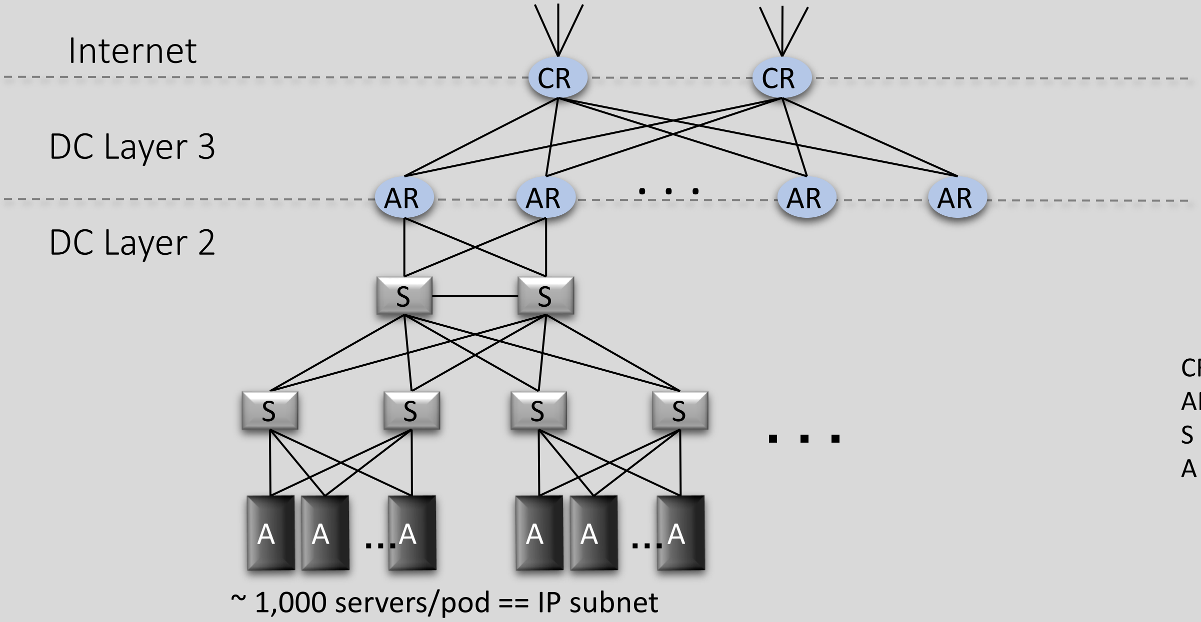
Rich interconnection among switches, racks:

- increased throughput between racks (multiple routing paths possible)
- increased reliability via redundancy



# 5. Data Centre Networks

## 6.3 Data Centre Routing




CR = Core Router (L3)
AR = Access Router (L3)
S = Ethernet Switch (L2)
A = Rack of app. servers

# Thanks

# for Your Attention

UNIVERSITÄT BERN

**Prof. Dr. Torsten Braun, Institut für Informatik**

Bern, 19.10.2020