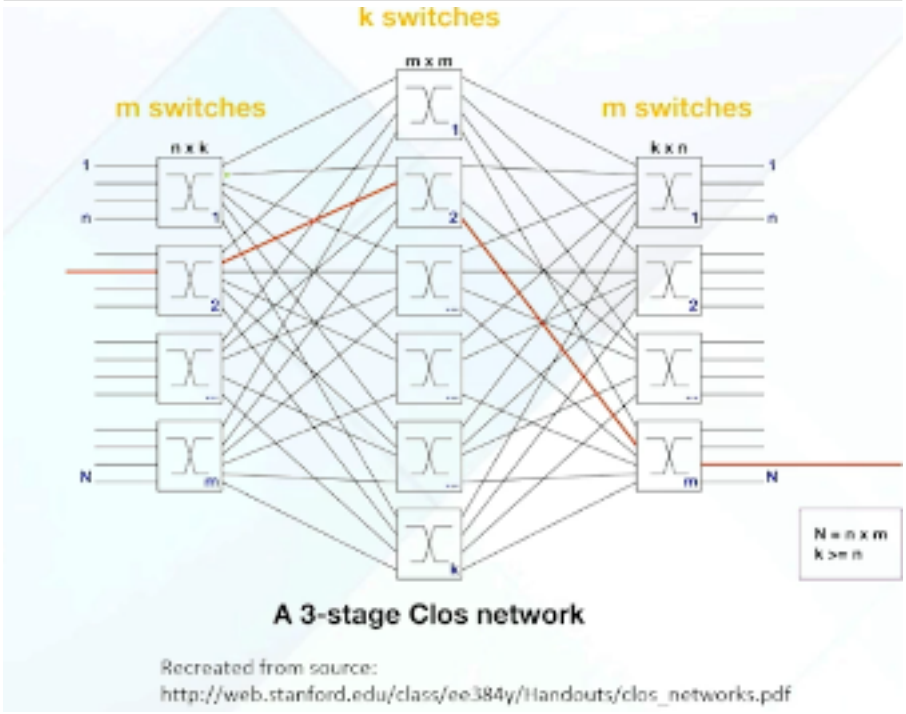
Evolution of Data Center Networks

Introduction

1. Ideas that originated from one domain find an application in a completely different domain
2. Ideas that once seemed farfetched at a given point in time find a compelling use case in the future

Data Center Network Design Considerations

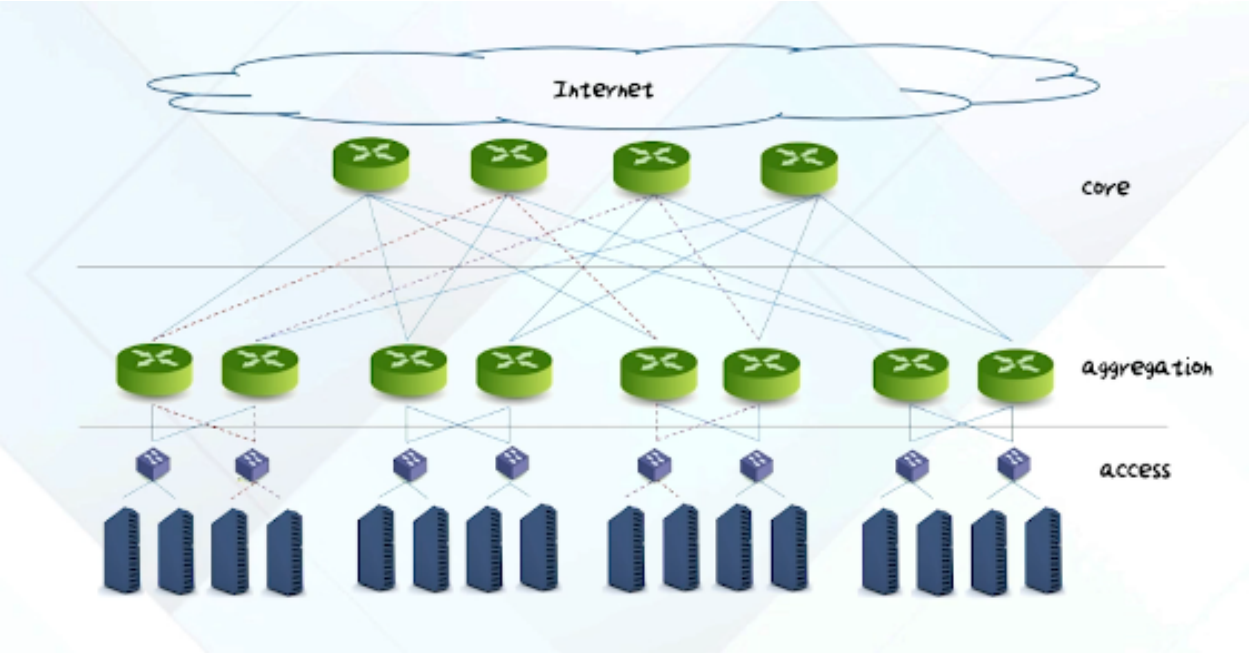
1. Data centers provid reliable and scalable computing infrastructure for massive Internet sources
    - Physically located in various (often remote) geographical areas optimizing for energy consumption and availbility of economic power
2. Data center networks
    - Network is one of the main components of the computational resources in a data center
    - Connects all the servers inside a data center and provides connectivity to the Internet outside
3. Design considerations
    - Data centers host ont he order of 100s of thousands of servers
    - A single app may run on thousands of servers
    - Servers have to communicate with one another
        - Need high throughput, low latency communication among the servers
    - Data center is multi-tenancy
        - Multiple independent apps running in the computational resources
    - Need uniform network capacity for supporting all the apps
        - Server-server communication should be bound only by the interface speeds independent of which servers are hosting a particular app
    - Need performance isolation for the apps
        - Traffic of one service should not affect another
    - Need layer-2 semantics as though the servers were on a LAN so far as each app is concerned
4. Design choices
    - Given the scale, data center networks have to be cost effective while meeting the design considerations
    - Should data center networks use commodity or special switching fabric?
    - What should be the network topology to meet the design considerations and be cost effective?
        - Need a crossbar interconnection network but that would not be cost effective
    - Same dilemma while building telephone switches back in the early days of telephony
        - Led to Clos network as the scalable interconnection network for meeting bandwidth requirements for end devices with commodity switches
    - Data center networks have taken the same approach
5. Advantages of Clos Network
    - What was good in the 1950s for telephony is good today for data centers
    - Use small sized commodity switches to connect large number of inputs to large number of outputs
        - Exactly 1 connection between ingress and middle stage
        - Exactly 1 connection between middle and egress stage
    - Constant latency between any two hosts
        - Number of hops from source to destination
    - If k >= n then Clos can be non-blocking like a crossbar switch
    - Redundant paths between any two hosts



Clos Network

Data Center Networks Design

1. Data center networks are an adaptation of Clos network
   - Use commodity Ethernet switches and routers
   - Two- or three-level trees of switches or routers
   - A three-tiered design
     - Core layer at the root of the tree; connects to the Internet via Layer 3 routers
     - Aggregation layer in the middle; transition from Layer 2 switched layer to Layer 3 routed core layer
     - Edge layer (aka access layer) at the leaves of the tree; Layer 2 switches connect to servers
   - A two-tiered design (less prevalent)
     - Core
     - Edge
2. Fat tree: A special form of Clos
   - Same aggregate bandwidth at each layer
   - Identical bandwidth at any bisection
   - Each port same speed as end host
   - All devices can transmit at line speed if packets are distributed uniformly along available paths
   - Scalability with k-ary Fat tree
     - k-port switch spports kˆ3/4 servers



Fat Tree Routing

Active Networks and SDN

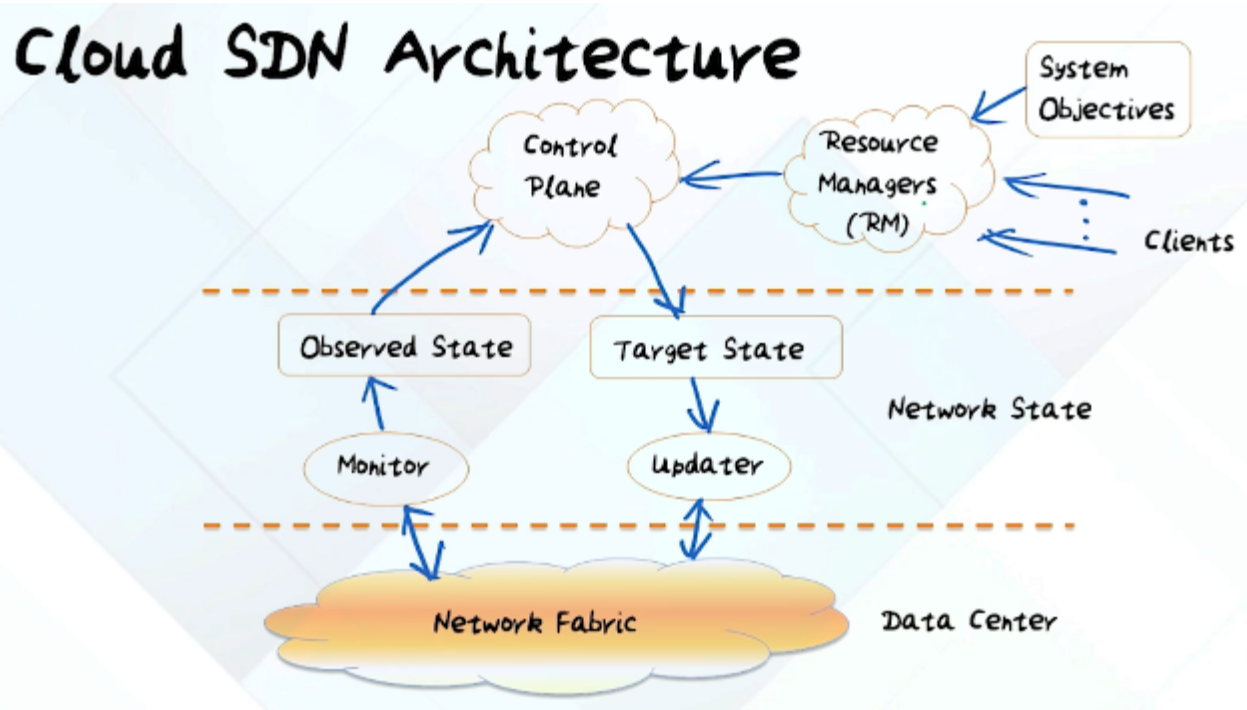
1. Meeting Other Critical Design Considerations
   - Clos network topology and its variants allow meeting the goal of scalability while being cost effective
   - However, intelligent routing decision is key to ensuring performance between end hosts
   - Statically deciding the routing between any two servers will result in network congestion and hot spots despite the redundant paths available in the Clos network
   - This calls for an active network where the routing decisions are dynamic
2. Back to the drawing board
   - Wanting network routing to be active is not a new idea
     - Traditionally routing decisions in the Internet are static decided by lookup tables in the routers that are updated periodically
     - Packet arrives, router looks up the table and sends it to the next hop based on the destination field and the corresponding table entry
   - Active Networks was a vision proposed by Tennenhouse and Wetherall in the mid 90s
     - Idea is for packets to carry code that can be executed in the routers en route from source to destination
     - Routers could make dynamic routing decisions
     - Way ahead of its time
   - Principal shortcomings of active networks
     - Potential vulnerabilities due to protection threats and resource management threats of the routers
     - Need buy in from router vendors since it "opens" up the network
     - Software routing (by executing code at the routers) cannot match "line" speeds of hardware routers
3. Resurrection of Active Networks as SDN
   - Wide area Internet (WAN traffic engineering)
     - Tech giants such as Google and Facebook want to build their own distribution networks
     - Control over how packets are routed possibly giving preferential treatment to some flows over

others
    - Violates "net neutrality" principle but that is still a hot political debate
- Cloud computing (per customer virtual networks)
    - Service providers want to support dynamic routing decisions to ensure performance isolation for network traffic with multi-tenancy
- Chip sets for assmebling routers are available which makes it easy to "build your own router" skirting the need for "buy in" form router vendors

Software-Defined Networking

1. Software-defined Networking
    - Eliminates the key drawbacks of the original Active Networks vision
    - Rule-based routing as opposed to executing arbitrary code in the routers
        - Removes protection and resource management threats
    - Routing for flows set centrally once so that each router can work at "line" speeds
        - Removes poor performance concern (of software routing)
    - Real breakthrough is separating control plane (rules setting for routing) from the data plane (transport of paylod in the packet)
    - Control plane is software controlled from a central entity
    - Well-defined APIs to program the control plane
        - OpenFlow
2. SDN Architecture
    - Elements of the architecture
        - Switch fabric implementing the desired network topology (e.g., Clos)
        - Central controller setting routing rules for the switches per traffic flow
    - Central controllers
        - Programmed using the APIs
        - State maintenance commensurate with control application directives and switch events
        - Set forwarding entries in the switches
    - Switch hardware
        - Simple rule-based traffic forwarding as instructed by the central controller
3. WAN and SDN
    - The focus is network-wide traffic engineering at-scale
        - Meet network-wide performance goals
        - Infer traffic patterns and centralize control
4. Cloud and SDN
    - Cloud provides the much needed "killer app" for SDN
    - "Perfect storm" of need, control, and personnel
    - Need
        - Network traffic isolation for each customer
        - Network performance isolation for each customer
    - Control
        - Complete control over the network fabric for the cloud provider
        - Data center network (and possibly inter-datacenter network fabric) is not open to all Internet traffic as is the case with WAN
    - Personnel
        - System developers (architecture, system software, and network gurus) all aligned on a common purpose
5. Cloud SDN Design Principles
    - Resource Managers field client requests
    - System objectives
        - SLAs with clients
    - RMs funnel requests to the control plane
        - Decide "target state"

- Control drives the network fabric
  - Monitor and record "observed state"
  - Update
  - Goal: Drive to "target state"
- Network fabric
  - Stateless switches
  - Carry out controller's strategy for packet forwarding
- Costs are amortized over life of application, giving a performance advantage



Cloud SDN Architecture

Traffic Engineering

1. Internet Traffic Patterns
   - Multiple independent data flows
   - "Well-behaved" (e.g., TCP) and "not so well-behaved" (e.g., UDP) flows injecting packets into the networks
     - Not to mention malicious traffic flow which may be indistinguishable at the packet level until it hits a host and detected at higher levels of the protocol stack
   - Packet level issues
     - Out of order delivery
     - Packet loss, mostly due to dropped packets due to congestion; when it happens, it happens in bursts... snowballing effect
     - Packet corruption (very rare, less than 0.1%)
2. Network Congestion
   - Offered laod to the network exceeds network capacity
   - Problem is common to WAN and the Cloud
     - Recall Cloud network fabric uses the same commodity switching gear as WAN
     - TCP is the dominant transport protocol in both; Sender uses a dynamically sized window to decide when to inject new packets into the network

3. General Strategies for Congestion Control
    - Principle of conservation of packets
        - Equilibrium state: Stable set of data packets in transit
        - Inject a new packet only when an old packet leaves
    - Metered on-ramps to highways
        - Dynamically change the metering rate commensurate with observed traffic congestion
    - Problems with conservation of packets
        - Sender injects packets before "old" leaves
        - Equilibrium state never reached due to resource constraints along the path
4. Congestion Avoidance
    - Slow start
        - Sender starts with a small send window
        - Increase the send window by 1 on each ACK ("additive increase")
        - Problem: Potentially increase the send window exponentially despite "slow" start
    - Conservation at equilibrium
        - Need good estimate of round trip time (RTT) to set retransmission timer
        - Upon retransmission, halve the send window size (exponential backoff) AKA "multiplicative decrease"
        - Retransmission: Don't receive an ACK so assume packet was lost and send it again
    - Congestion avoidance
        - Combine slow start and conservation at equlibrium
        - Current window size = send_window / 2 upon retransmission
        - Current window size = send_window + 1 upon ACK

Cloud Network Traffic Engineering

1. Data Center Traffic Engineering Challenges
    - Scale of data center: 100s of thousands of servers -> plethora of switches in the routing fabric
    - Change is the order of the day
        - With so many components in a data center, failure is not an "if" questions... it is a "when" question
        - VM migration for load balancing and failure tolerance
    - Application characteristics and their traffic patterns
    - Implications of performance loss
2. Data Center Networks
    - Flow
        - Data center applications generate diverse mix of short and long flows
    - Requirements
        - Applications tend to have the partition/aggregate workflow (low latency for short flows, high burst tolerance for short flows)
        - Applications need internal state maintenance for freshness of data (high throughput for long flows)
        - Partition/aggregate means splitting a problem across systems then combining the results
    - Implications
        - Congestions can lead to severe performance penalty and loss of revenue
        - Bad performance means people won't use some data center
    - One the bright side
        - Round trip times can be more accurately estimated
        - Congestion avoidance schemes can be tailored to take advantage of the data center ecosystem

Conclusion

1. Covered origin of software-defined networks and network technologies that are prevalent in modern data centers (traced back to the 1950s)

2. Data centers use administrative control to solve the traffic engineering challenges that are specific to data center networks