

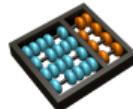
Comunicação em Datacenters

Gerência de Redes

Leandro Souza da Silva
Luís Felipe Mattos

IC - Unicamp

06 de Dezembro de 2016



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

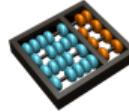
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

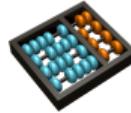
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta



Visão Geral de um Data Center.

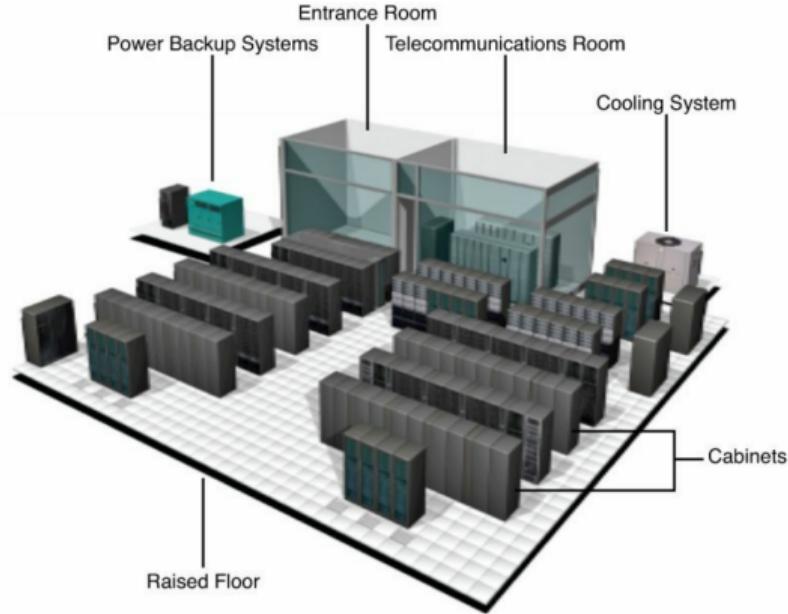
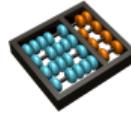
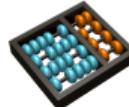


Figure : Data Center



Introdução

- Com o crescimento da computação em nuvem, os datacenters passaram a receber funções novas.
- Certas aplicações necessitam de certos requisitos:
 - ▶ Escalabilidade
 - ▶ Tolerância a Falhas
 - ▶ Latência
 - ▶ Capacidade da Rede
 - ▶ Virtualização



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

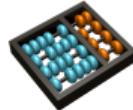
4 Protocolos

- Roteamento
- Transporte

5 Tendências

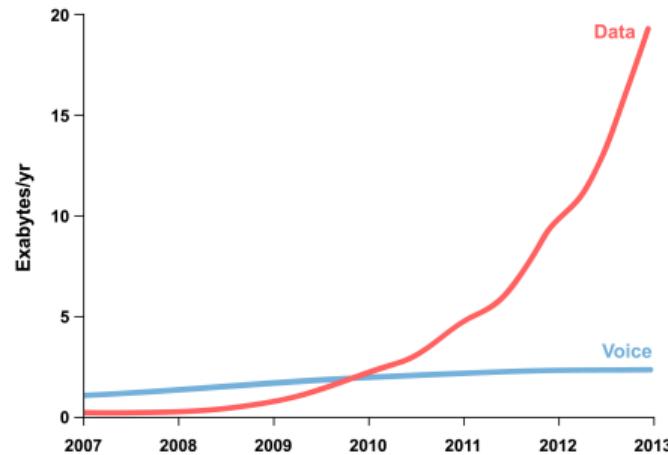
6 Conclusão

7 Pergunta

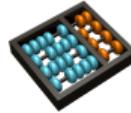


Motivação

O consumo de dados pelos usuários está crescendo exponencialmente a cada ano.

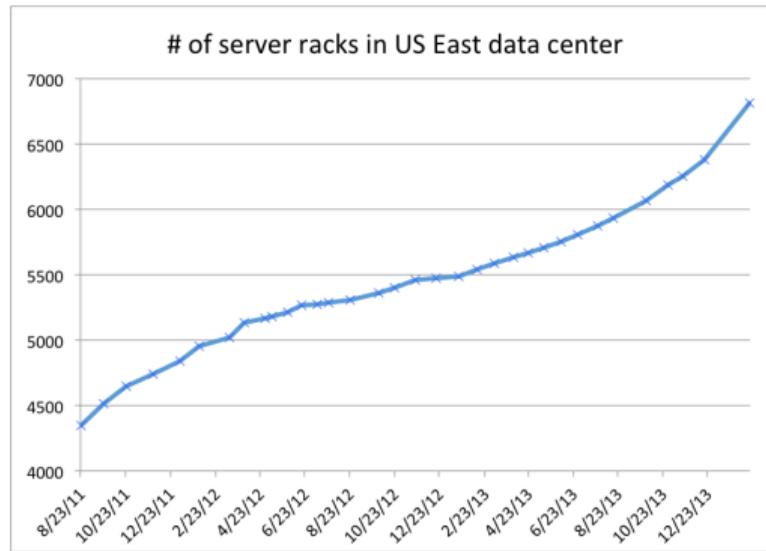


Source: The Cloud Begins With Coal , Ericsson Mobility Report, June 2013
Figure : Consumo de dados e voz



Motivação

Por causa disso, o número de servidores em Data Centers deve crescer exponencialmente para acompanhar a demanda, o que traz dificuldades em desenvolver redes eficientes e de baixo custo.



Source: <https://huanliu.wordpress.com/category/cloud/>
Figure : Número de servidores racks

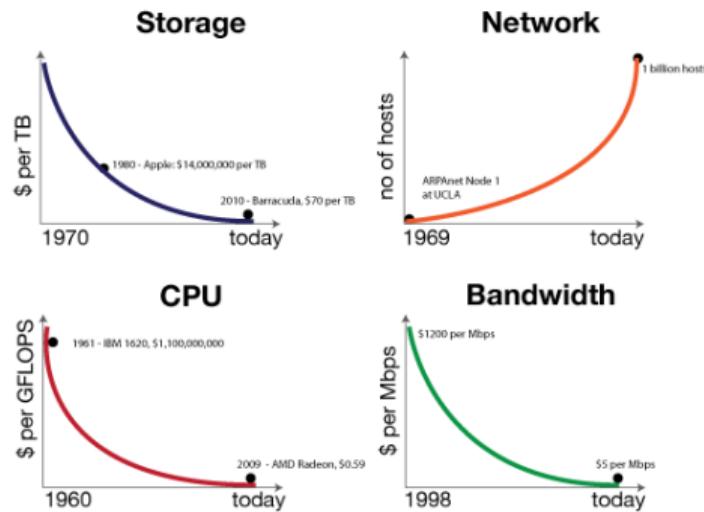
Motivação

Disponibilidade de dados e segurança se tornaram aplicações críticas.

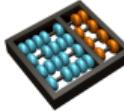


Motivação

Por outro lado, a criação de novas tecnologias faz com que o custo dos componentes seja cada vez menor.

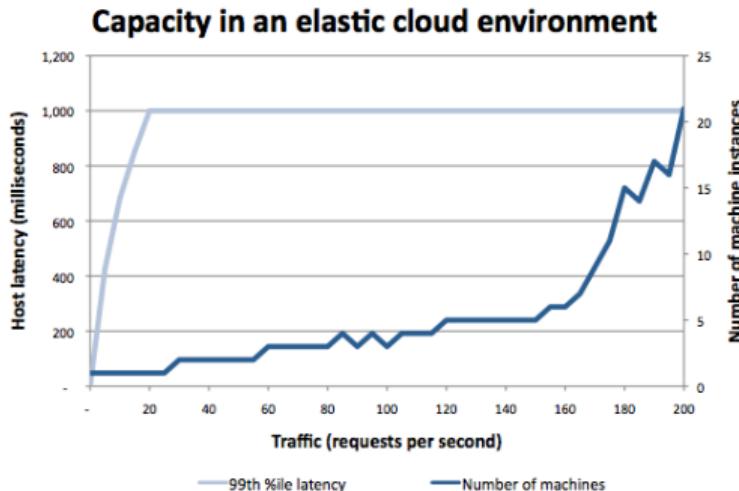


Source: <http://radar.oreilly.com/2011/08/building-data-startups.html>
Figure : Custo de tecnologias



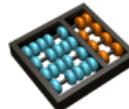
Motivação

Apesar disso, surge uma dificuldade cada vez maior de criar redes escaláveis exponencialmente sem que haja perda de eficiência.



Source: <http://www.bitcurrent.com/the-clouds-most-important-equation/>

Figure : Capacidade elástica em ambiente de nuvem



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

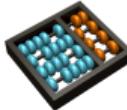
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta



Topologias Tradicionais

- Baseadas em árvores.
 - ▶ Basic Tree
 - ▶ Fat-Tree
 - ▶ VL2

- Recursivas.
 - ▶ Dcell
 - ▶ Bcube
 - ▶ FiConn
 - ▶ FlatNet
 - ▶ SprintNet



Basic Tree

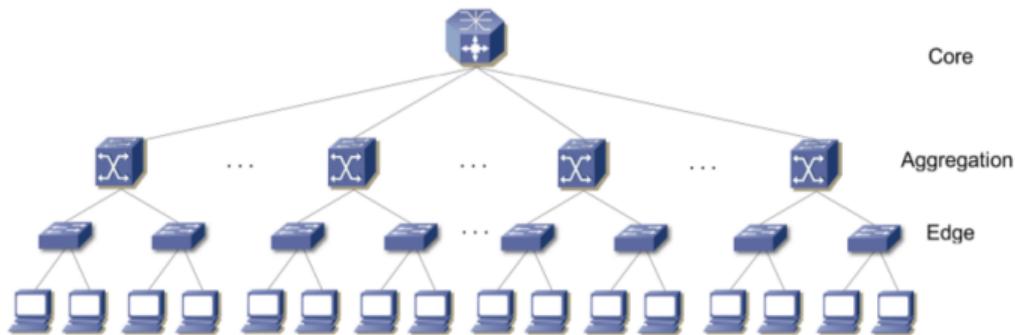
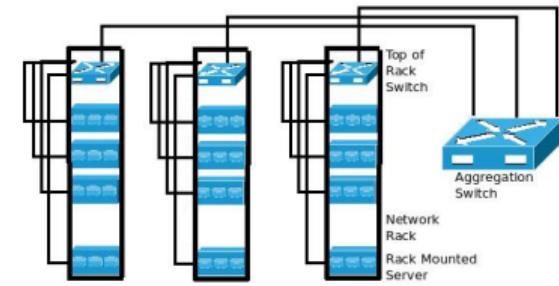
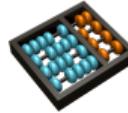


FIGURE 1. A traditional 3-level tree-based data center network topology.

Top-Of-Rack (TOR) - Network Connectivity Architecture



- Crescimento do Oversubscription na direção do Core da rede.
- Oversubscription é a multiplexação dos recursos de banda, para economizar enlaces e equipamentos, sem reduzir o desempenho da rede.
- O Oversubscription total é a soma do Oversubscription no domínio de acesso dos servidores + Oversubscription da agregação



Exemplo:

Domínio de acesso dos servidores.

Switch 48 Portas – 1 Gbps

20 Gbps para uplink – servidores

Oversubscription = $(48G/20G) = 2.4$

$1\text{Gbps}/2.4 = 416\text{Mbps}$

Oversubscription Total = $2.4 + 1.5 = 3.9$

Velocidade Max de Transmissão dos Servidores = $(416\text{Mbps}/1.5) = 277\text{Mbps}$. Se Todos os servidores resolverem transmitir no máximo da capacidade da interface de 1Gbps, então haverá congestionamento e perda de pacotes.

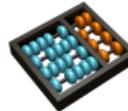
Agregação.

20 Portas – 10 Gbps

8 Para se conectar ao Core

12 Para se conectar ao switch Tor

Oversubscription = $(120G/80G) = 1.5$



Fat Tree, baseada na rede CLOS

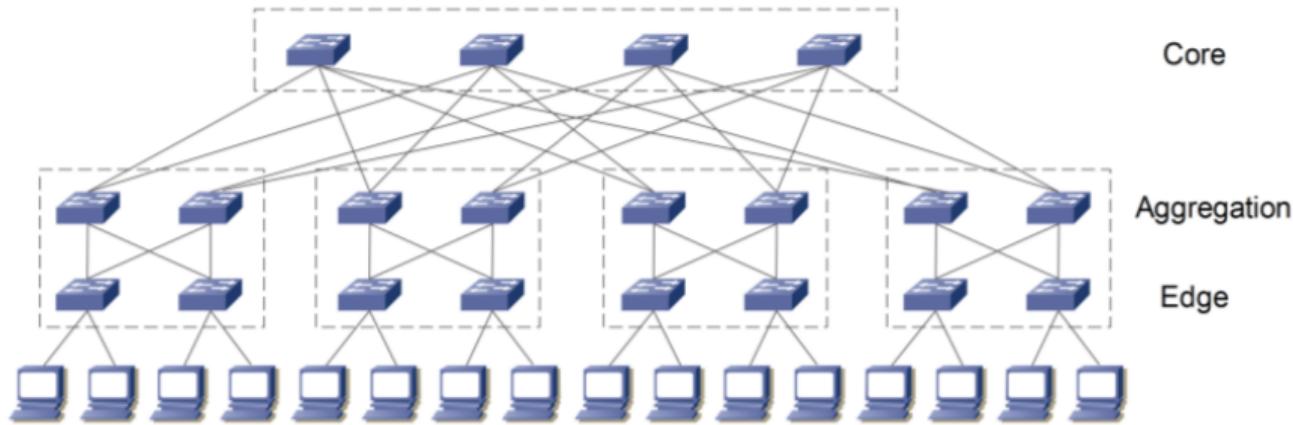
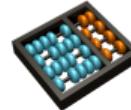


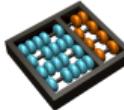
FIGURE 2. A simple 3-level Fat-Tree topology.

Fat Tree, baseada na rede CLOS

- Criada por Charles Clos do MIT.
- É uma estrutura rearranjável não bloqueante.
- Fornece uma relação de Oversubscription de 1:1 a todos os servidores.
- Suporta $n^3/4$ servidores, onde n é o número de portas do switch.
- No entanto, a complexidade da fiação é $O(n^3)$, que é um desafio sério.



- Switches em uma topologia de rede CLOS
- Usa o VLB (Valiant Load Balancing) para distribuir tráfego entre os caminhos da rede
- Usa o protocolo ARP (Address Resolution Protocol) para que seja escalável para um número grande de servidores



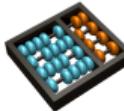
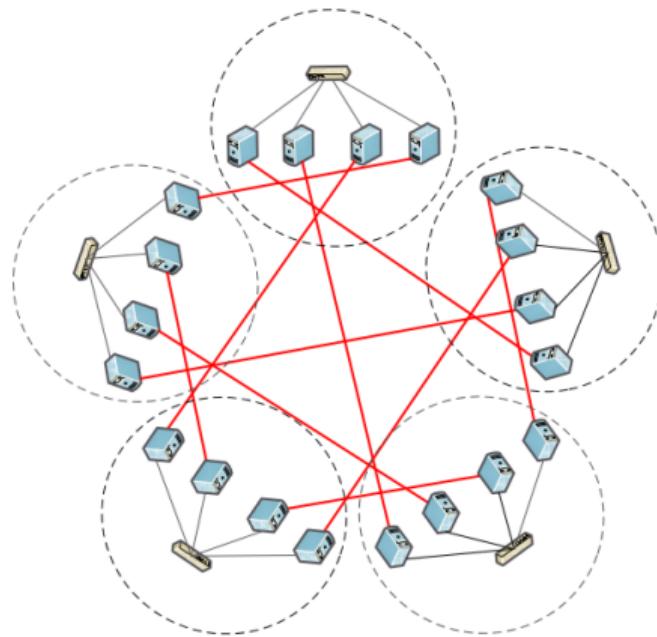
Topologias Tradicionais

Topologias recursivas



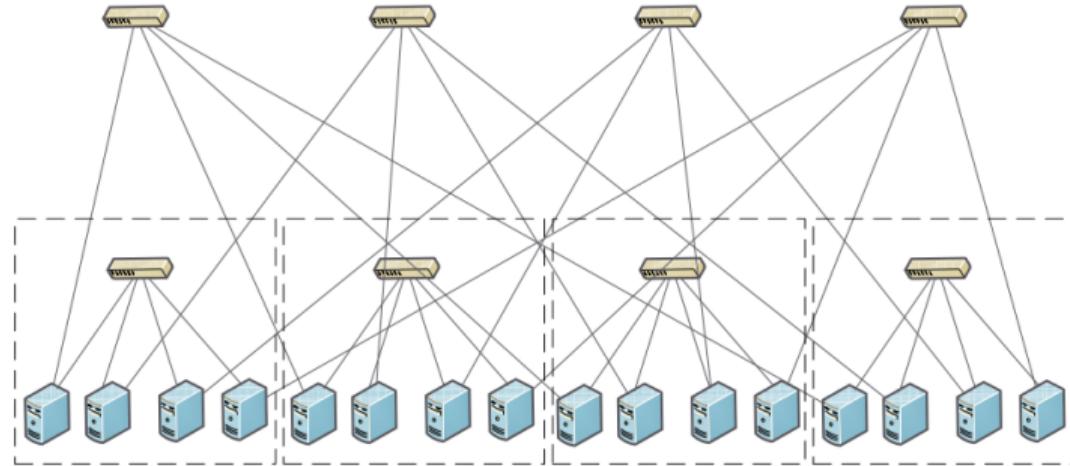
Topologias Recursivas: Dcell

Baseada em células interligadas entre servidores



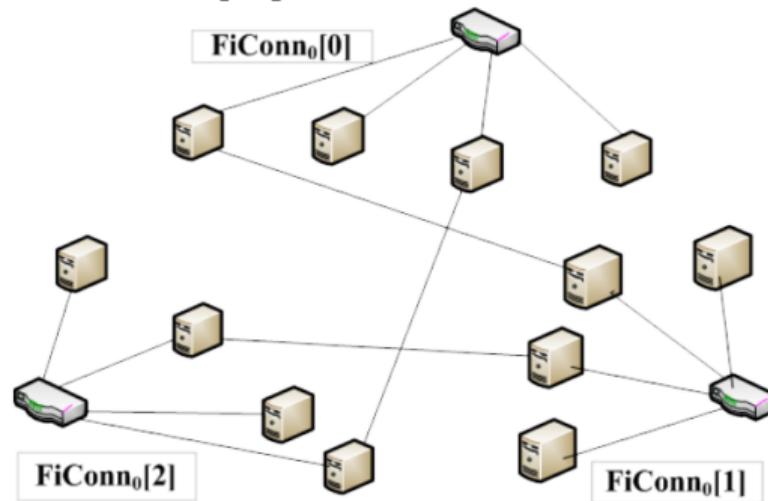
Topologias Recursivas: Bcube

Baseada em células interligadas entre switches



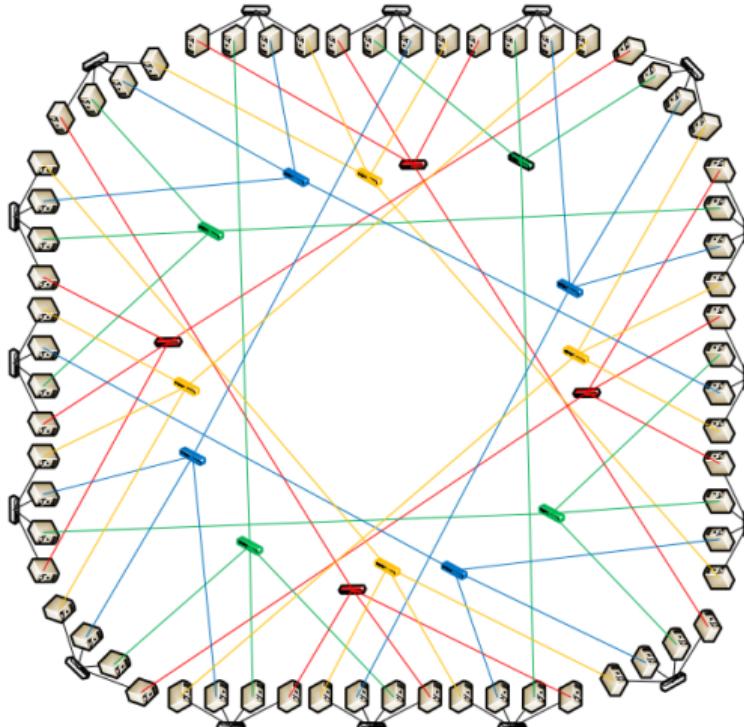
Topologias Recursivas: FiConn

Semelhante à Dcell, mas o grau de cada célula é sempre 2



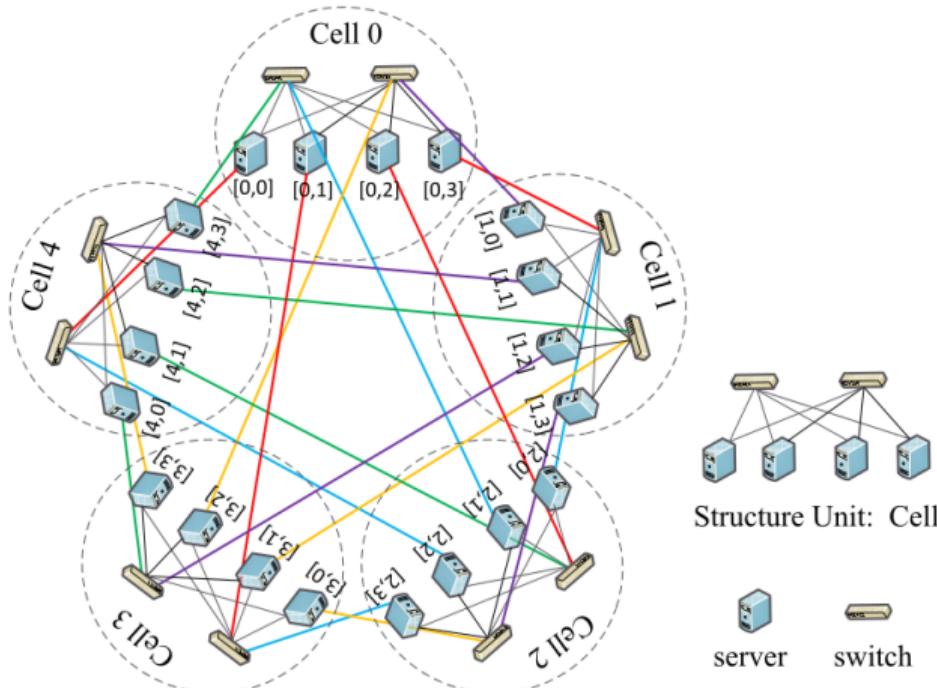
Topologias Recursivas: FlatNet

Semelhante ao BCube, porém é mais escalável



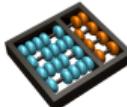
Topologias Recursivas: SprintNet

Semelhante à DCell, porém as células são compostas por 4 servidores e 2 switches de 6 portas



Comparação

	Fat Tree (3 layers)	VL2 (3 layers)	DCell (2 layers)	BCube (2 layers)	FlatNet (2 layers)	SprintNet (2 layers)
Servers Number	$\frac{n^3}{4}$	$\frac{(n-2)n^2}{4}$	$n(n+1)$	n^2	n^3	$(\frac{c}{c+1})^2 n^2 + \frac{c}{c+1} n$
Links Number per Server	$\frac{3n^3}{4}$	$\frac{(n+2)n^2}{4}$	$\frac{3n(n+1)}{2}$	$2n^2$	$2n^3$	$\frac{c^2 n^2}{c+1} + cn$
Switches Number per Server	$\frac{5n^2}{4}$	$\frac{n^2}{4} + \frac{3n}{2}$	$n+1$	$2n$	$2n^2$	$\frac{c^2}{c+1} n + c$
Bisection Bandwidth per Server	$\frac{n^3}{8}$	$\frac{n^2}{4}$	$\frac{n^2}{4} + \frac{n}{2}$	$\frac{n^2}{2}$	$\frac{n^3}{4}$	$\frac{c^2 n^2}{2(c+1)^2} + cn$
Network Diameter	6	6	5	4	8	4



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

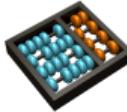
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

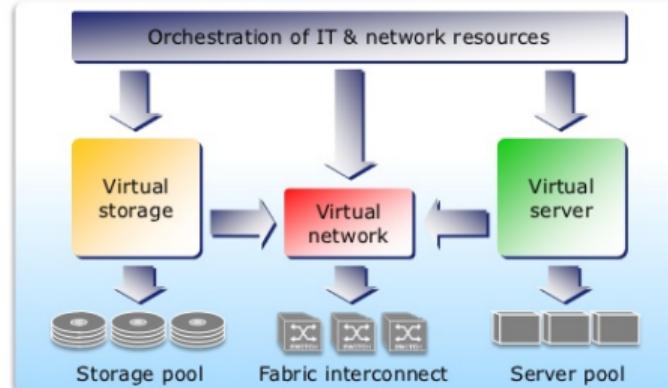
7 Pergunta



SDN em Datacenters

Com o avanço do SDN, a ideia mais básica é definir servidores virtualizados e criar uma rede virtualizada

SDN Inside The Data Center

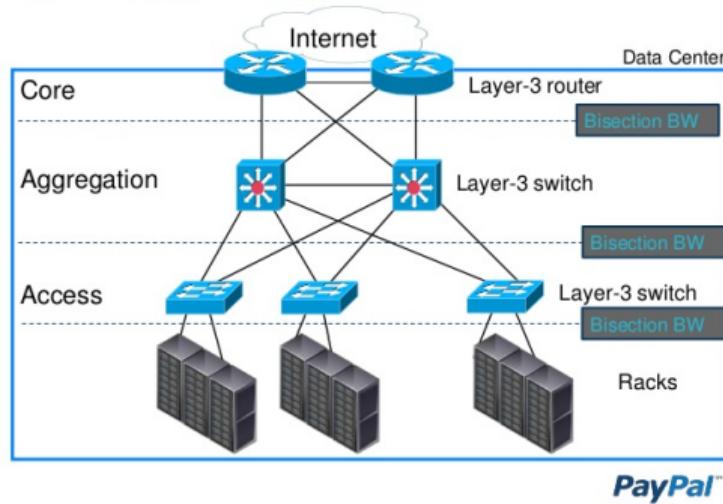


SDN adds missing piece to the virtualization puzzle: Network virtualization.

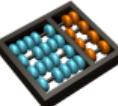
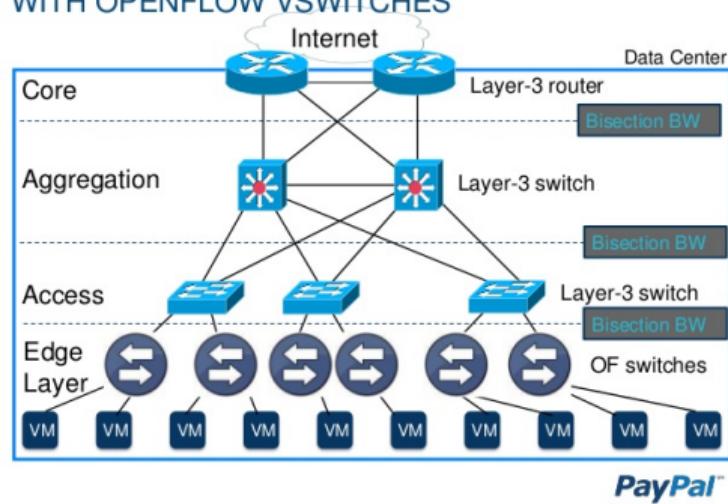
SDN em Datacenters

Esta técnica já é utilizada atualmente (PayPal por exemplo)

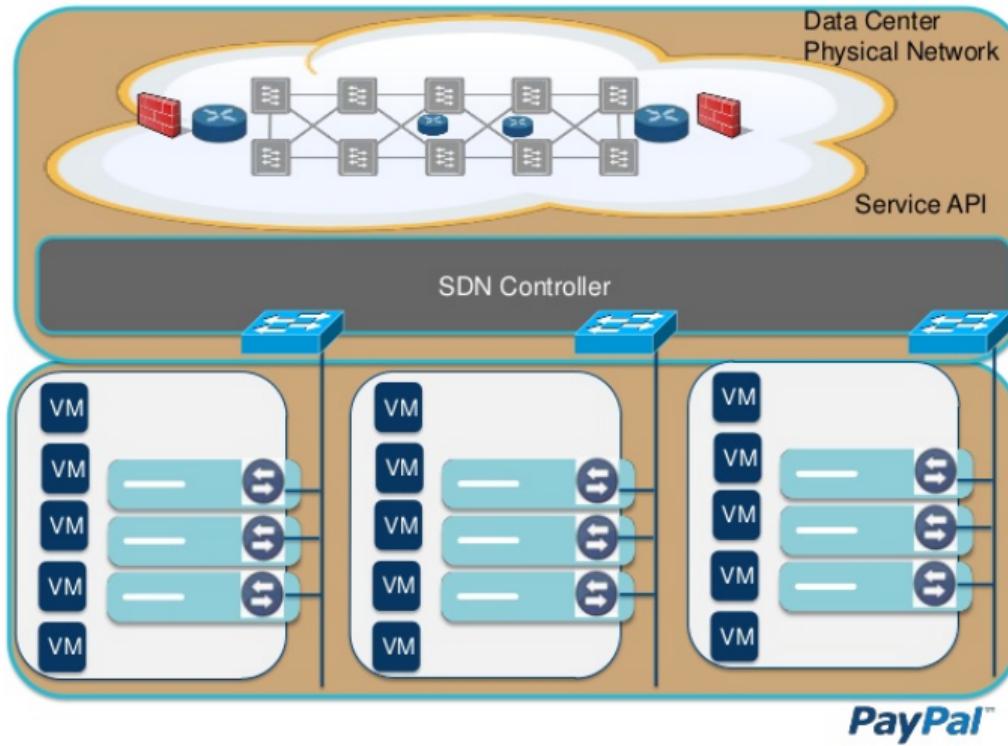
DATACENTER ARCHITECTURE



DATACENTER ARCHITECTURE WITH OPENFLOW VSWITCHES



SDN em Datacenters



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

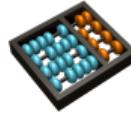
4 Protocolos

- Roteamento
- Transporte

5 Tendências

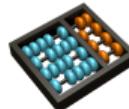
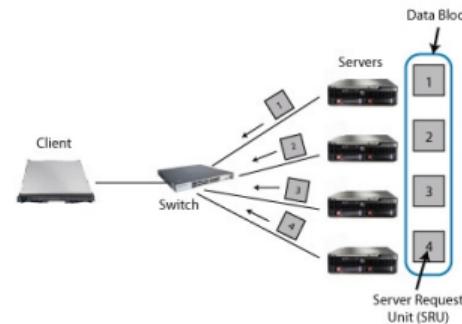
6 Conclusão

7 Pergunta

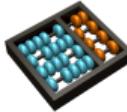


Datacenters têm alguns requisitos básicos, mas que causam algumas dificuldades:

- RTTs precisam ser da ordem de microsegundos
- Poucas multiplexações
- Perda de pacotes baixa
- Lidar com Incast



- Esquemas para TE
 - ▶ ECMP
 - ▶ VL2
 - ▶ DARD

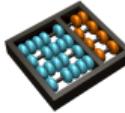


Traffic Engineering (TE)

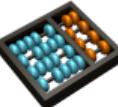
- Embora a maioria dos esquemas básicos de roteamento busquem rotas entre dois servidores com latência curta, um encaminhamento mais sofisticado exige maior consideração e otimização da latência, confiabilidade, throughput, energia e etc. Esse tipo de otimização é conhecido como problema de engenharia de tráfego (TE).
- Existem poucos mecanismos para a otimização de roteamento DCN hoje em dia.



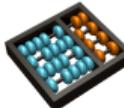
- Equal-Cost-MultiPath (ECMP).
- Abordagem distribuída de seleção de caminho a nível de fluxo.
- Pode ser configurado com vários saltos seguintes para um determinado destino.
- Encaminha um pacote de acordo com um hash de campos do cabeçalho do pacote.
- Divide o tráfego para cada destino em vários caminhos.
- Grandes fluxos podem colidir em seus valores de hash e congestionar uma porta de saída.
- É um algoritmo de roteamento utilizado para balanceamento de carga.



- Mecanismo de seleção de caminho distribuído.
- Coloca a lógica de seleção nos switches de borda.
- switch da borda encaminha primeiro um fluxo a um switch do núcleo selecionado aleatoriamente.
- O switch do núcleo encaminha o fluxo para o destino.



- Distributed Adaptive Routing for Datacenter Networks (DARD).
- Seleção de caminho adaptativo distribuído.
- Difere do ECMP e VL2 em dois aspectos.
- Primeiro, seu algoritmo de seleção de caminho é sensível à carga.
- Se vários fluxos colidem no mesmo caminho, o algoritmo deslocará os fluxos do caminho colidido para os caminhos mais ligeiramente carregados.



- Em segundo lugar, coloca a lógica de seleção de caminho em sistemas finais em vez de em switches, para facilitar a implantação.
- os switches são habilitados para usar OpenFlow.



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

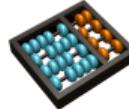
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta

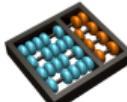


- Deadline-Agnostic

- ▶ DCTCP
- ▶ MPTCP
- ▶ ICTCP

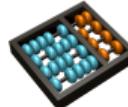
- Deadline-Aware

- ▶ D^3
- ▶ D^2 TCP
- ▶ DeTail
- ▶ PDQ



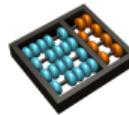
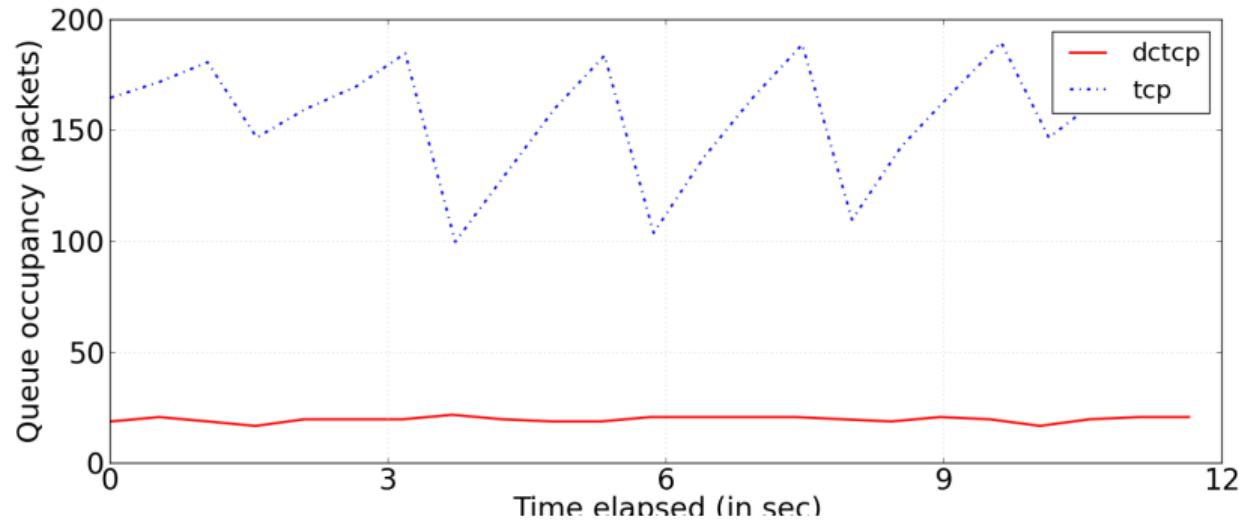
Deadline-Agnostic: *DCTCP*

- Buffers menores do que os do TCP tradicional
- No switch, quando a ocupância da fila ultrapassa o limiar K , marca os pacotes
- O receptor envia um ACK para cada pacote marcado, com a flag ECN-Echo
- O emissor ajusta o tamanho a janela de congestionamento, baseado na proporção de pacotes marcados



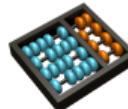
Deadline-Agnostic: *DCTCP*

- Alta taxa de transferência
- Baixa ocupância de buffer
- Não lida com o incast



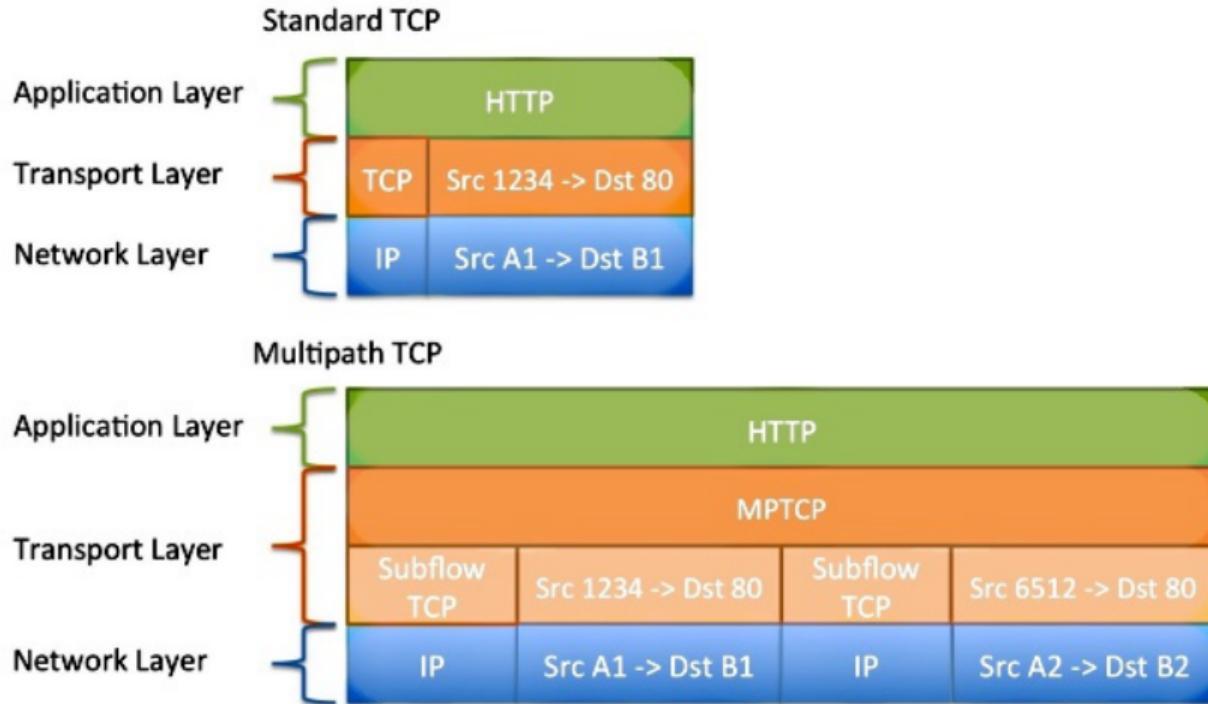
Deadline-Agnostic: *MPTCP*

- Aproveita a existência de múltiplos caminhos na camada de agregação e na camada de núcleo No switch, quando a ocupância da fila ultrapassa o limiar K , marca os pacotes.
- Divide cada fluxo fonte-destino em subfluxos e usa o algoritmo de roteamento de múltiplos caminhos (ECMP).
- Apesar de melhorar a taxa de transferência, pode piorar o problema do incast.



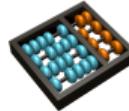
Deadline-Agnostic: *MPTCP*

- Inclui os subfluxos na pilha do protocolo, na camada de transporte.



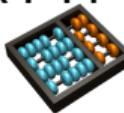
Deadline-Agnostic: *ICTCP*

- Variante do TCP especializada em resolver o problema do incast.
- Previne a perda de pacote ao invés de reenviar quando há perda.
- Receptor sabe a taxa de transferência e largura de banda disponível.



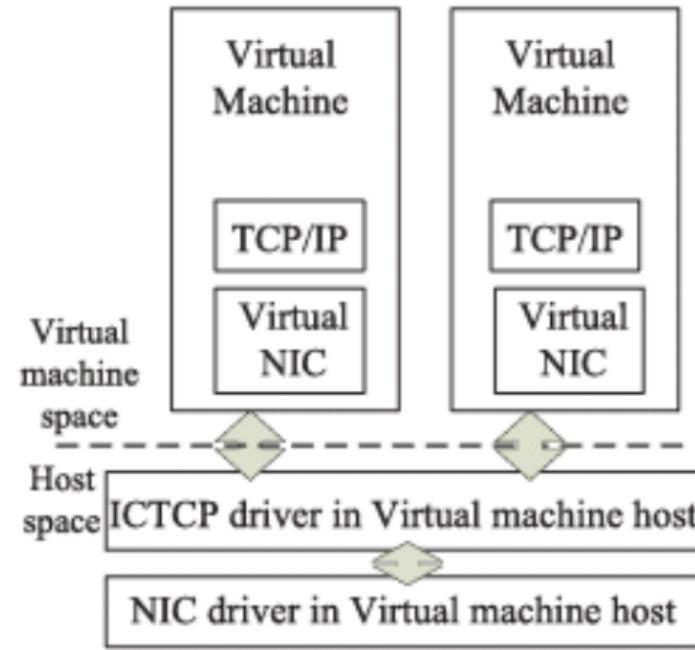
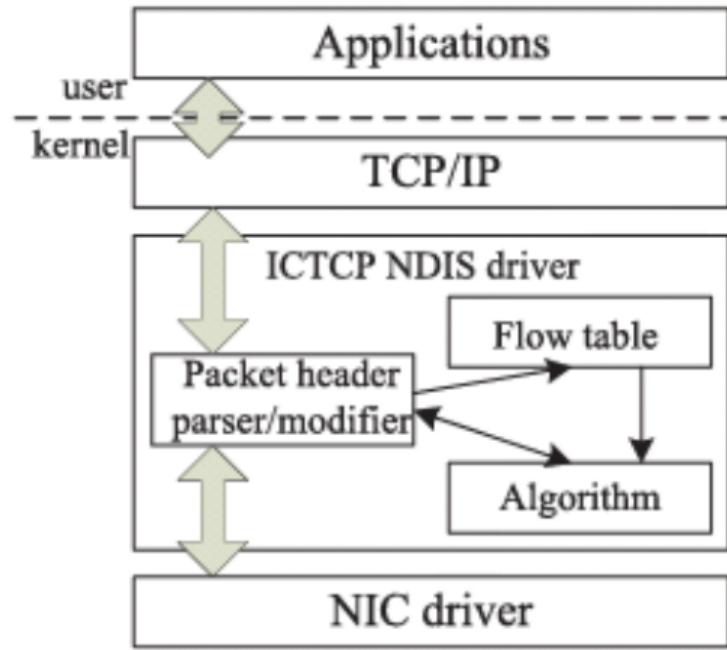
Deadline-Agnostic: *ICTCP*

- Receptor usa a largura de banda disponível como indicador para prevenir o incast.
- O intervalo do controle de congestionamento é ajustado para o RTT de cada fluxo.
- O tamanho da janela de congestionamento deve levar em conta o estado de congestionamento e o requisitado pela aplicação.
- Basicamente, o tamanho da janela de congestionamento é ajustado para cada conexão, estimando a largura de banda disponível e o RTT.



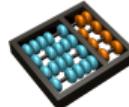
Deadline-Agnostic: ICTCP

- Precisa ser implementado na pilha do TCP



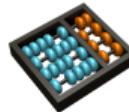
Deadline-Aware: D^3

- Recebe a informação do tamanho do fluxo e o deadline
- Algoritmo guloso para tentar cumprir o máximo de deadlines possíveis
- Roteadores tentam alocar uma taxa adequada para cada fluxo
- Emissor envia dados com a mínima taxa alocada no próximo RTT
- A fonte periodicamente requisita uma nova taxa baseada no deadline e o tamanho do fluxo restante



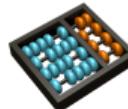
Deadline-Aware: D^3

- Switches precisam de modificações para lidar com as requisições
- Não é compatível com o TCP tradicional, por causa da alocação de largura banda baseado em prioridades sem a informação do deadline no header
- Alocação com algoritmo guloso pode alocar largura de banda para fluxos com deadlines distantes ao invés de deadlines próximos, o que pode causar maior perda de deadlines
- Requisições constantes de taxas tem um overhead



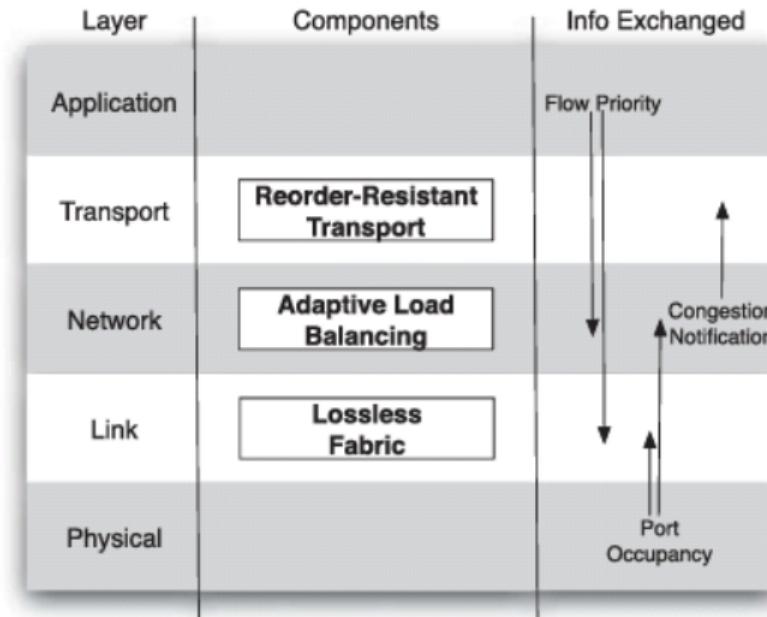
Deadline-Aware: D^2TCP

- Baseado no DCTCP, mas leva o deadline em consideração para redimensionar a janela de congestionamento
- Se a maioria dos deadlines são próximos, ainda pode ocorrer congestionamento
- Se a maioria dos deadlines são distantes, ocorrerá a sub-utilização da rede e baixa taxa de transferência
- Se todos os deadlines são próximos, estão competindo pela largura de banda e nenhum fluxo é adiado, todos os deadlines não serão cumpridos. Caso um fluxo seja adiado, todos os outros podem ser cumpridos.
- O problema é saber qual fluxo sacrificar para satisfazer o máximo de deadlines possíveis



Deadline-Aware: *DeTail*

- Fluxos são associados com prioridades e os switches usam filas de prioridades nas portas de saída e de entrada
- Cada camada da abstração da rede tem uma função



Deadline-Aware: *DeTail*

Enlace:

- Controle de fluxo “hop-by-hop”
- Tenta amenizar o bloqueio de “head-of-line” (HOL)
- Recebe informações da camada de rede em relação ao balanço de carga adaptativo
- Recebe informações da camada de transporte sobre o estado do ECN



Deadline-Aware: *DeTail*

Rede:

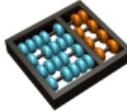
- Balanço de carga adaptativo baseado em pacotes de acordo com o nível de congestionamento
- Pode transmitir pacotes por caminhos que estão com pouca carga



Deadline-Aware: *DeTail*

Rede:

- Usa o ECN para marcar fluxos de baixa prioridade quando os bytes transmitidos para o destino ultrapassam um certo limiar
- Previne congestionamento persistente



Deadline-Aware: *DeTail*

Aplicação:

- Seleciona as prioridades de cada fluxo baseado na sensibilidade de latência



Deadline-Aware: *PDQ*

- Semelhante ao D^3 , mas ao contrário do D^3 , escalona taxas de acordo com a criticalidade dos fluxos ao invés da política “first-come first-serve”
- Duas políticas de alocação, implementadas de forma totalmente distribuída:
 - ▶ EDF (Earliest Deadline First)
 - ▶ SJF (Shortest Job First)



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

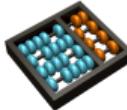
4 Protocolos

- Roteamento
- Transporte

5 Tendências

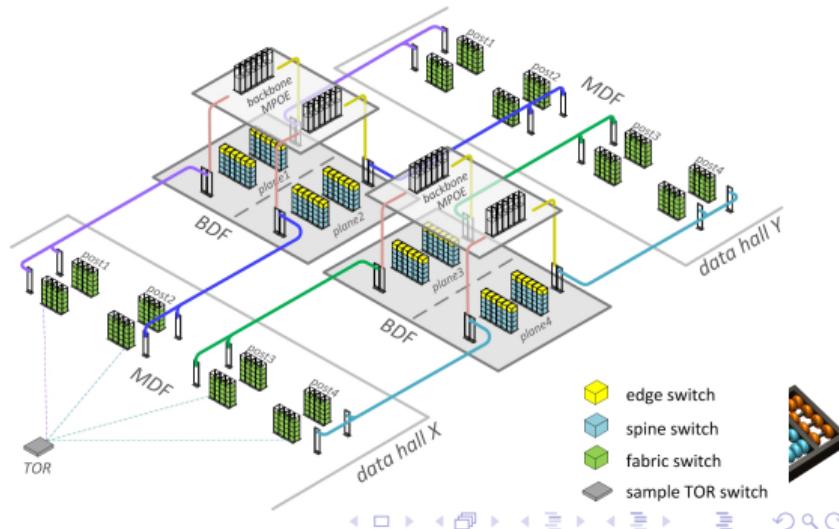
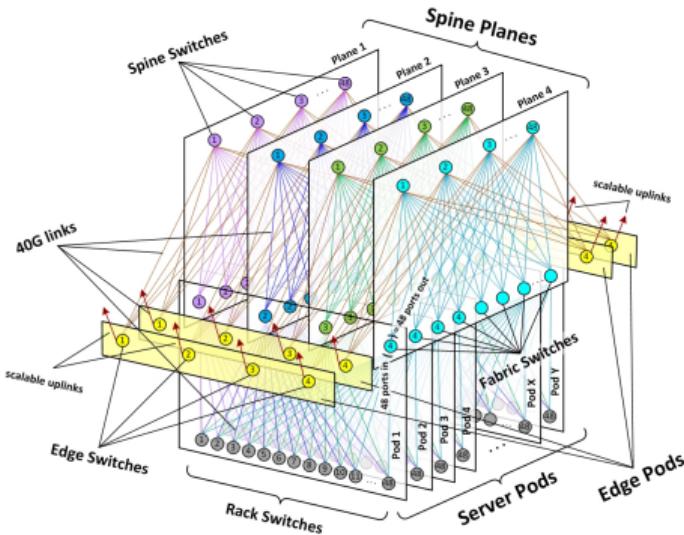
6 Conclusão

7 Pergunta



Facebook Fabric

- A próxima geração de data center do Facebook
- Topologia não hierarquizada
- Rede de alto desempenho
- Sem oversubscription



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

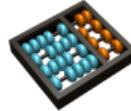
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta



Ponha aqui seu texto



Sumário

1 Introdução

2 Motivação

3 Topologias

- Tradicionais
- SDN

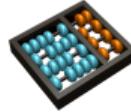
4 Protocolos

- Roteamento
- Transporte

5 Tendências

6 Conclusão

7 Pergunta



Ponha aqui seu texto

