Gatekeeper: Supporting Bandwidth Guarantees for Multi-tenant Datacenter Networks

Henrique Rodrigues, Yoshio Turner, Jose Renato Santos,
Paolo Victor, Dorgival Guedes

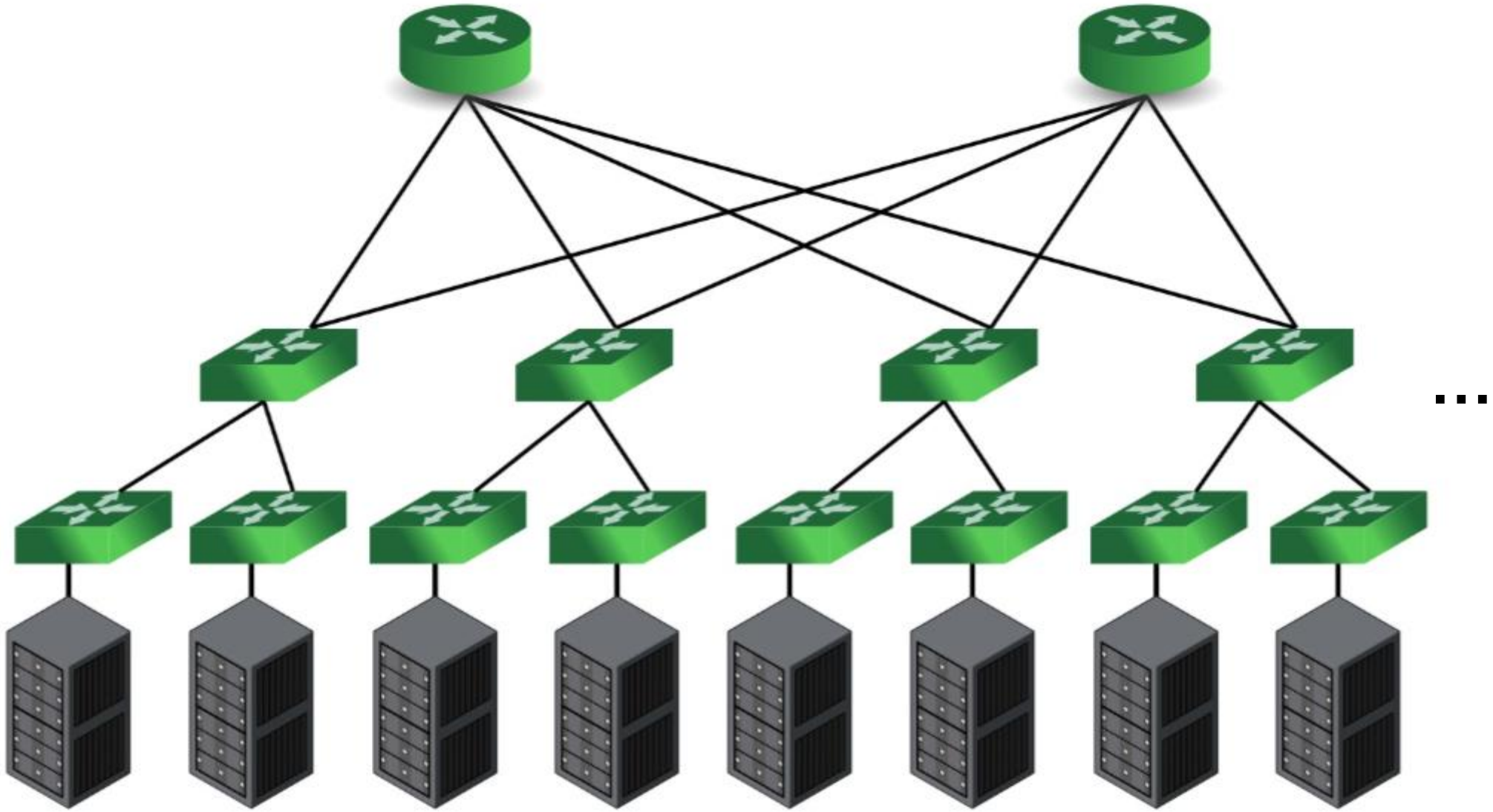
UFMG

HP Labs

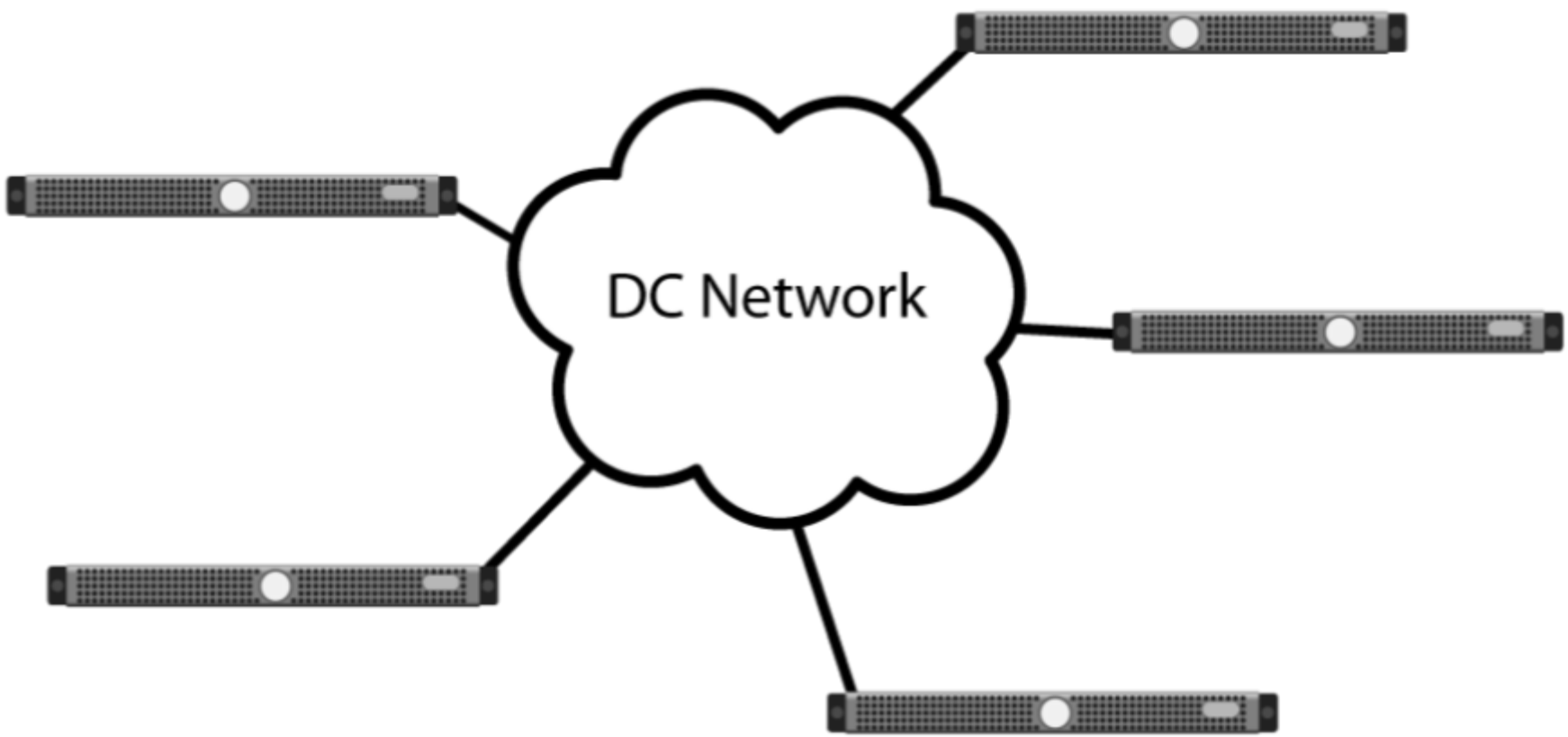
WIOV 2011, Portland, OR

The Problem: Network Performance Isolation

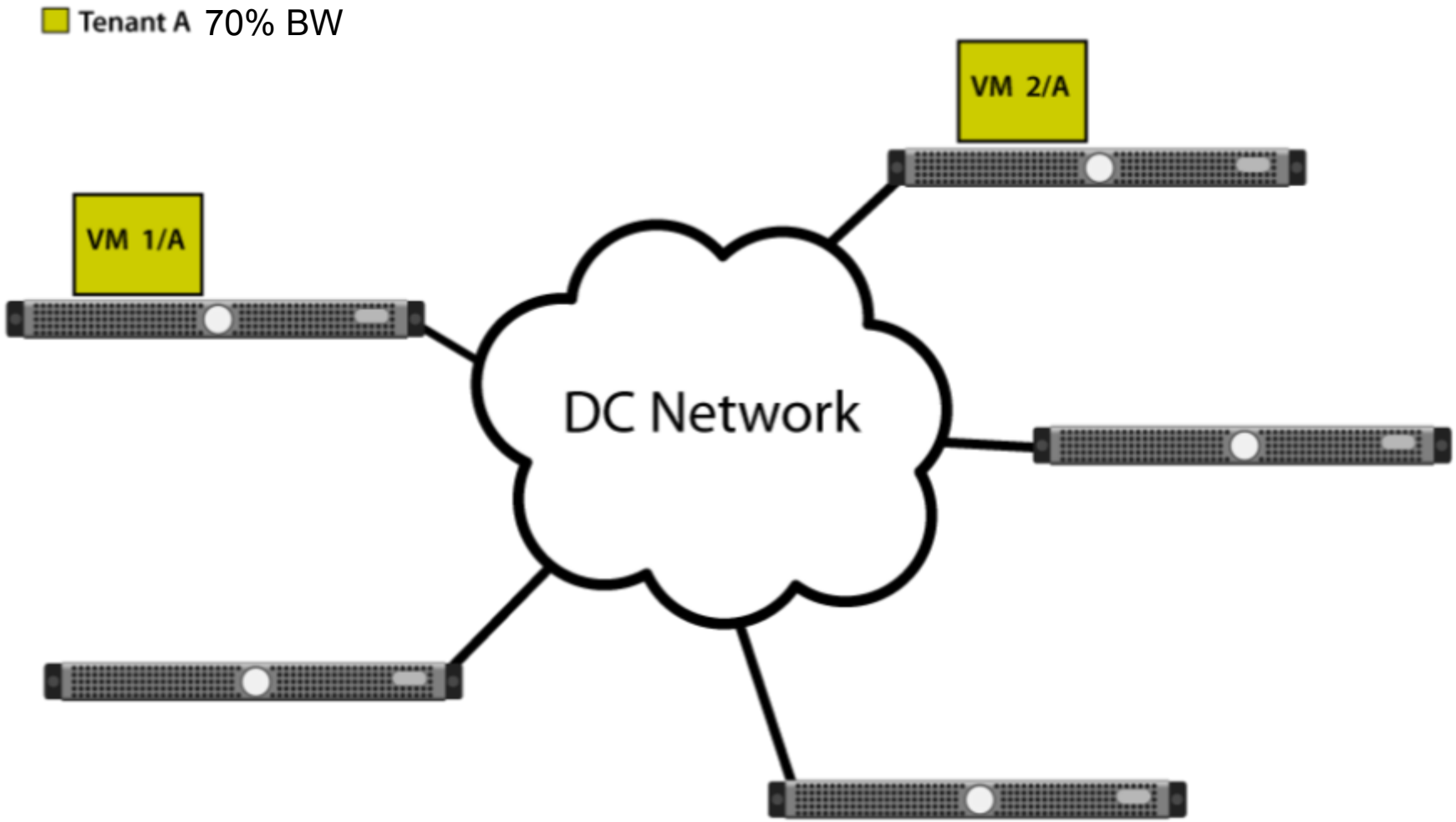
Suppose that you have a datacenter…



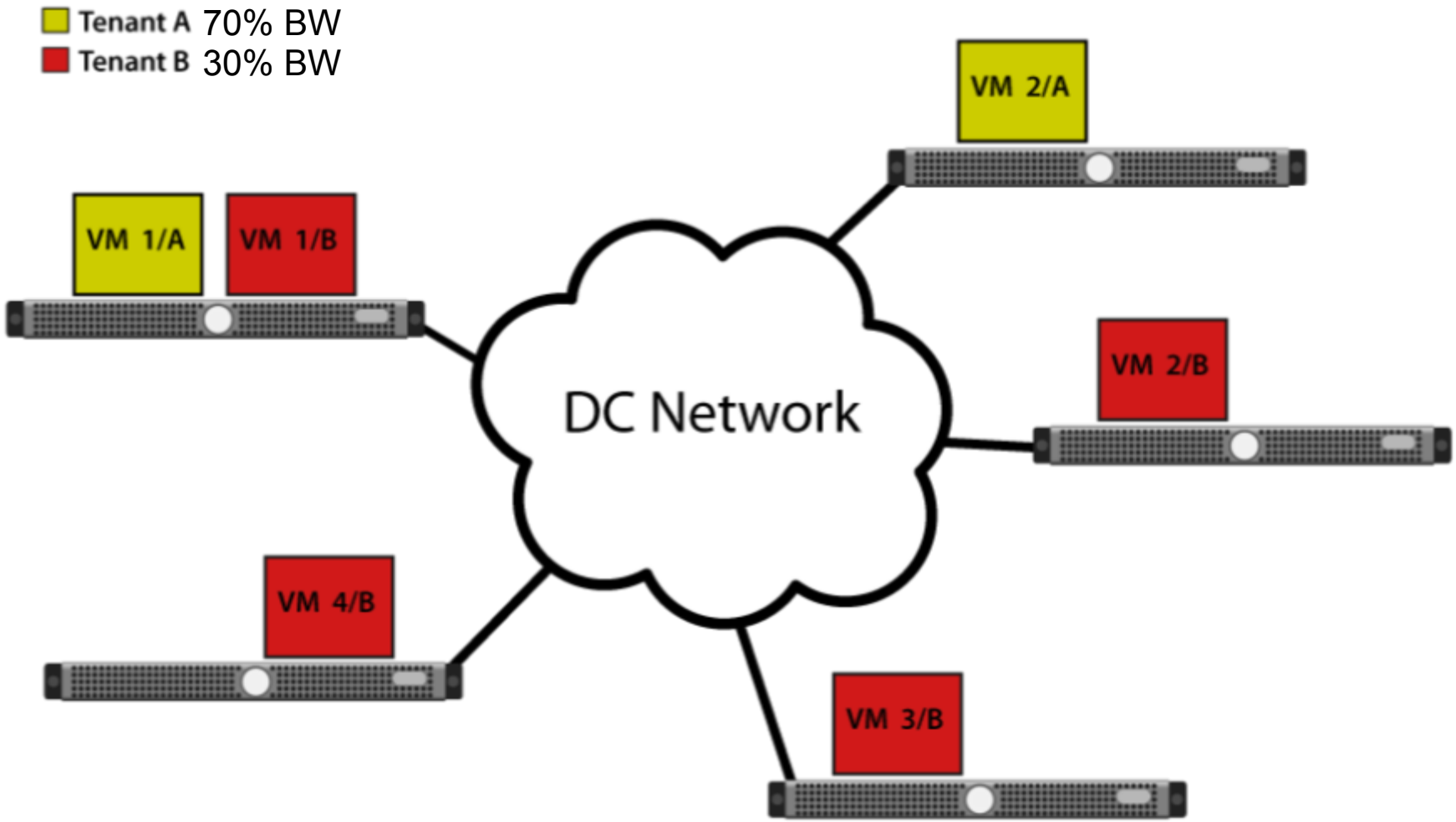
Suppose that you have a datacenter…



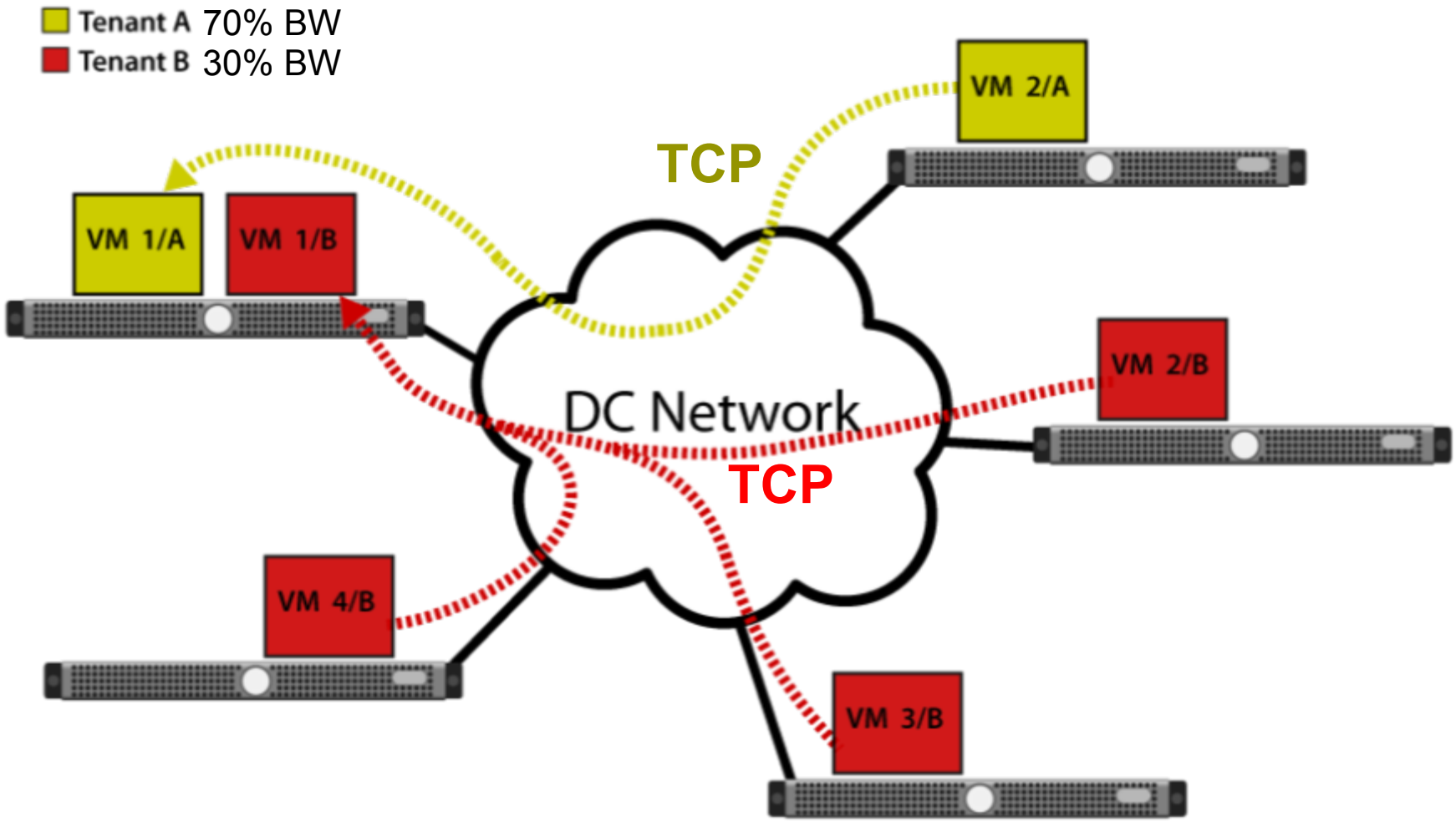
And you are an IaaS provider …



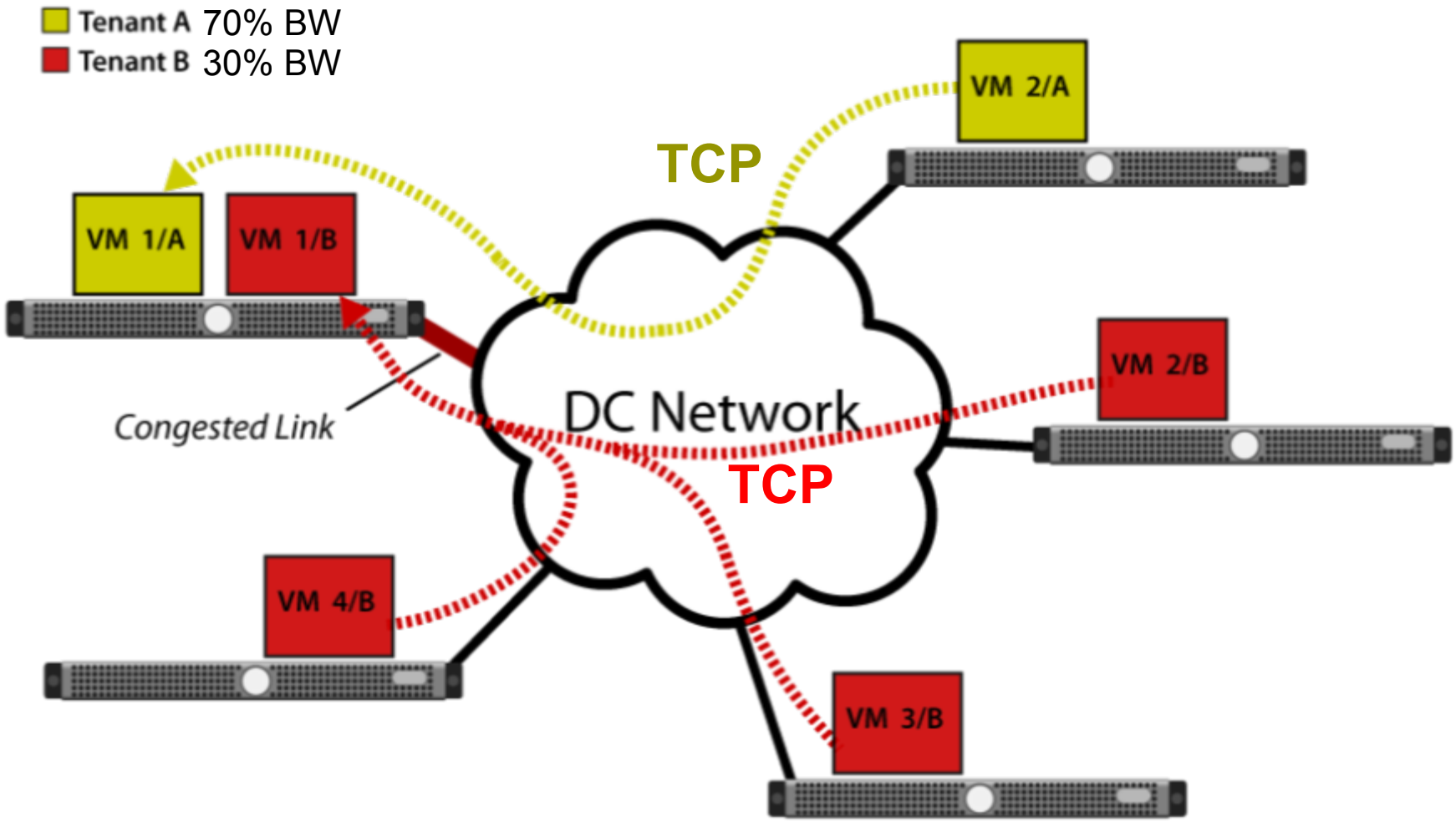
And you are an IaaS provider …



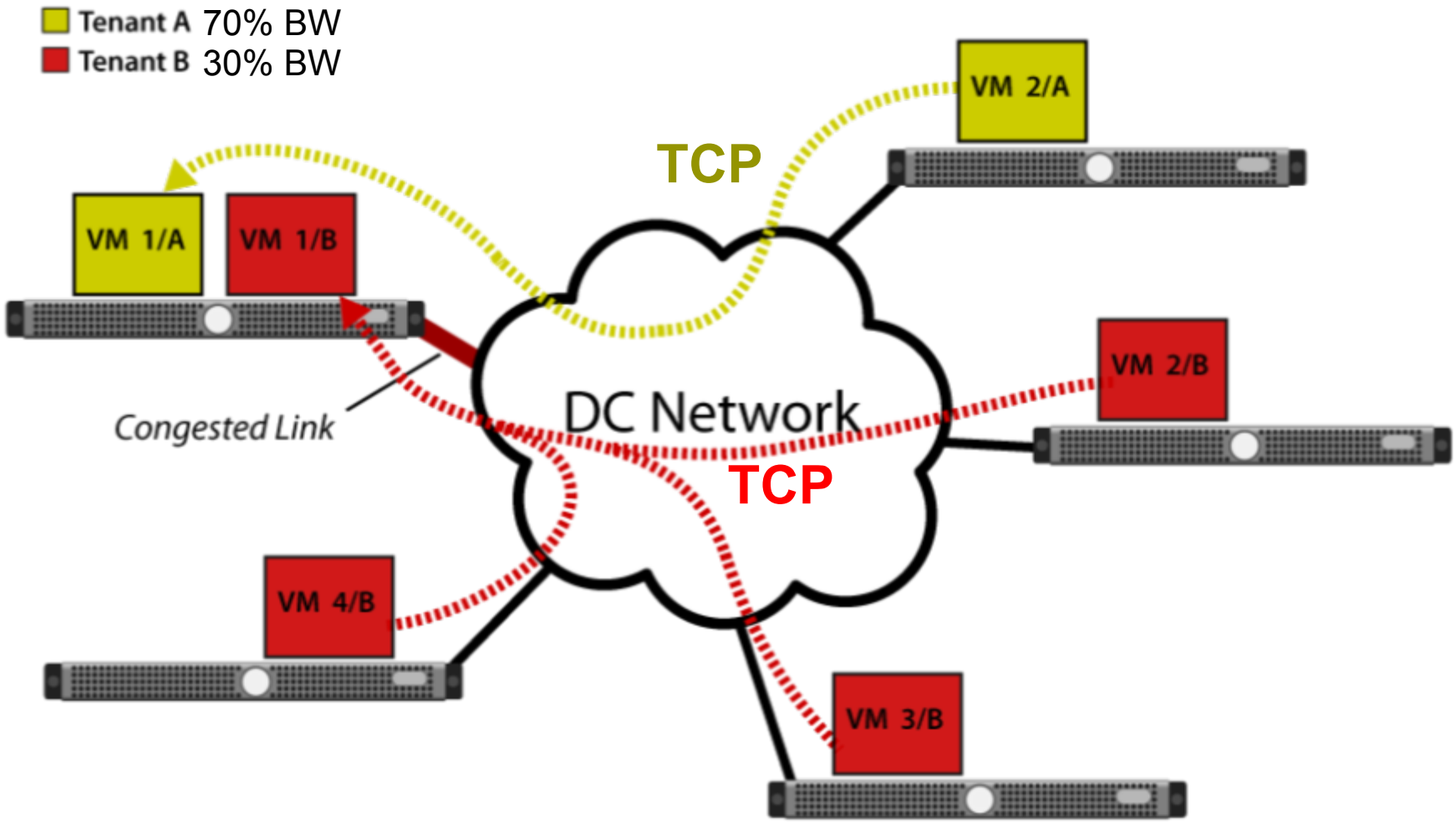
... and your network faces this traffic pattern:



... and your network faces this traffic pattern:

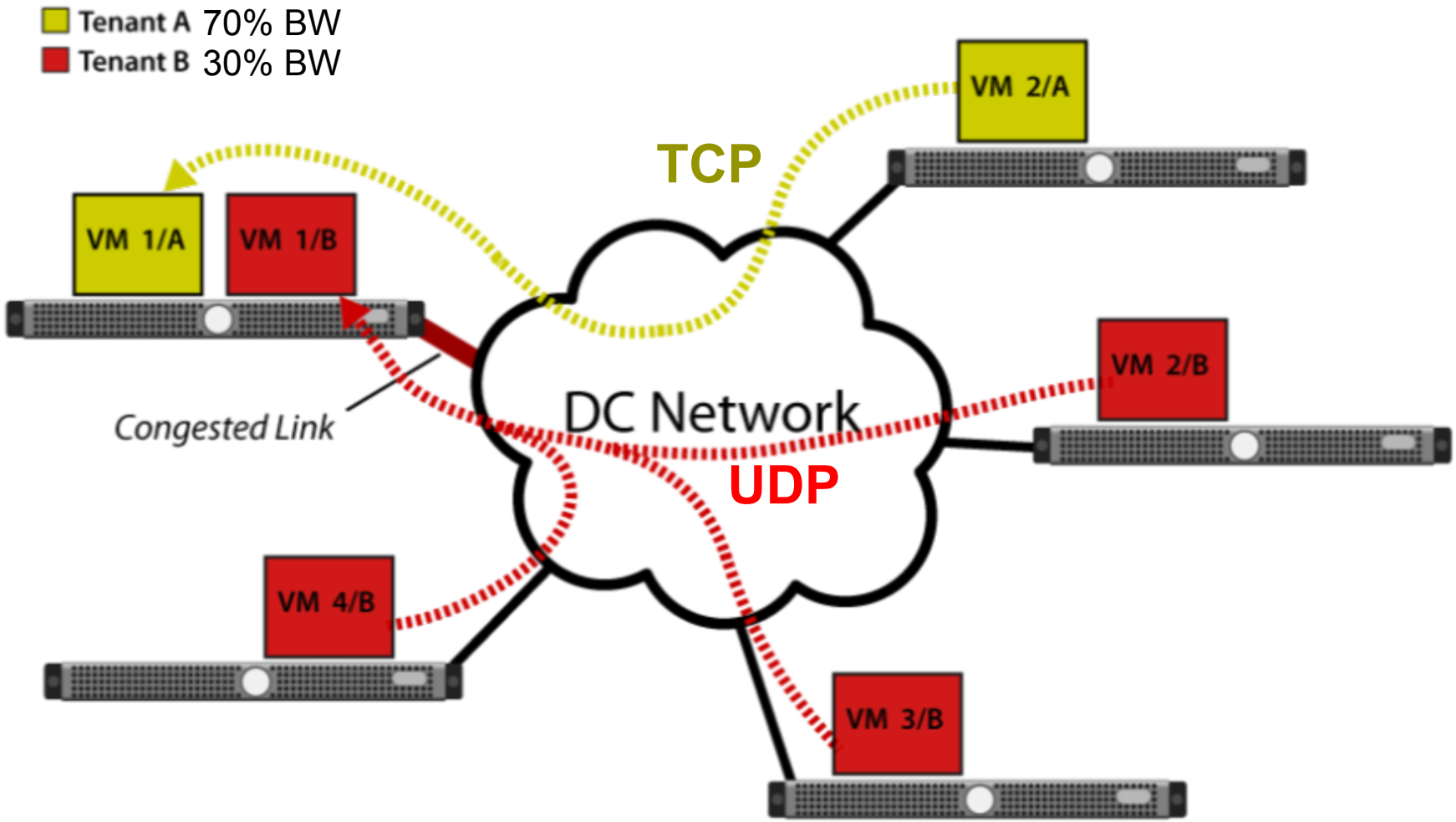


... and your network faces this traffic pattern:

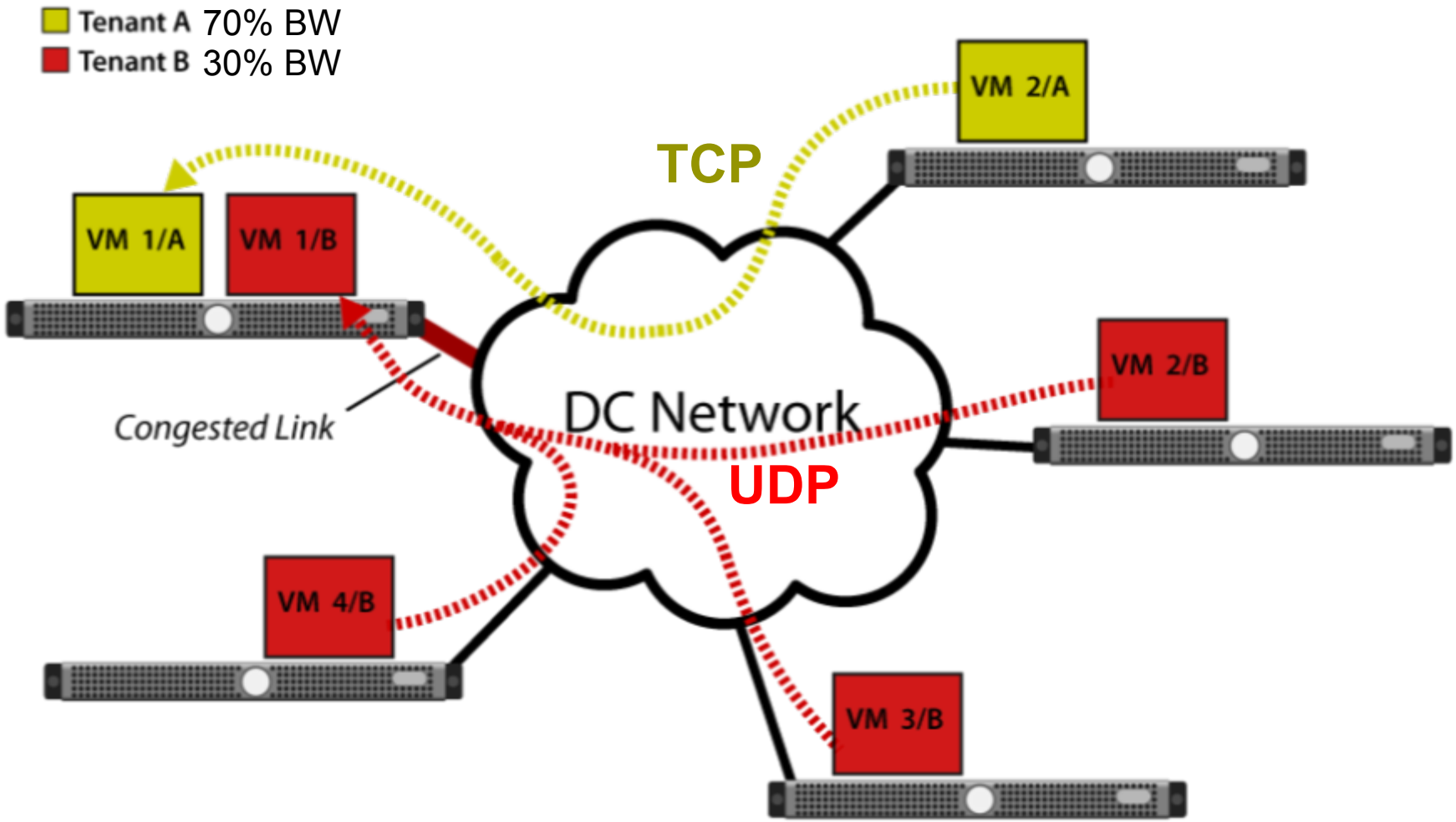


TCP is flow-based, not tenant-aware...

It becomes worse with these transport protocols:

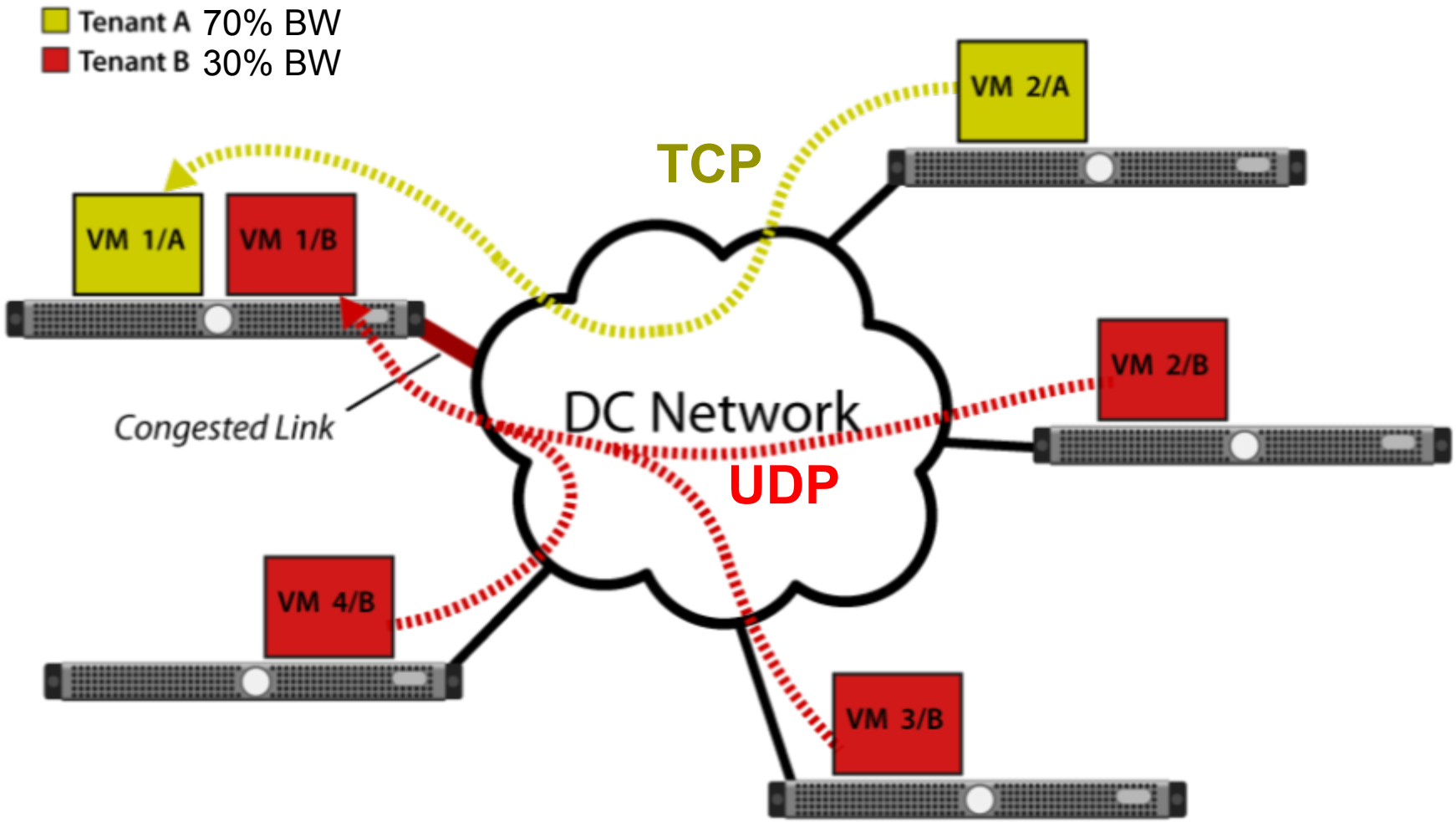


It becomes worse with these transport protocols:



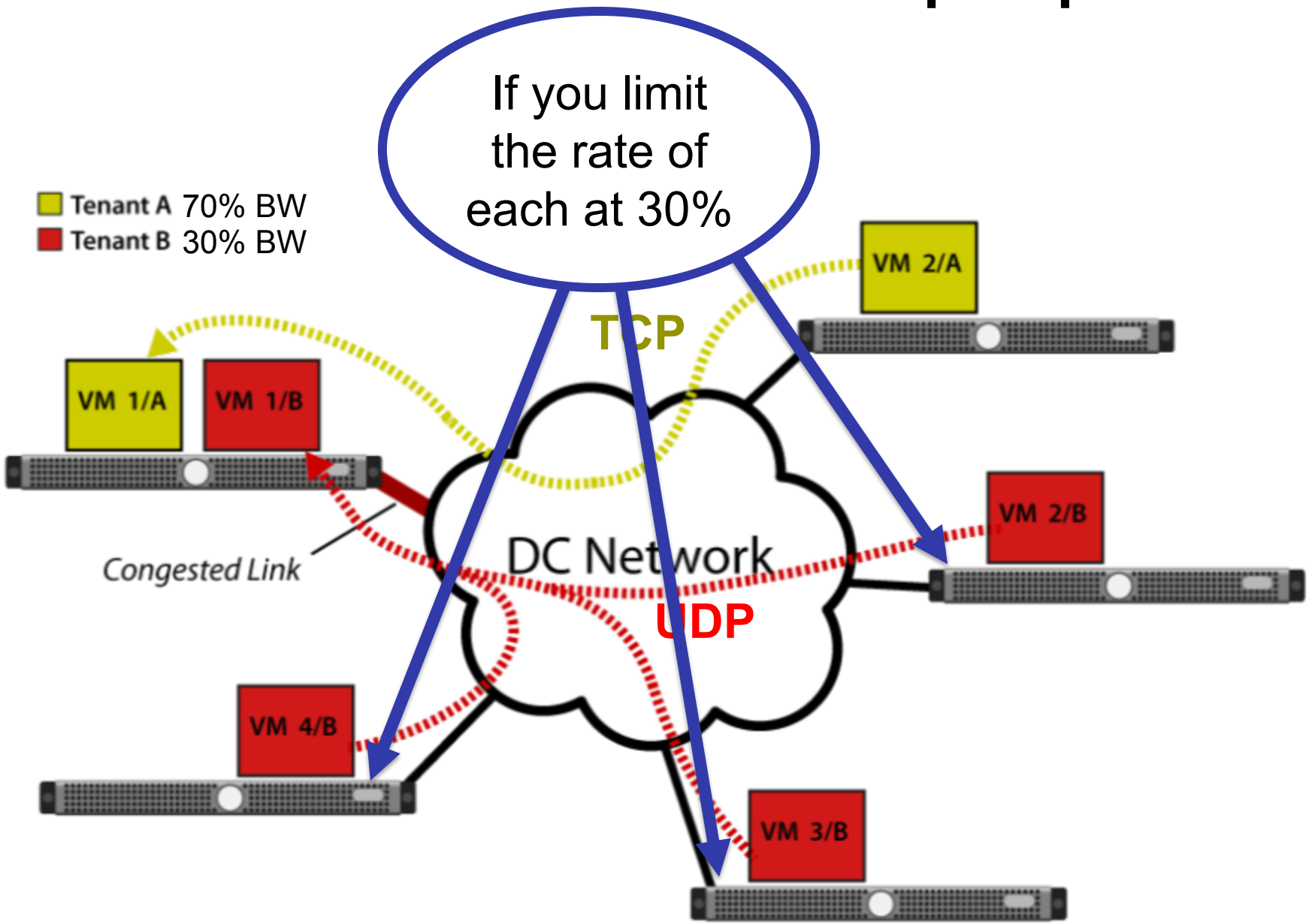
UDP will consume most of the bandwidth

It becomes worse with these transport protocols:



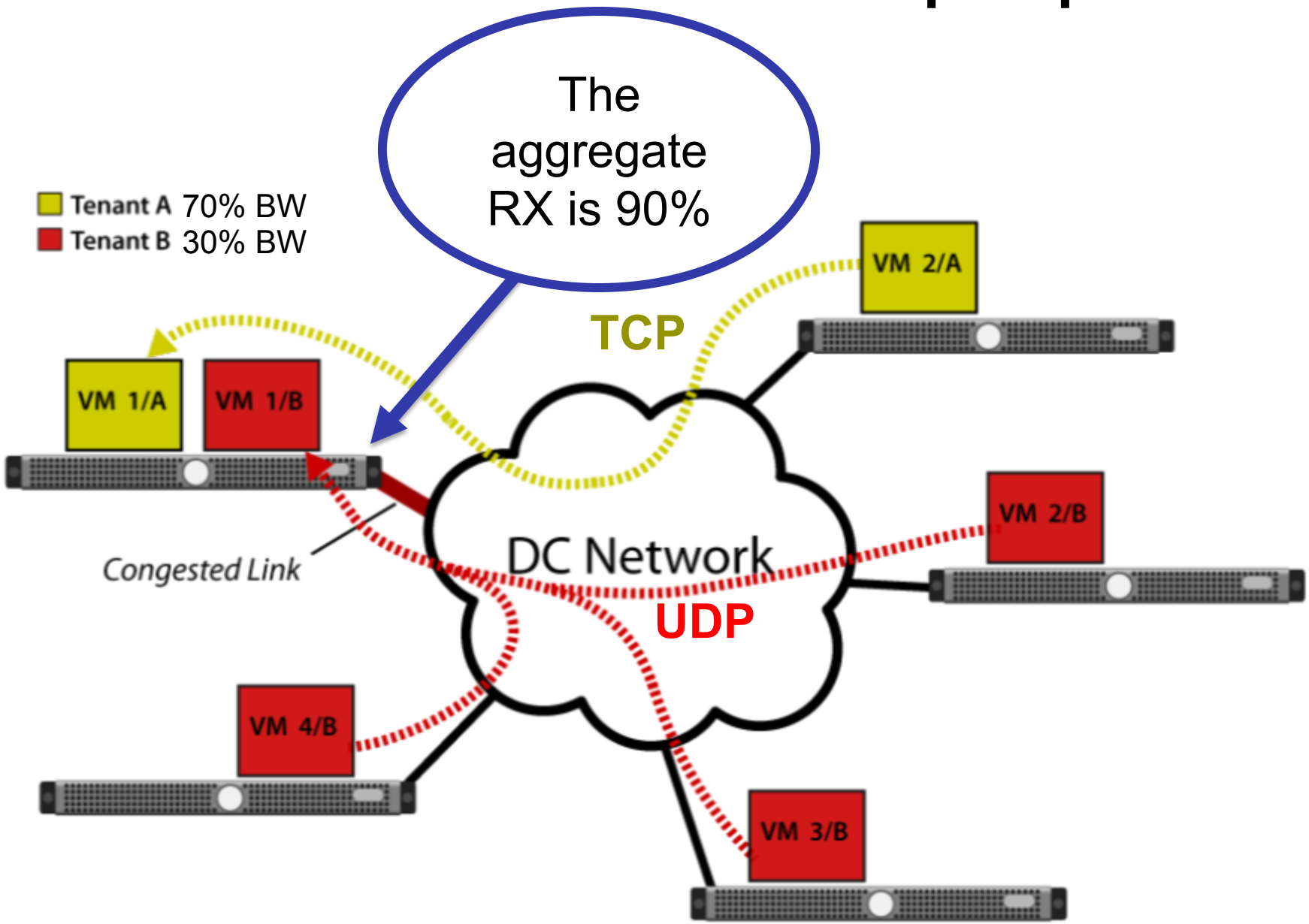
Using rate limiters at each server doesn't solve the problem…

It becomes worse with these transport protocols:



Using rate limiters at each server doesn't solve the problem…

It becomes worse with these transport protocols:



Using rate limiters at each server doesn't solve the problem…

The Problem: Network Performance Isolation

- *How can we enforce that all tenants will have at least the minimum amount of network resources they need to keep their services up?*

  - In other words, *how to provide network performance isolation to multi-tenant datacenters*?

Practical requirements for a traffic isolation mechanism/system

Requirements for a practical solution

- **Scalability**
  Datacenter supports thousands of physical servers hosting 10s of thousands of tenants and 10s to 100s of thousands of VMs
- **Intuitive Service Model**
  Straightforward for tenants to understand and specify their network performance needs
- **Robust against untrusted tenants**
  IaaS model allows users to run arbitrary code as tenants, giving users total control over the network stack. Malicious users could jeopardize the performance of other tenants
- **Flexibility / Predictability**
  What should we do with the idle bandwidth?
  Work conserving vs non-work conserving?

Existing solutions don't meet all these requirements

Solution	Scalable	Flexibility / Predictability	Intuitive Model	Robustness
TCP	✔	✗	✗	✗
BW Capping (policing)	✔	✗	✔	✗
Secondnet	✔	✔	✗	✔
Seawall	✔	✗	✗	✔
AF-QCN	✔	✗	✗	✔

Our approach

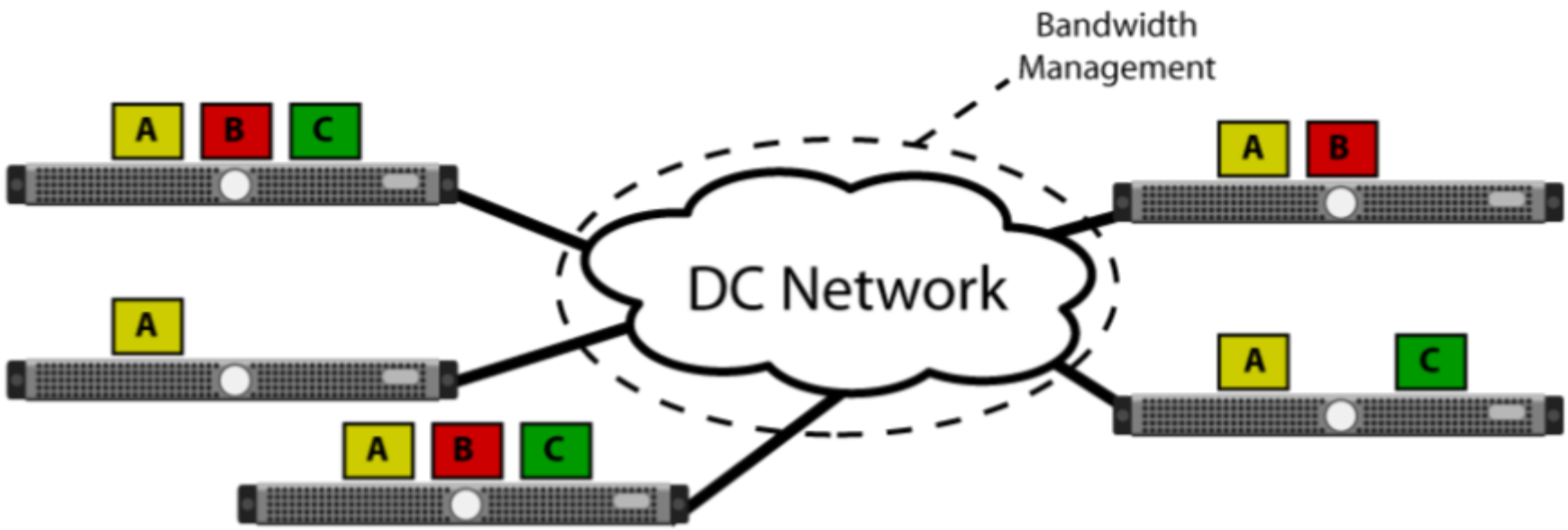
Assumption

Bisection bandwidth should not be a problem:

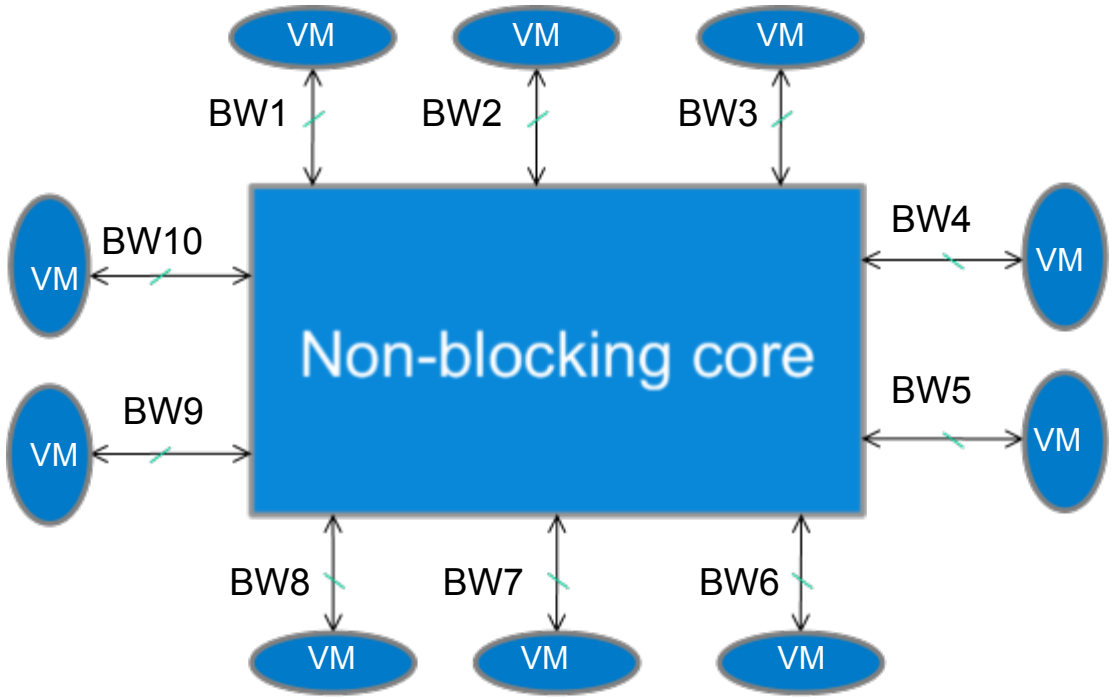
- Emerging multi-path technologies will enable high bandwidth networks with full-bisection bandwidth
- Smart tenant placement: tenant VMs placed close to each other in the network topology
  - Results on DC traffic analysis show that most of the congestion happens within racks, not at the core

Our approach

- ***Assume core is over-provisioned and manage bandwidth at edge***
  - Addresses scalability challenge:
    Limited number of tenants in each edge link



Tenant Performance Model Abstraction



- Simple abstraction to tenant
  - Model similar to physical servers connected to a switch
- Guaranteed bandwidth for each VM (TX and RX)
  - Minimum and Maximum rate per vNIC

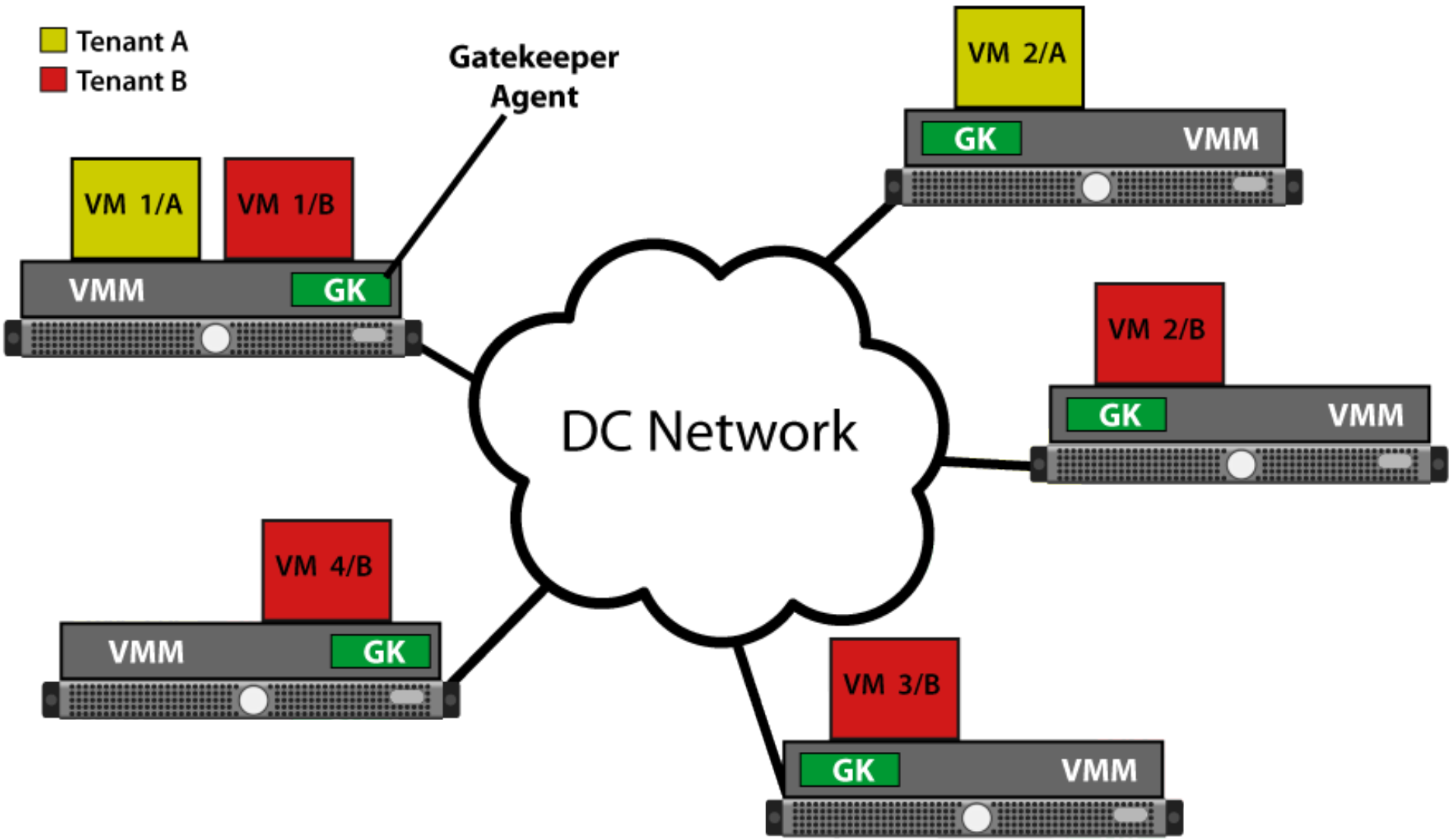
Gatekeeper

- Provides network isolation for multi-tenant datacenters using a distributed mechanism

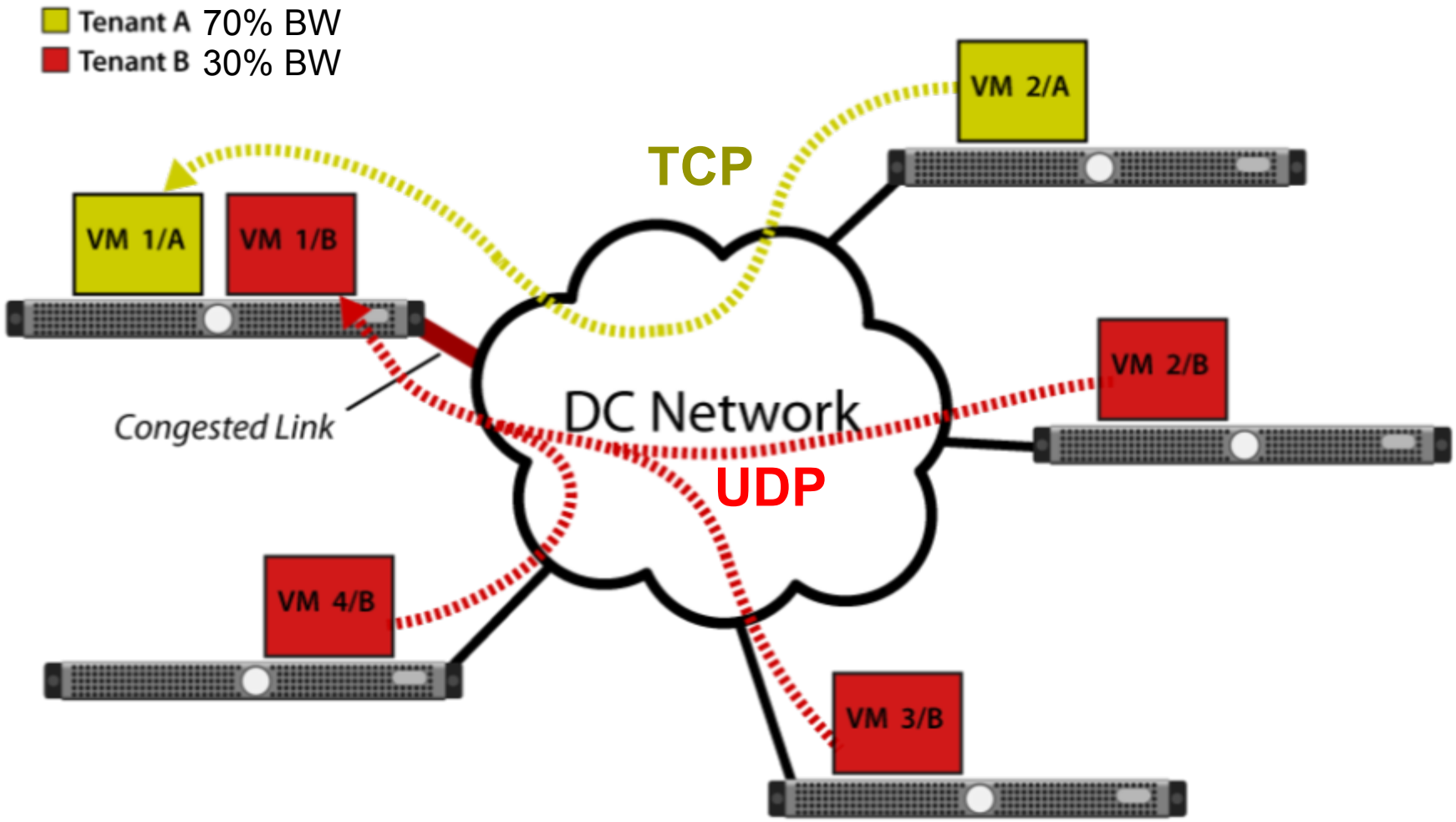
- Agents implemented at the virtualization layer coordinate bandwidth allocation dynamically, based on tenants' guarantees

Gatekeeper

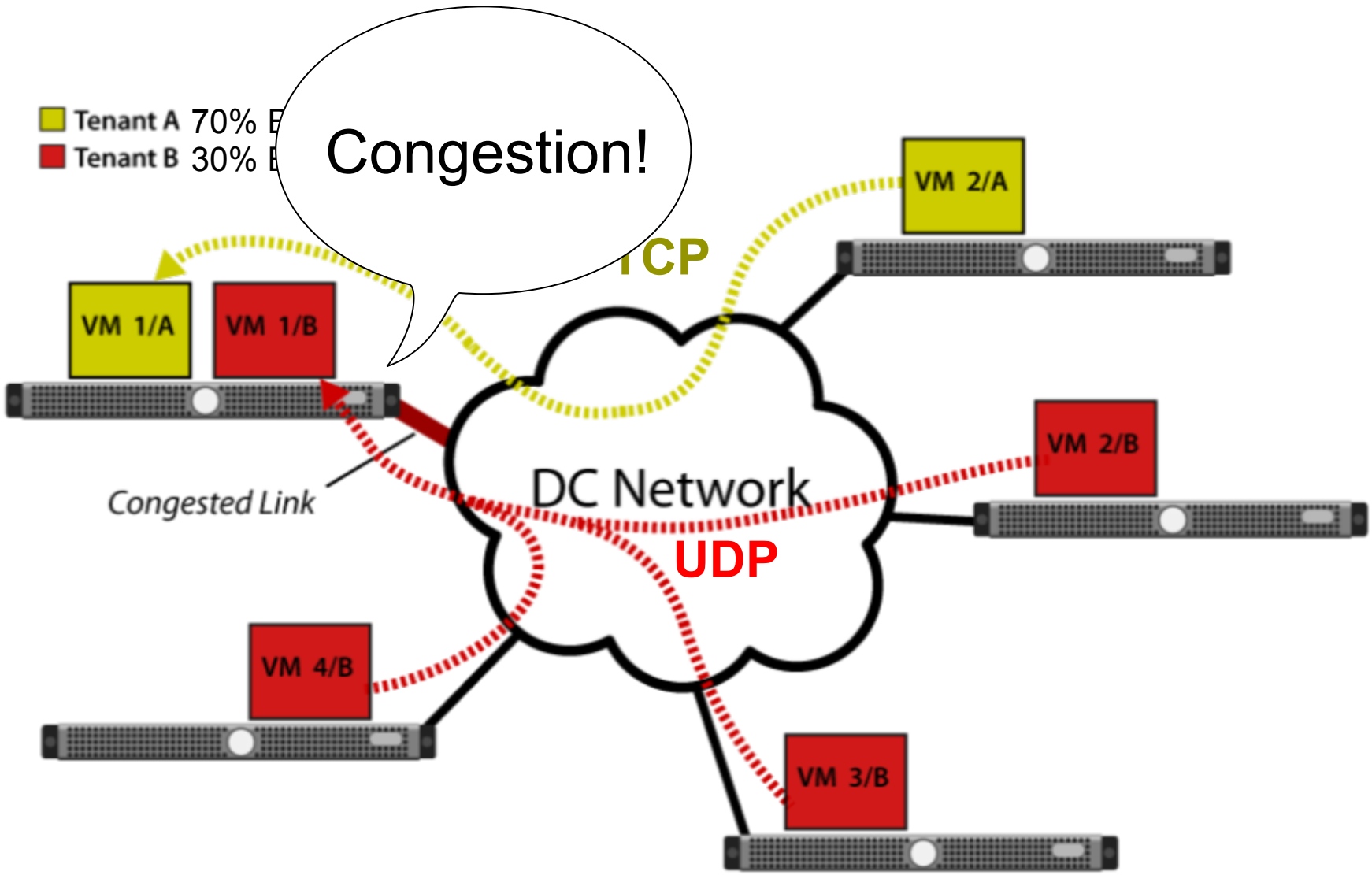
- Agents in the VMM control the transmission (TX) and coordinate the reception (RX)



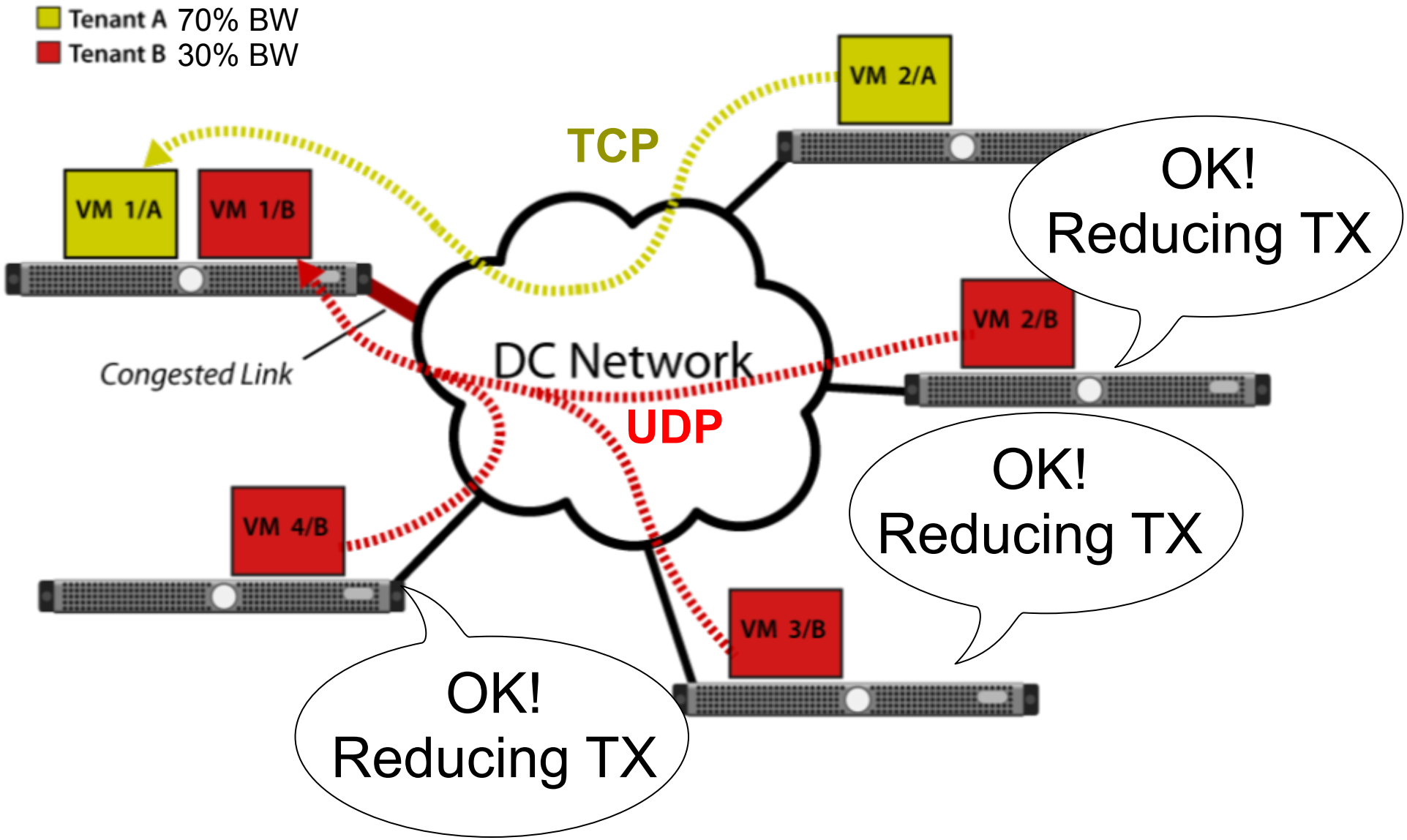
Gatekeeper - Overview



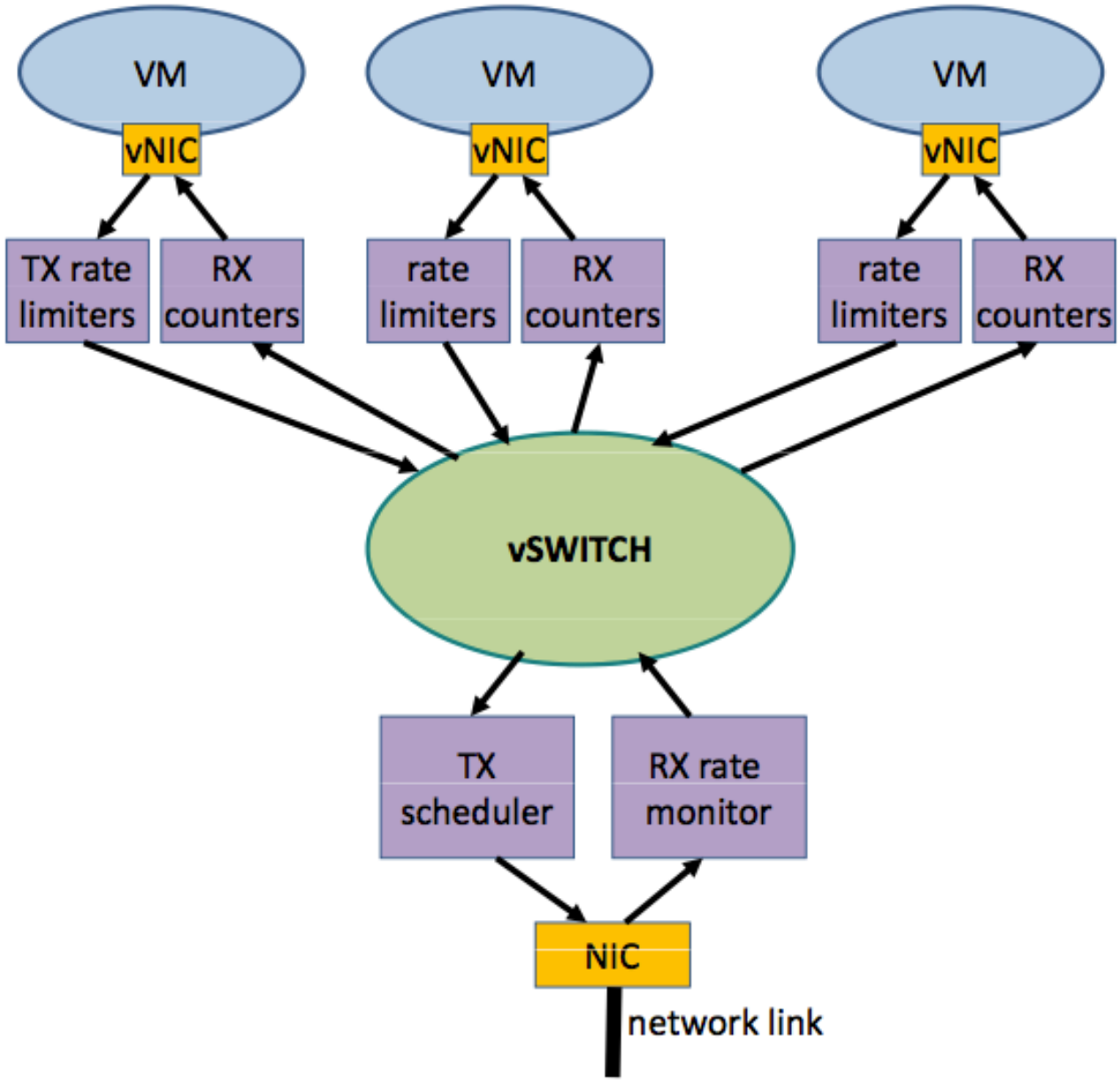
Gatekeeper - Overview



Gatekeeper - Overview



Gatekeeper Architecture



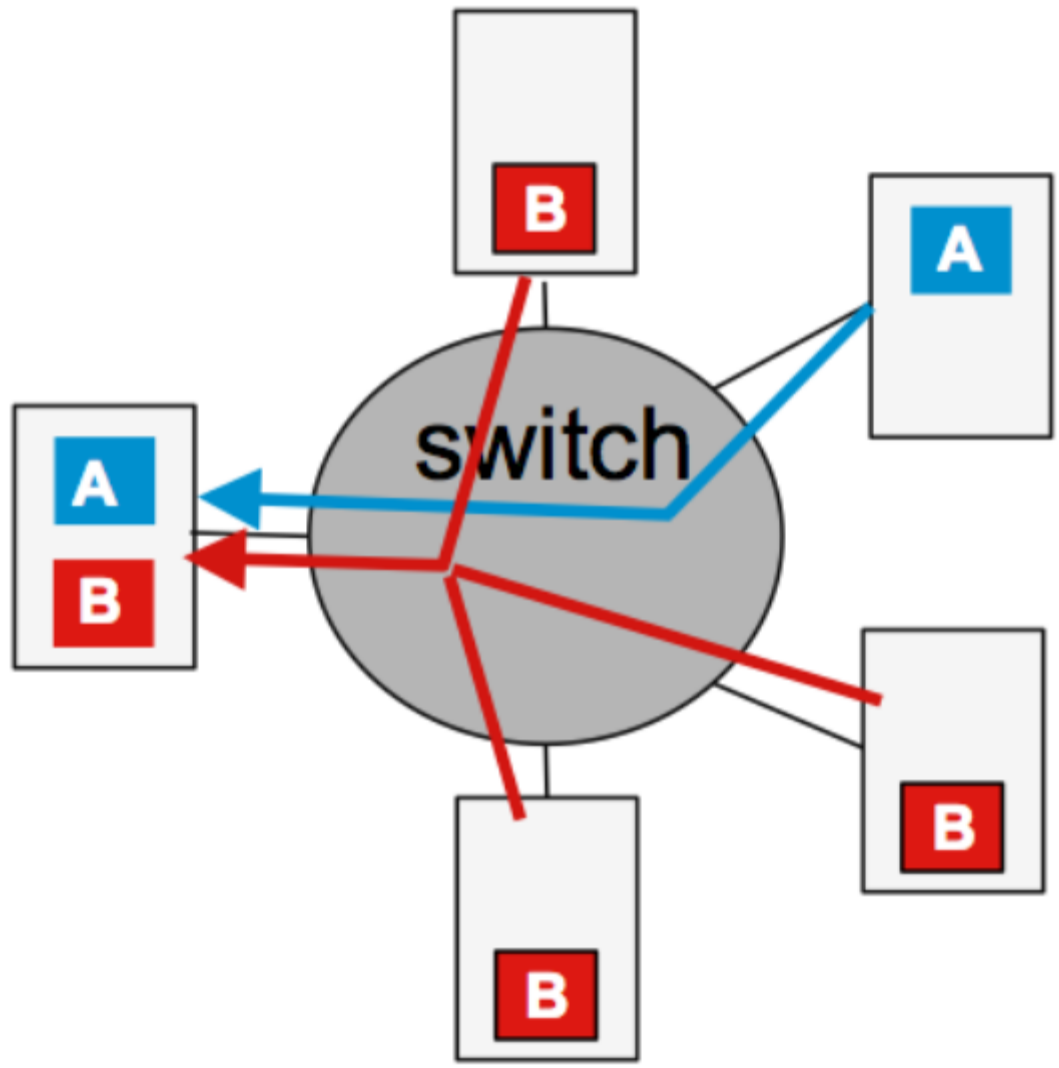
Gatekeeper Prototype

- Xen/Linux
- Gatekeeper integrated into Linux Open vSwitch
- Leverage Linux traffic control mechanism (HTB) for rate control

Example - RX

2 Tenants share a gigabit link:

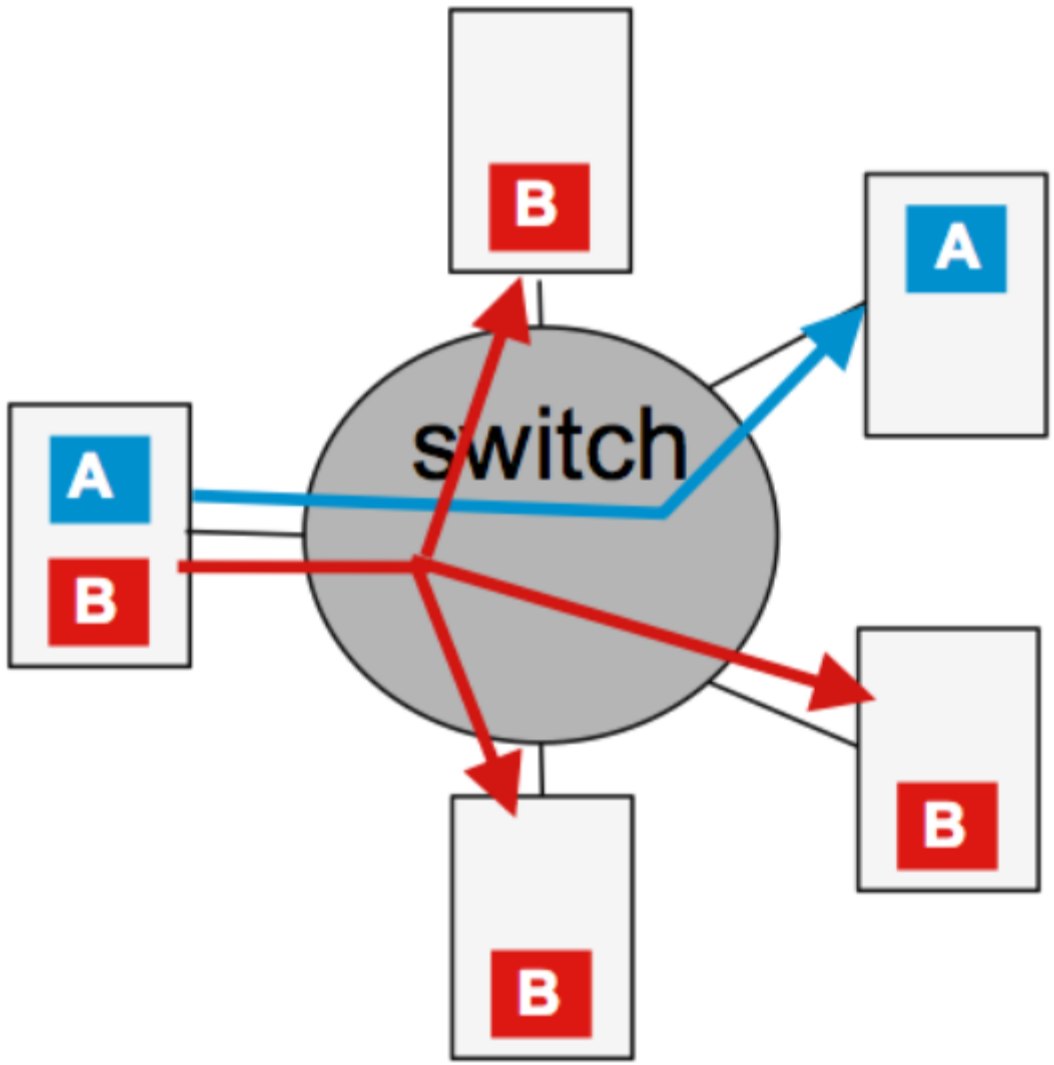
- Tenant A
  - **70%** of the link,
  - **1 TCP** Flow
- Tenant B
  - **30%** of the link,
  - 3 Flows (TCP or UDP)



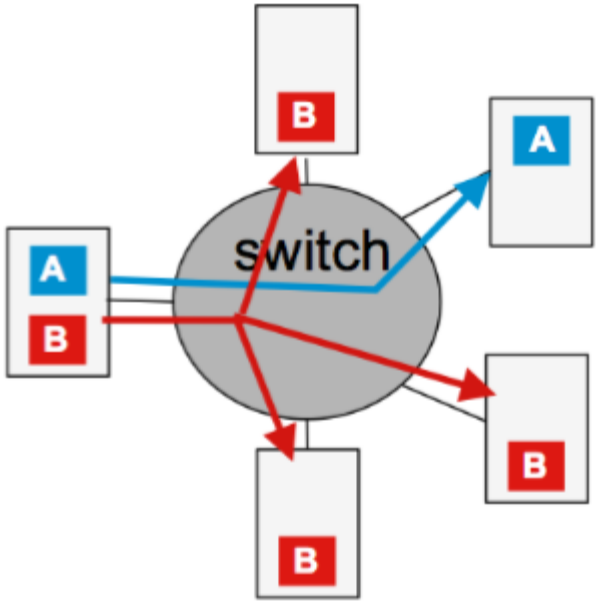
Example - TX

2 Tenants share a gigabit link:

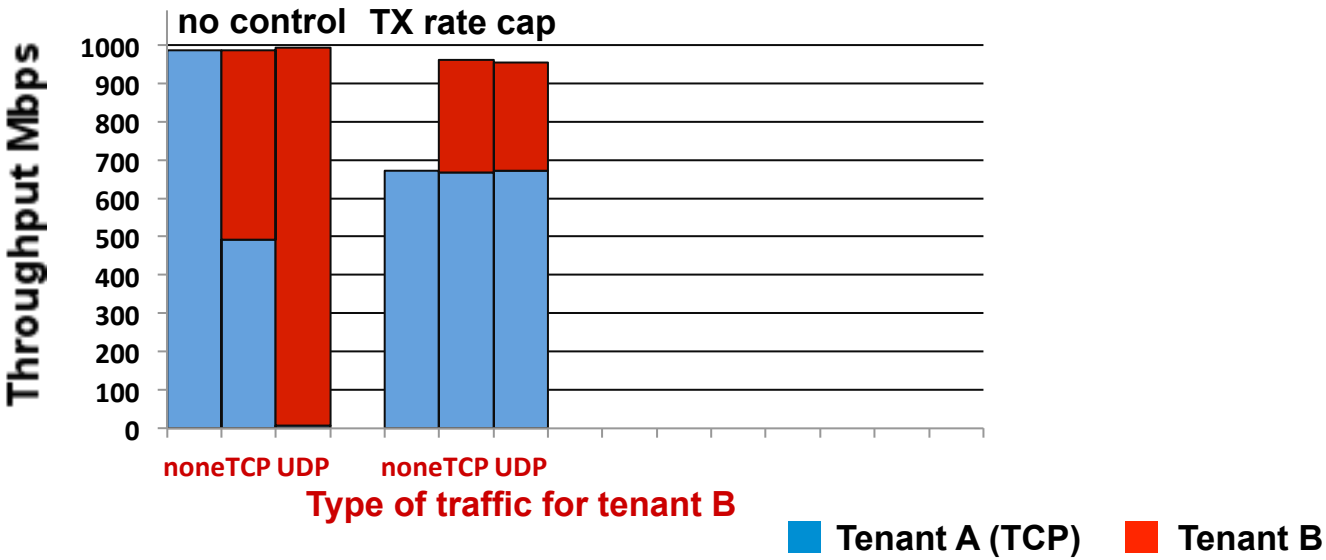
- Tenant A
  - **70%** of the link,
  - **1 TCP** Flow
- Tenant B
  - **30%** of the link,
  - 3 Flows (TCP or UDP)



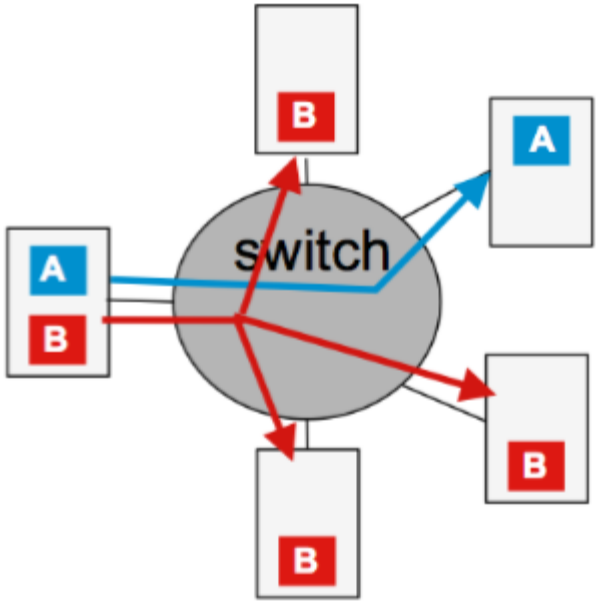
Example – Results ***without*** Gatekeeper

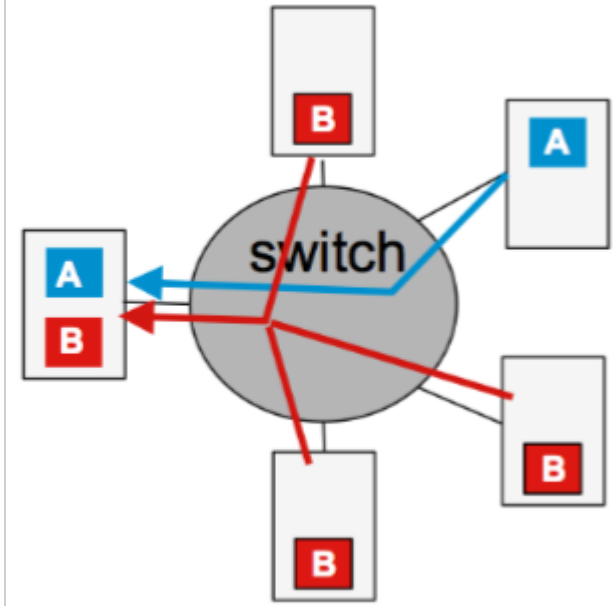


Transmit (TX) Scenario

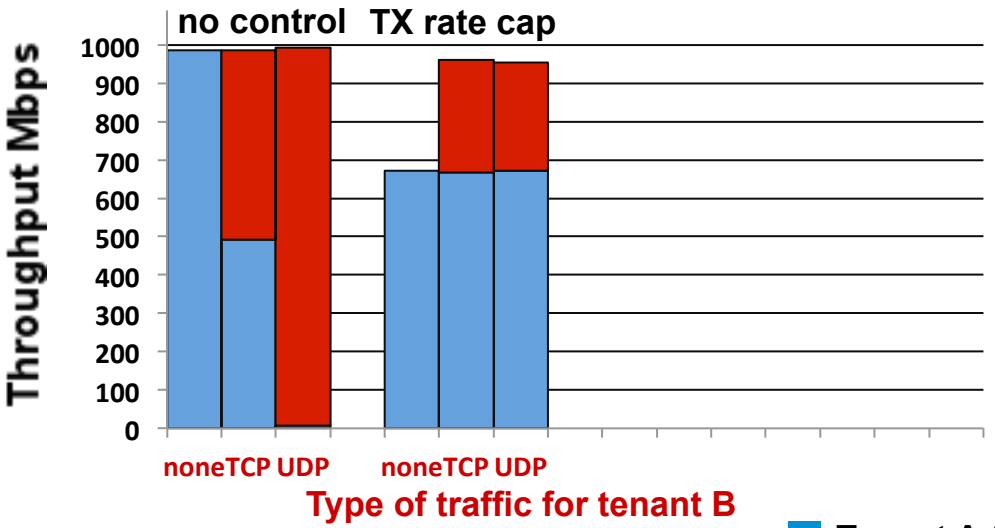


Example – Results ***without*** Gatekeeper

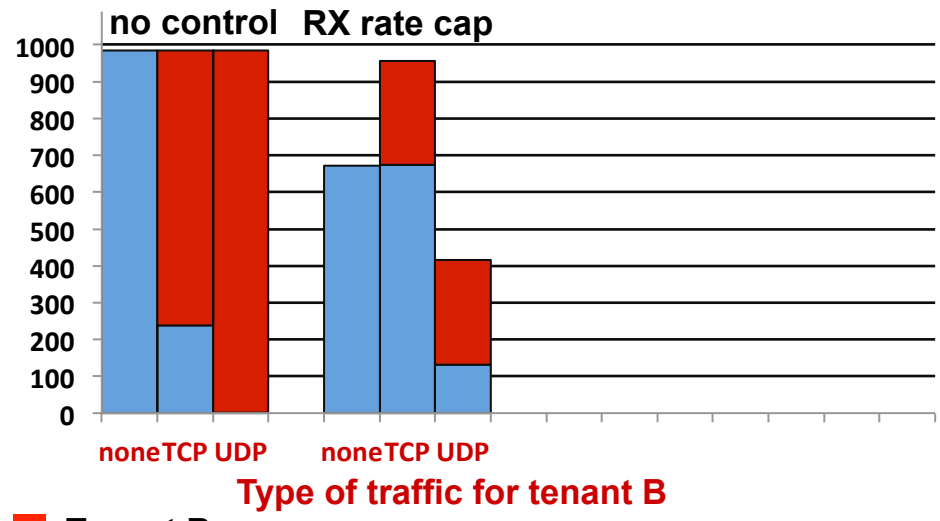




Transmit (TX) Scenario

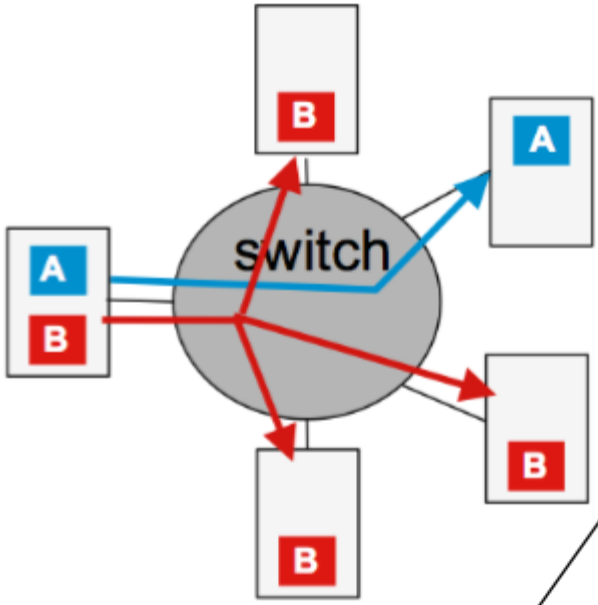


Receive (RX) Scenario

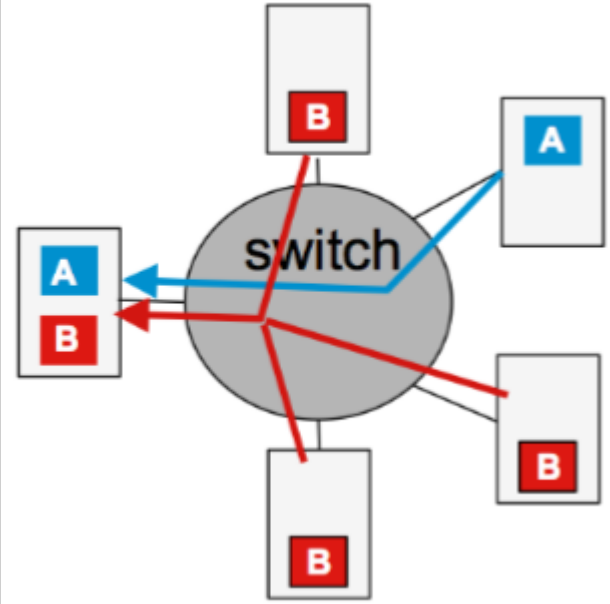


Tenant A (TCP) Tenant B

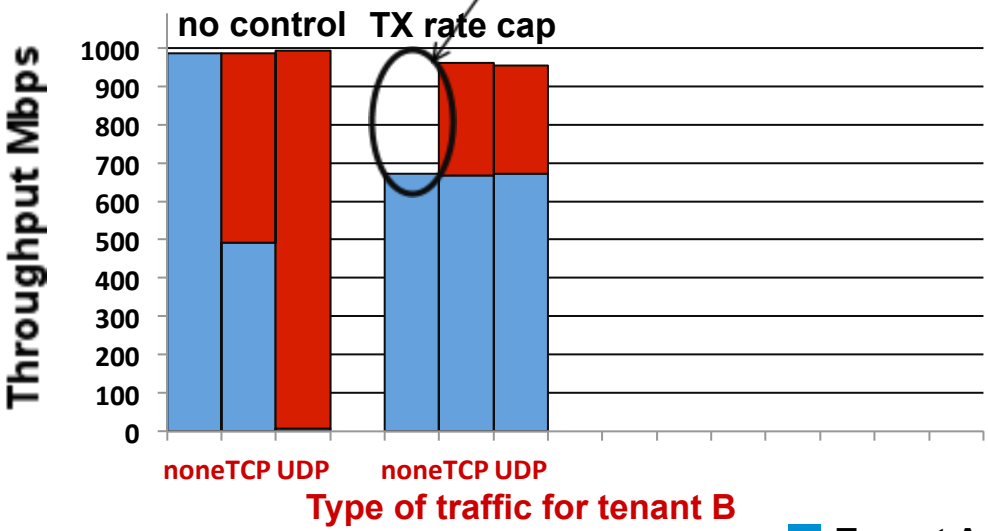
Example – Results ***without*** Gatekeeper



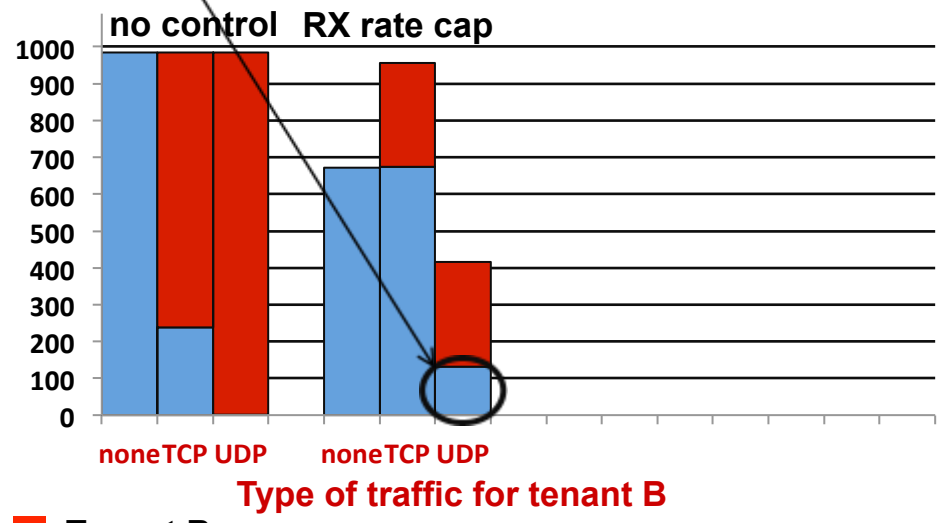
- Bandwidth Capping doesn't reallocate unused bandwidth (non work-conserving)
- UDP consumes most of the switch resources



Transmit (TX) Scenario

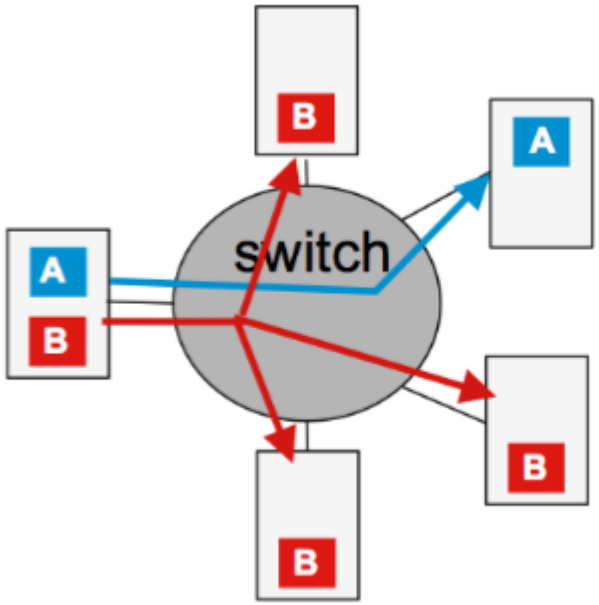


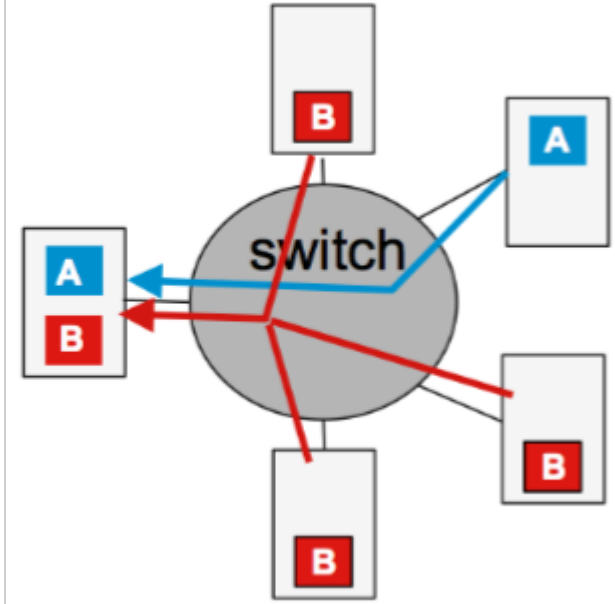
Receive (RX) Scenario



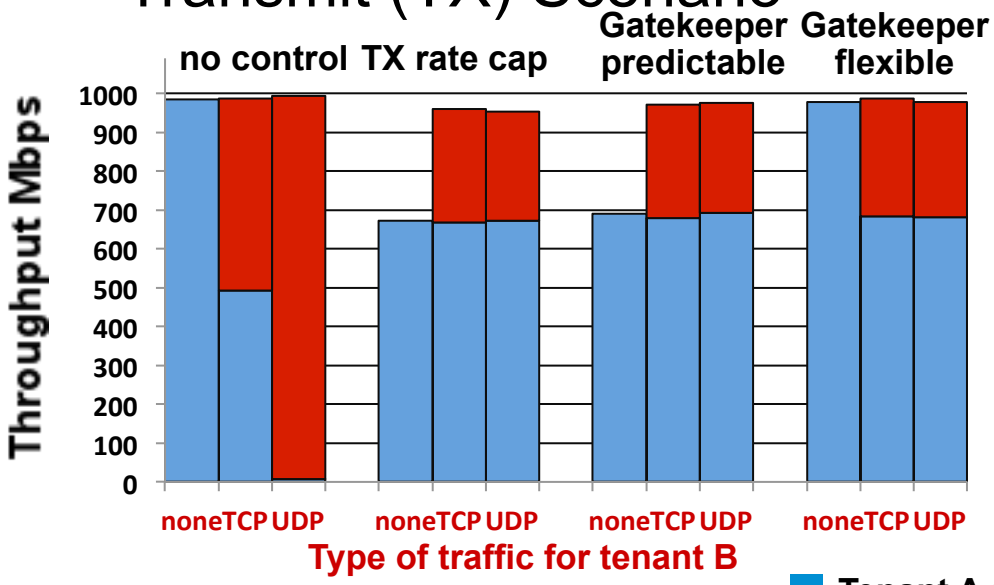
Tenant A (TCP) Tenant B

Example – Results *with* Gatekeeper

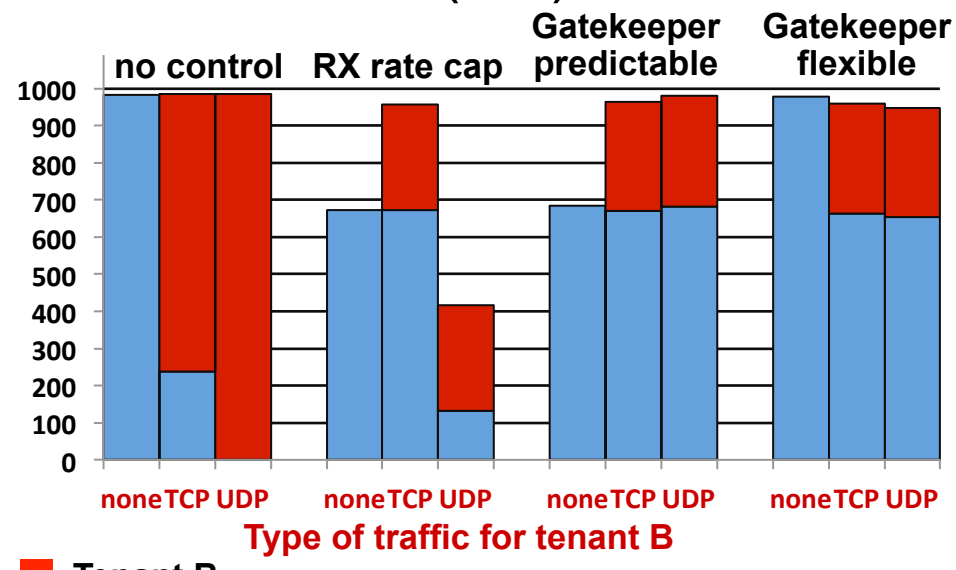




Transmit (TX) Scenario



Receive (RX) Scenario



Tenant A (TCP) Tenant B

Summary

- Gatekeeper provides network bandwidth guarantee at the server virtualization layer
  - Extends hypervisor to control RX bandwidth
- Prototype implemented and used to demonstrate Gatekeeper in simple scenario
- Future work
  - Evaluate Gatekeeper at larger scales
    - HP Labs Open Cirrus testbed (100+ nodes)
  - Further explore the design space
    - Functions to decrease/increase rate, etc
  - Evaluate Gatekeeper with more realistic benchmarks and applications

Gatekeeper: Supporting Bandwidth Guarantees for Multi-tenant Datacenter Networks

Contacts:

{hsr,dorgival}@dcc.ufmg.br

{yoshio_turner,joserenato.santos}@hp.com

Acknowledgements:

WIOV 2011, Portland, OR