Participatory Networking:
An API for Application Control of SDNs

Andrew Ferguson, Arjun Guha, Chen Liang,
Rodrigo Fonseca, and Shriram Krishnamurthi

BROWN Cornell

Participatory Networking

1. SSHGuard
2. Ekiga
3. ZooKeeper
4. Hadoop

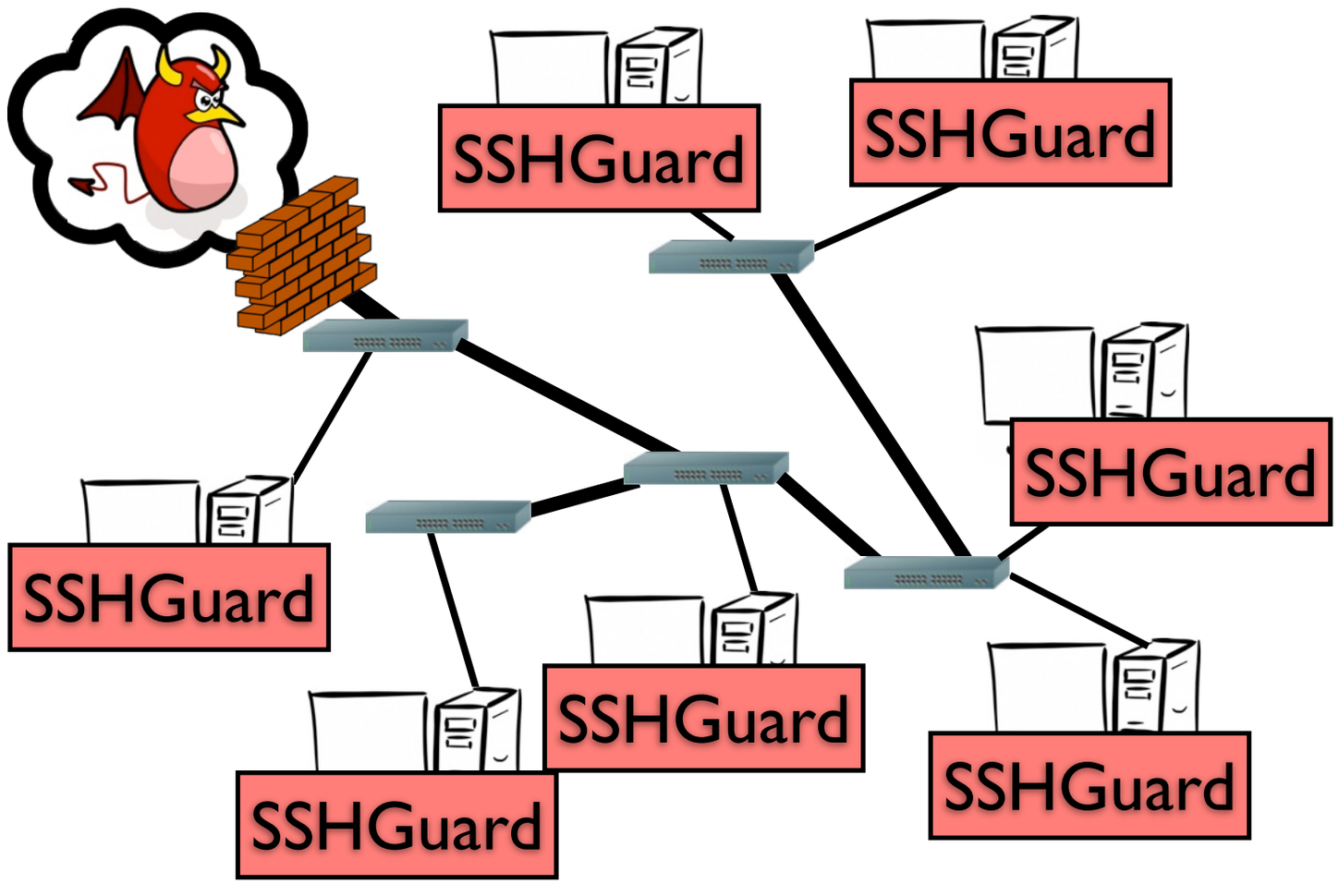
Motivation

1. SSHGuard — blocks hosts in response to login attempts

2. Ekiga — uses knowledge from host OS

3. ZooKeeper — prefers to deny traffic close to source

4. Hadoop



1. SSHGuard

2. Ekiga

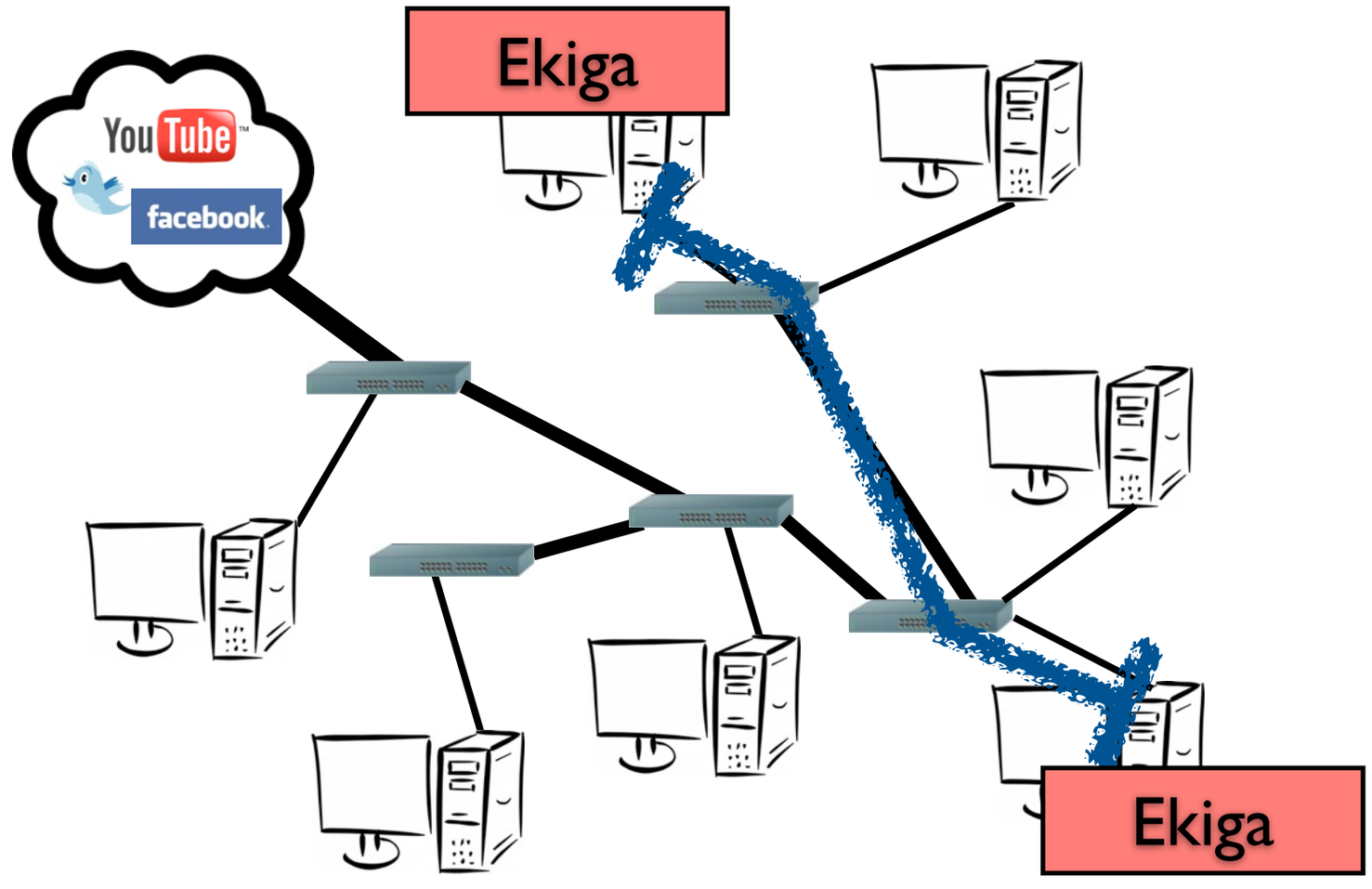
3. ZooKeeper

4. Hadoop

open source VOIP client

network needs dictated by end-user

prefers to reserve bandwidth



1. SSHGuard

2. Ekiga

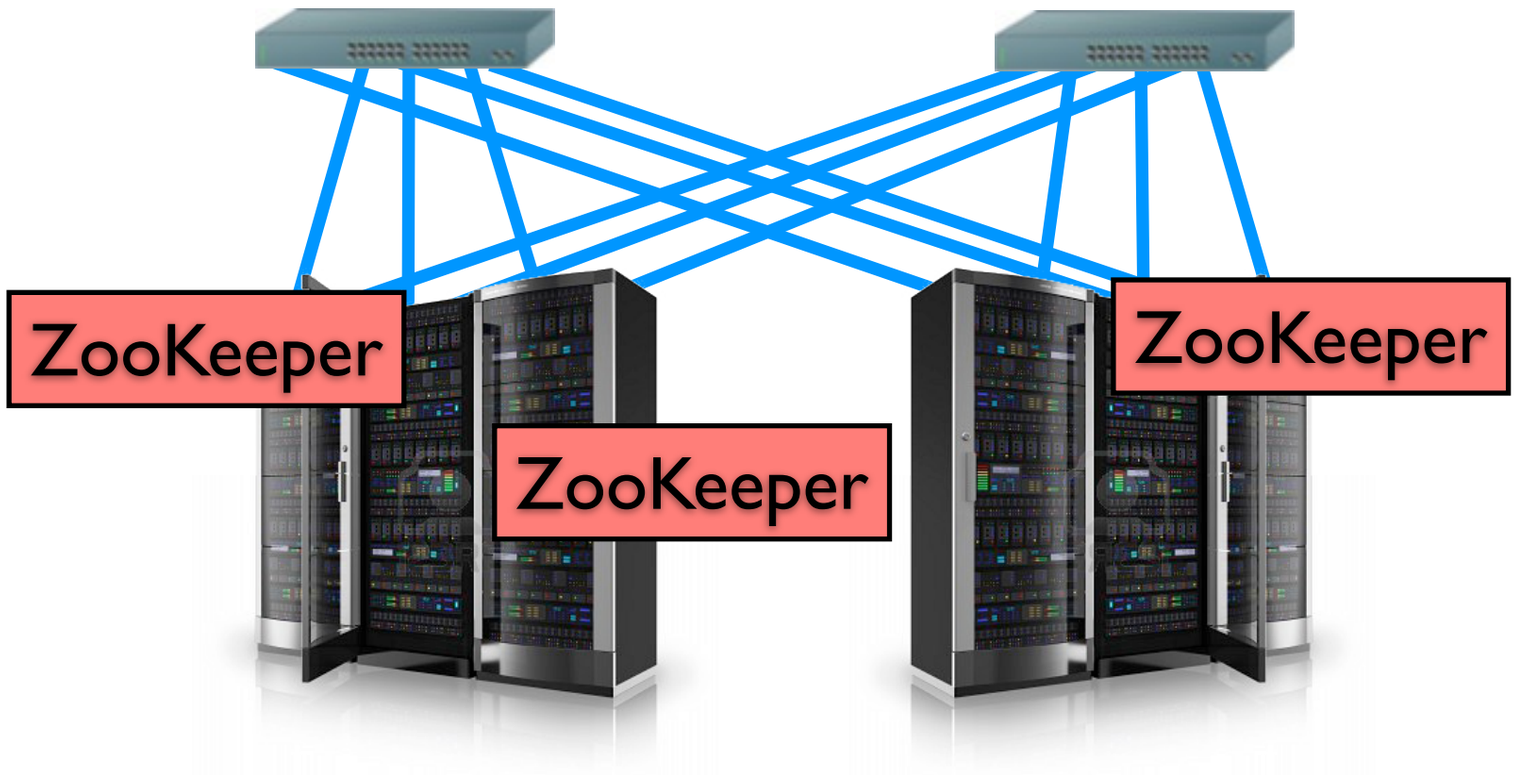
3. ZooKeeper

4. Hadoop

Paxos-like coordination service

network needs dictated by placement

prefers high-priority switch queues



1. SSHGuard

2. Ekiga

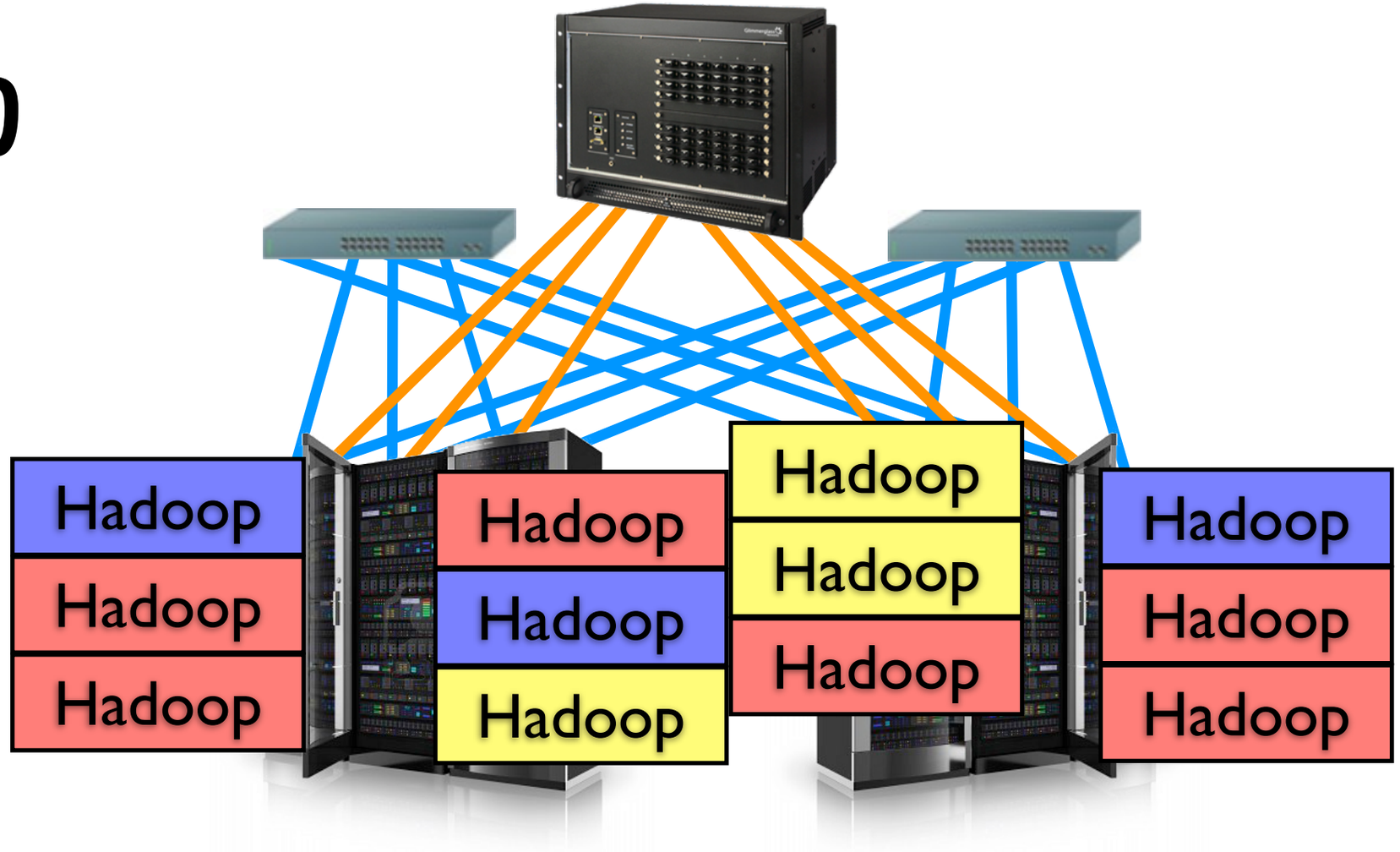
3. ZooKeeper

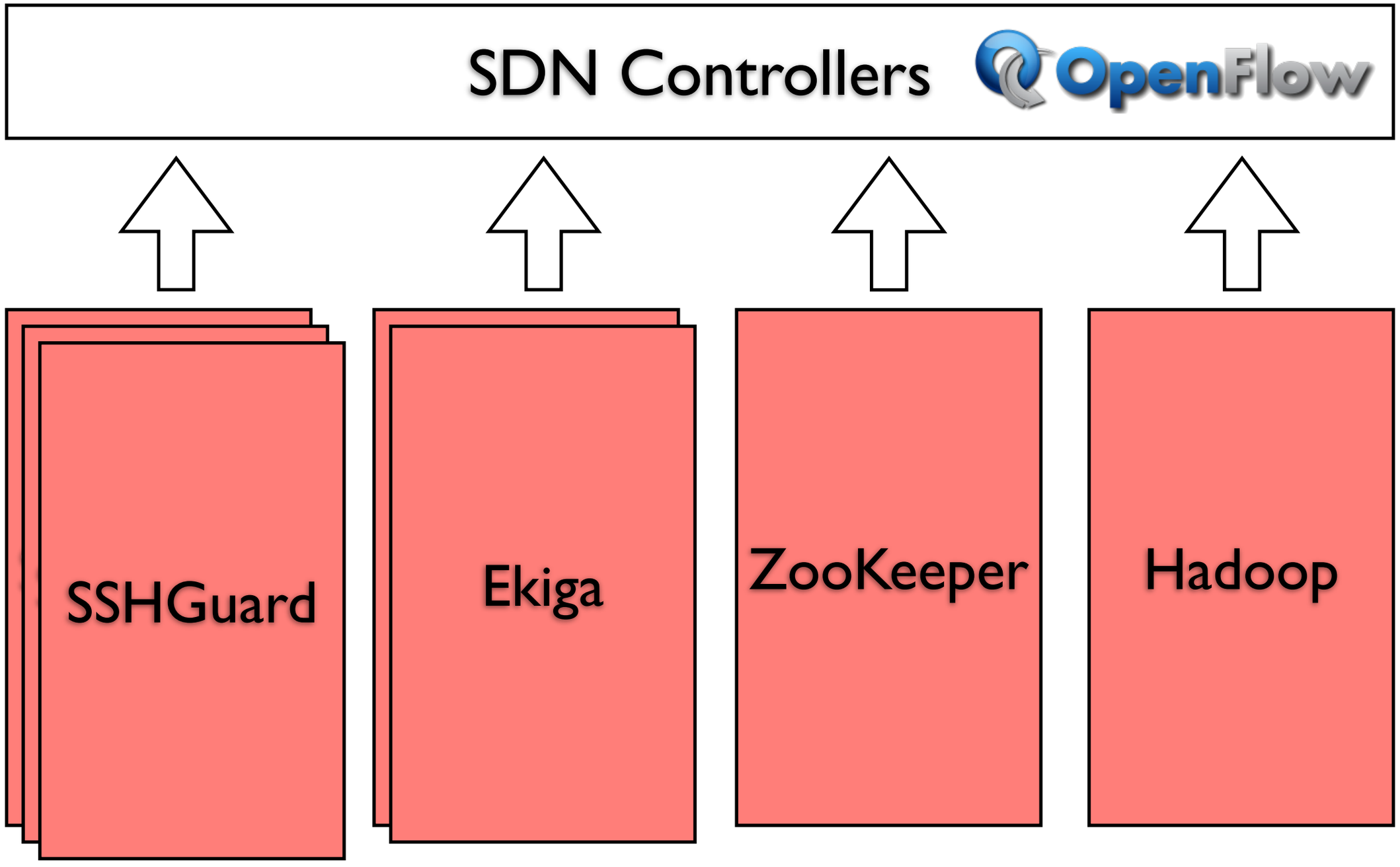
4. Hadoop

open source data processing platform

network weights known by scheduler

prefers to reserve bandwidth




SDN Controllers
OpenFlow
SSHGuard
Ekiga
ZooKeeper
Hadoop

THE HAYMARKET RIOT. The Explosion and the Conflict.

1. decompose control and visibility

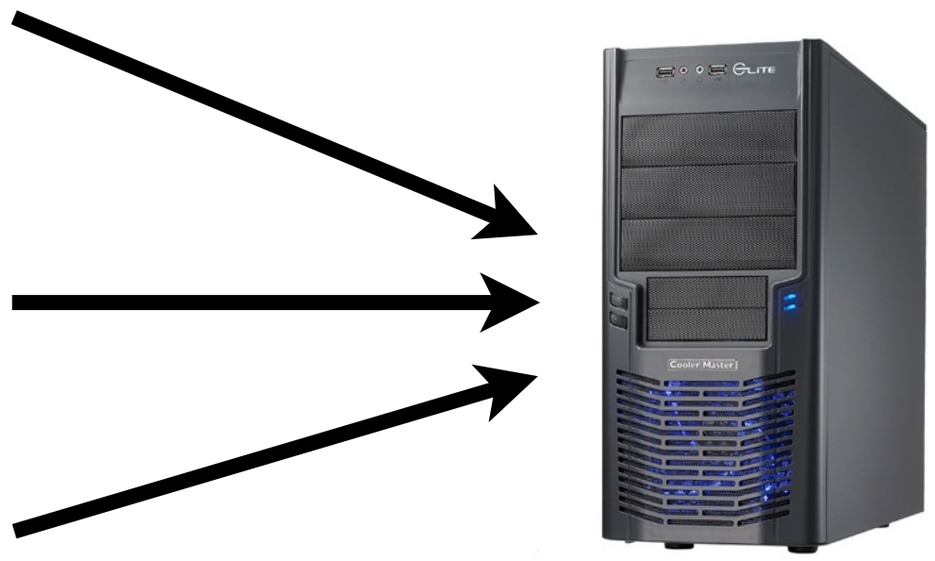
2. resolve conflicts between requests

Challenges

Participatory Networking

Participatory Networking

1. Requests
2. Hints
3. Queries



PANE

Participatory Networking

- End-user API for SDNs
- Exposes existing mechanisms
- No effect on unmodified applications

Decomposing Control

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Principals Alice Bob Hadoop	Privileges *deny, allow* *bandwidth: 5Mb/s* *limit: 10Mb/s* *hint* *query*

Shares

Flowgroup

src=128.12/16 ∧ dst.port ≤1024

Principals

Alice

Bob

Hadoop

Privileges

deny, allow

bandwidth: 5Mb/s

limit: 10Mb/s

hint

query

Shares

Flowgroup

src=128.12/16 ∧ dst.port ≤1024

Principals	Privileges
Alice Bob Hadoop	*deny, allow* *bandwidth: 5Mb/s* *limit: 10Mb/s* *hint* *query*

Shares

Flowgroup

src=128.12/16 ∧ dst.port ≤1024

Principals

Alice

Bob

Hadoop

Privileges

deny, allow

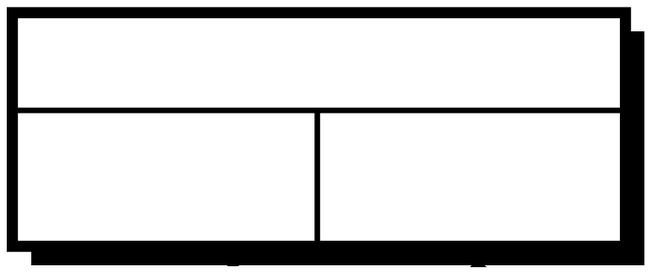
bandwidth: 5Mb/s

limit: 10Mb/s

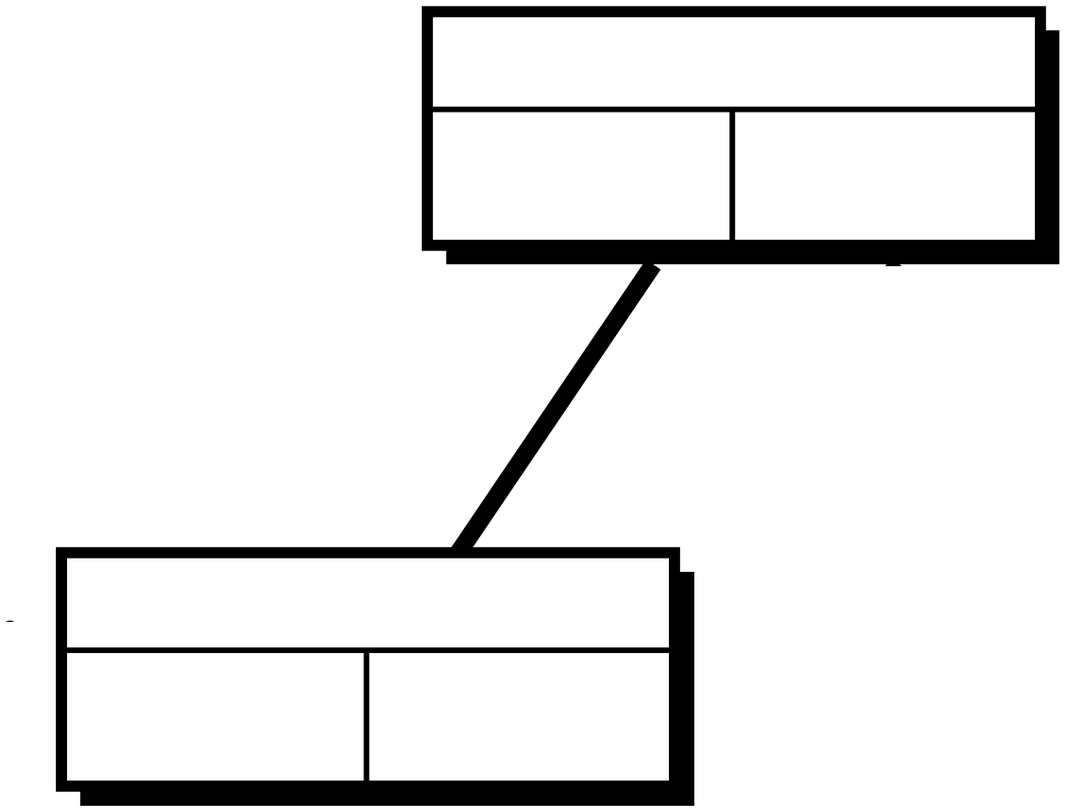
hint

query

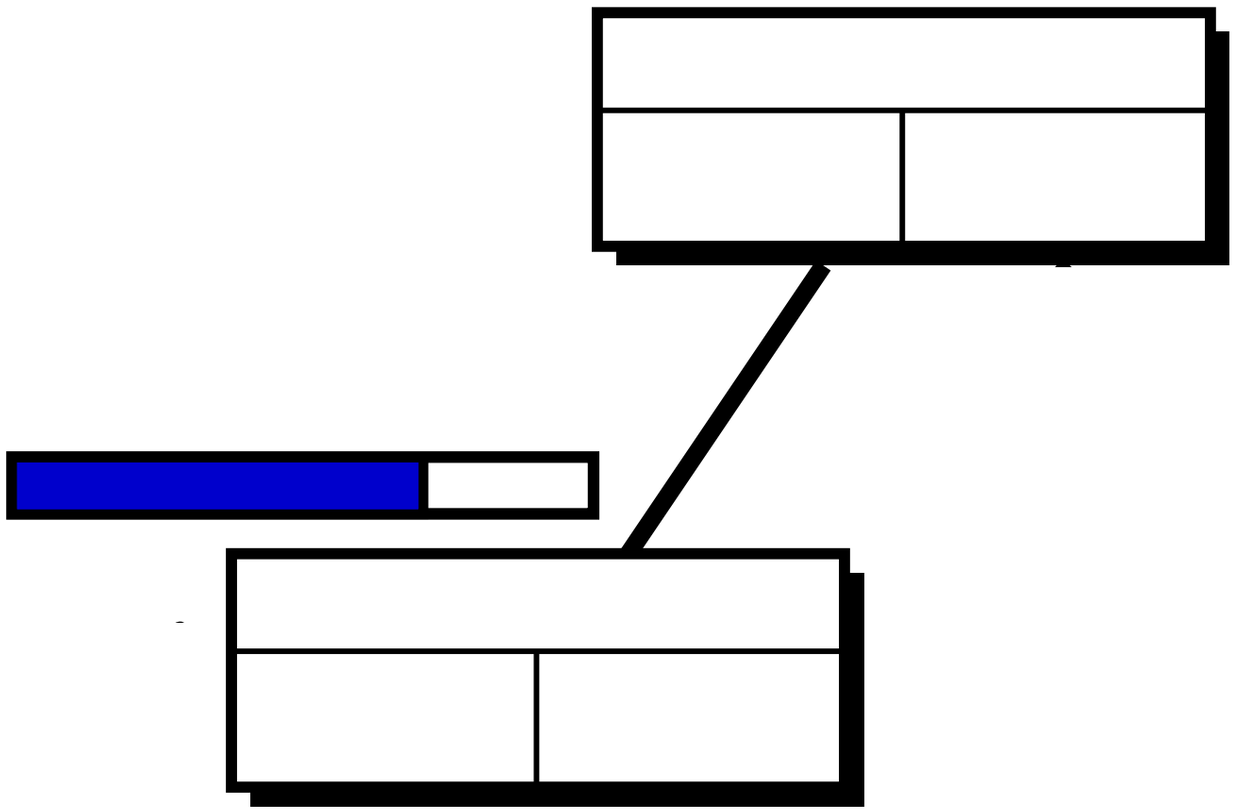
Shares



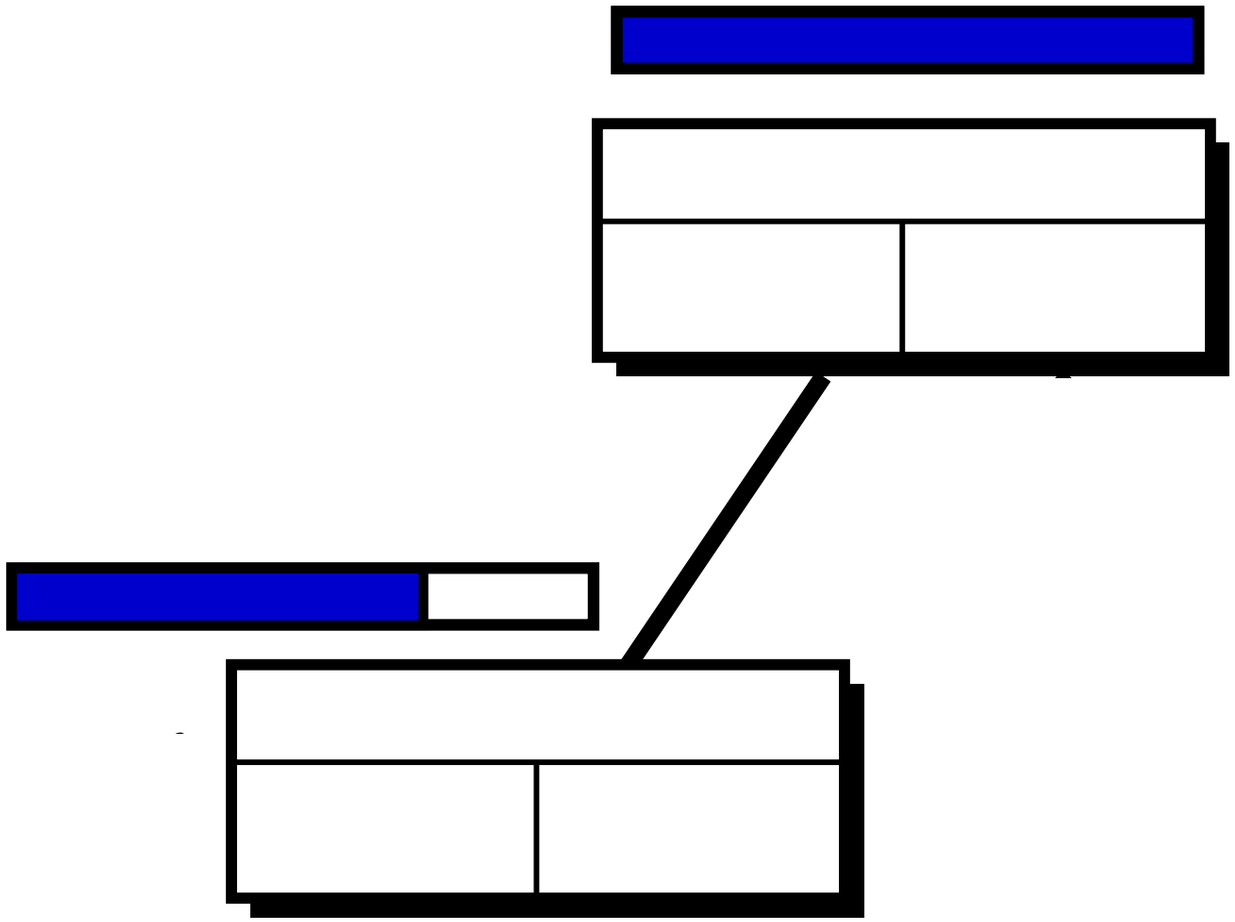
Share Tree



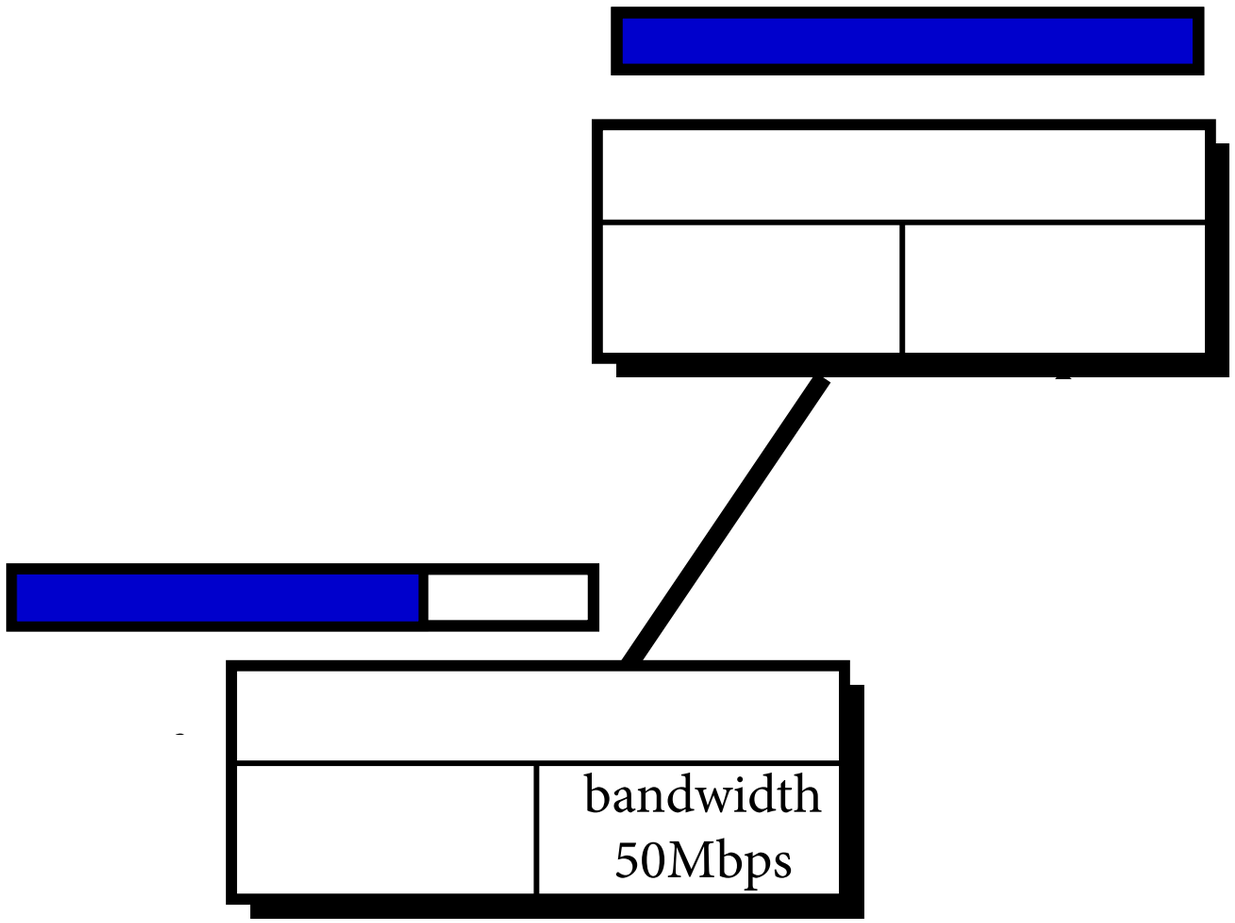
Share Tree



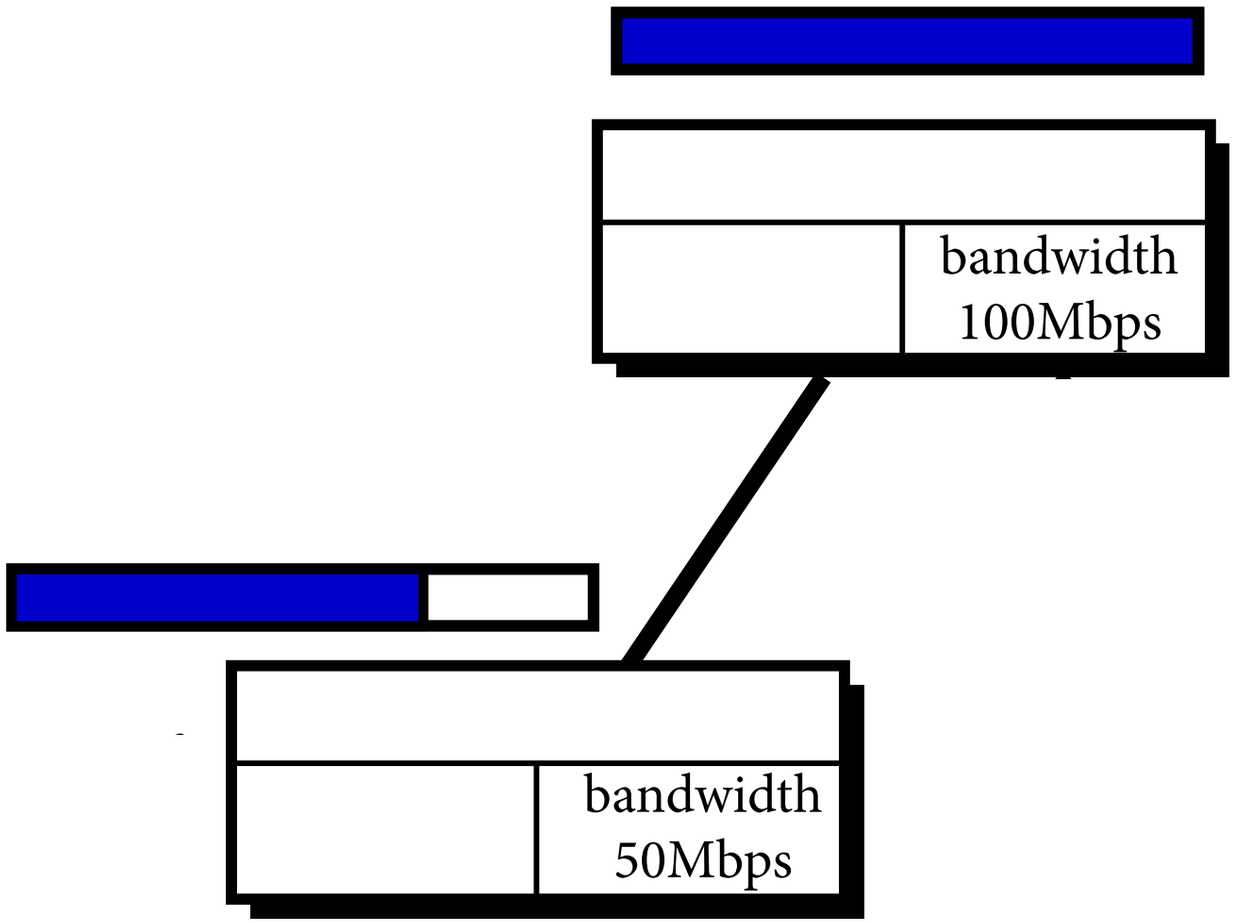
Share Tree



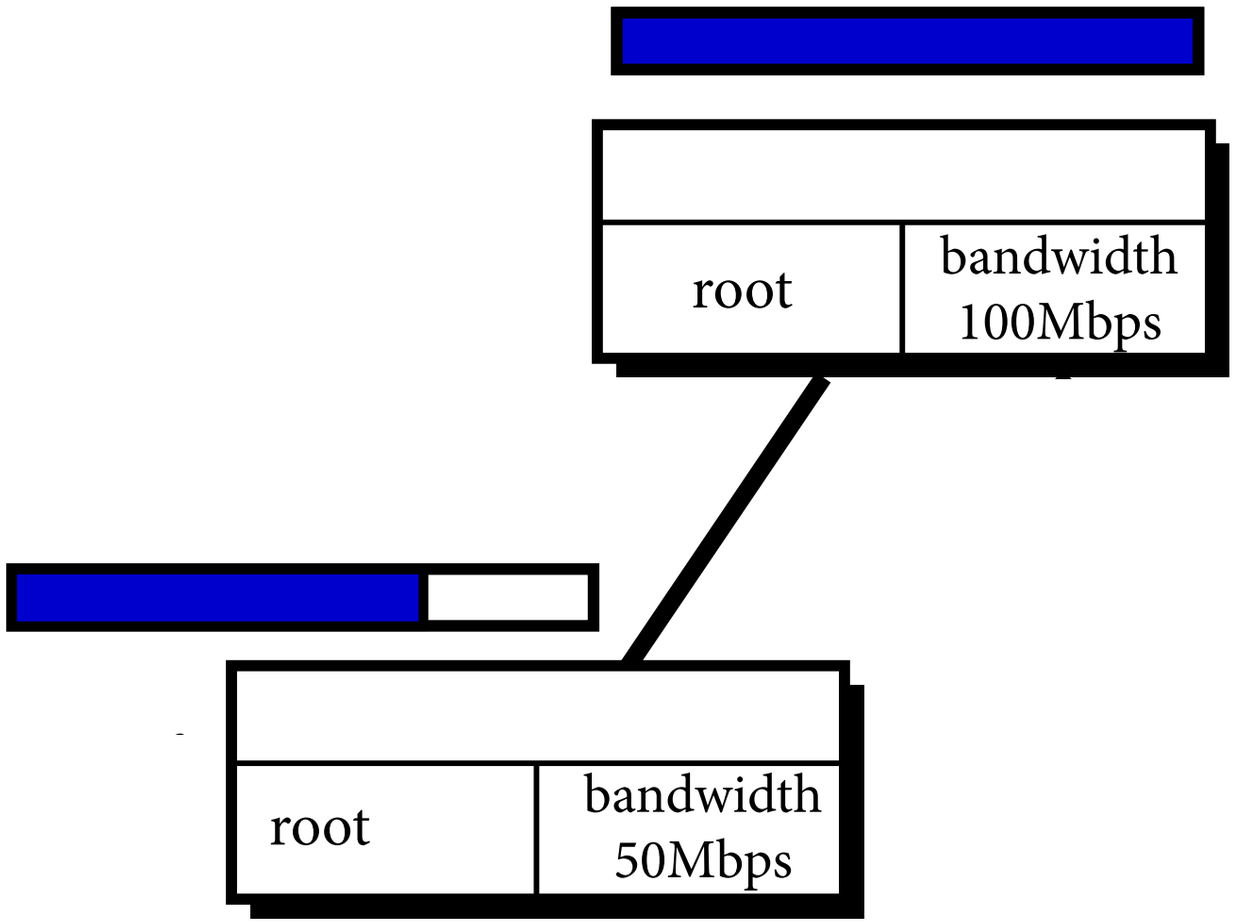
Share Tree



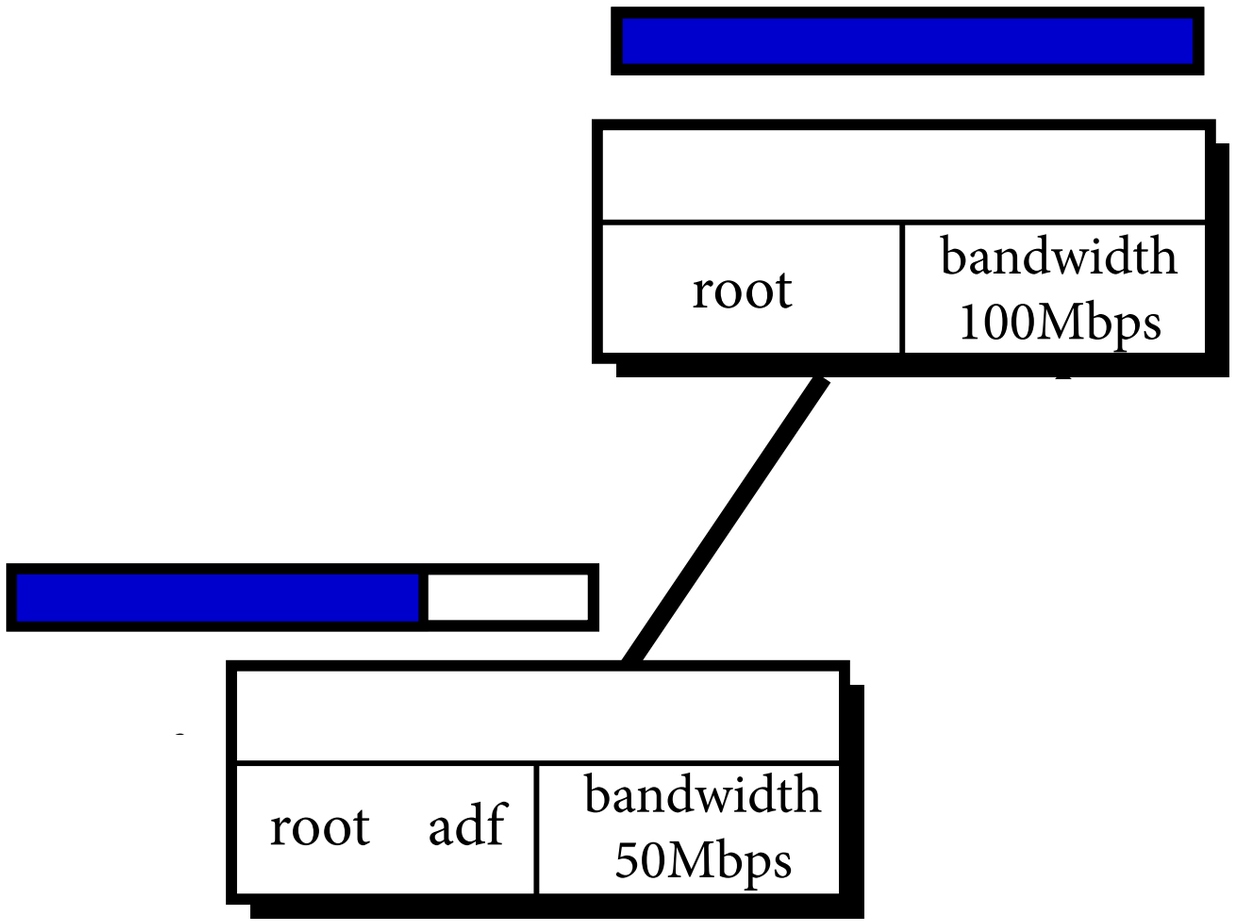
Share Tree



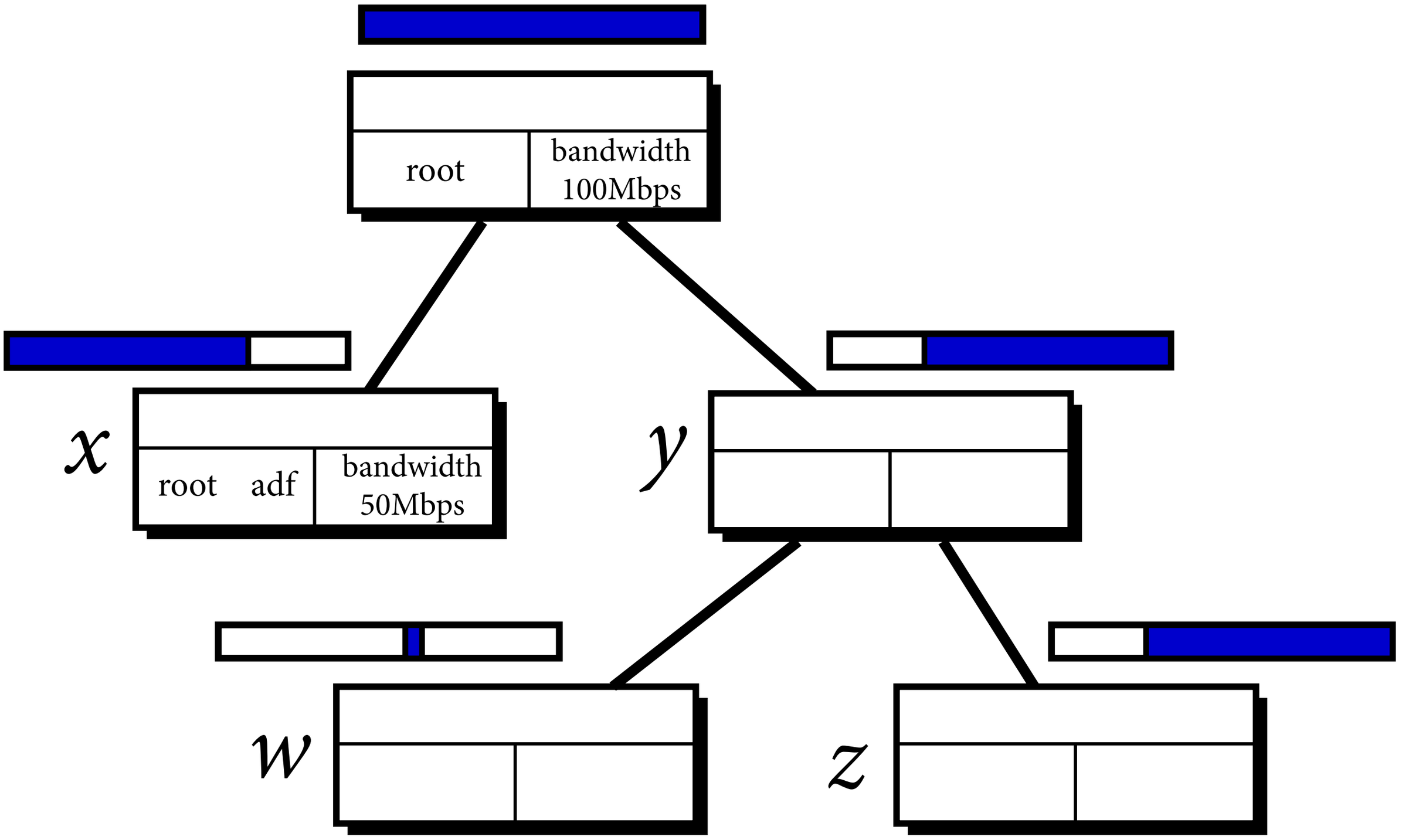
Share Tree



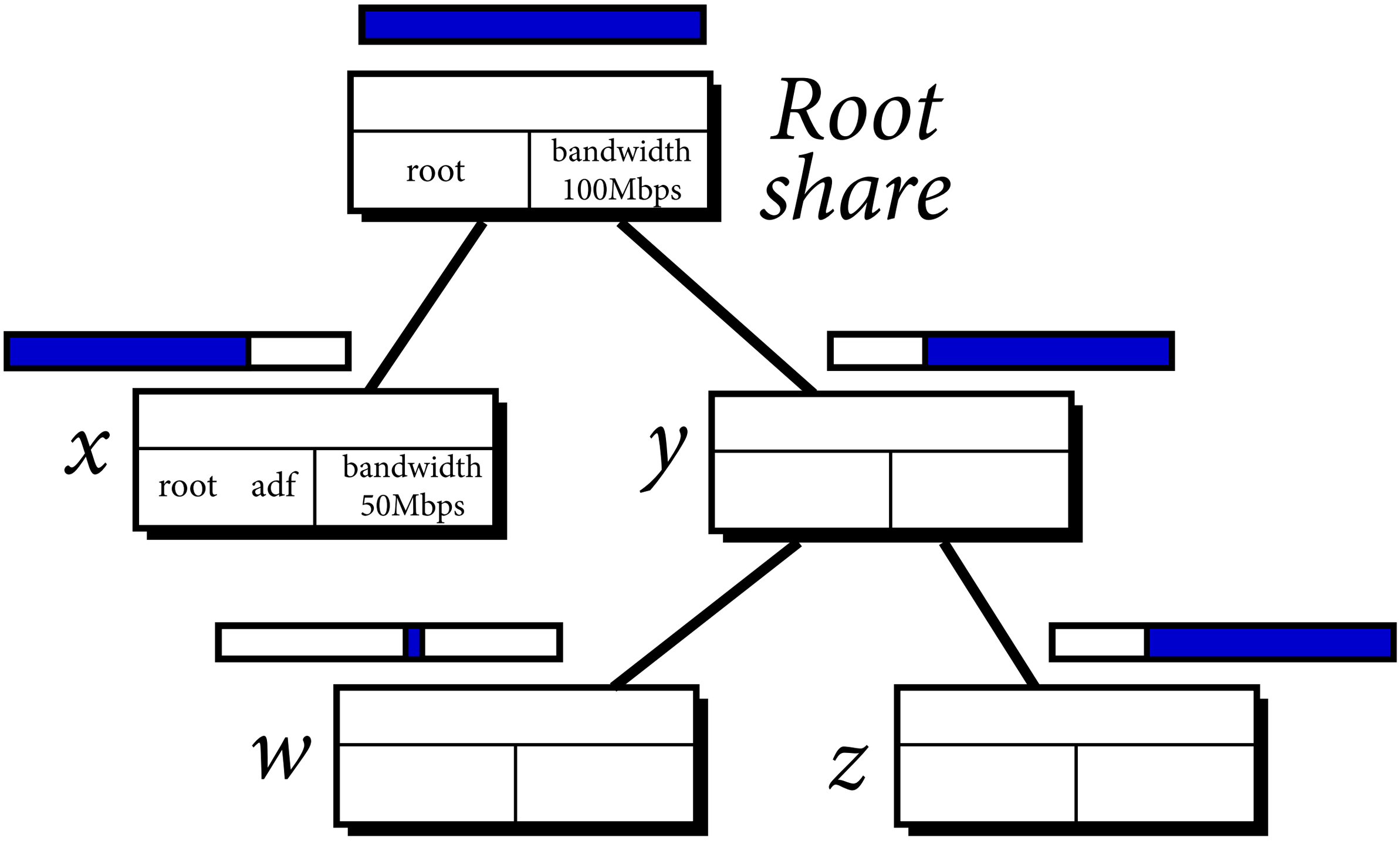
Share Tree



Share Tree



Share Tree


root
bandwidth
100Mbps
Root
share
x
root adf
bandwidth
50Mbps
y
w
z

Share Tree

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

Reserve 2 Mbps
from now to +5min?

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

Yes

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

This traffic will be short and bursty

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

OK

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

How much web traffic in the last hour?

PANE

Flowgroup src=128.12/16 ∧ dst.port ≤1024	
Speakers Alice Bob	Privileges deny, allow bandwidth: 5Mb/s limit: 10Mb/s *hint* *query*

67,560 bytes

PANE

Current: 0 Mbps

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

Current: 0 Mbps

ShareB

bandwidth
100Mbps

Current: 0 Mbps

PANE

Current: 0 Mbps

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

Current: 0 Mbps

ShareB

bandwidth
100Mbps

Current: 0 Mbps

PANE

Current: 0 Mbps

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

Current: 0 Mbps

ShareB

bandwidth
100Mbps

Current: 0 Mbps

PANE

Current: 0 Mbps

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

ShareB

bandwidth
100Mbps

Current: 0 Mbps

Current: 0 Mbps

Reserve 80 Mbps?

PANE

Current: **80 Mbps**

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

ShareB

bandwidth
100Mbps

Current: **80 Mbps**

Current: 0 Mbps

Yes

PANE

Current: **80 Mbps**

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

Current: **80 Mbps**

ShareB

bandwidth
100Mbps

Current: 0 Mbps

Reserve 50 Mbps?

PANE

Current: **80 Mbps**

bandwidth
100Mbps

Root share

ShareA

bandwidth
100Mbps

Current: **80 Mbps**

ShareB

bandwidth
100Mbps

Current: 0 Mbps

No

PANE

Current: **80 Mbps**

bandwidth
100Mbps

Root share

ShareA

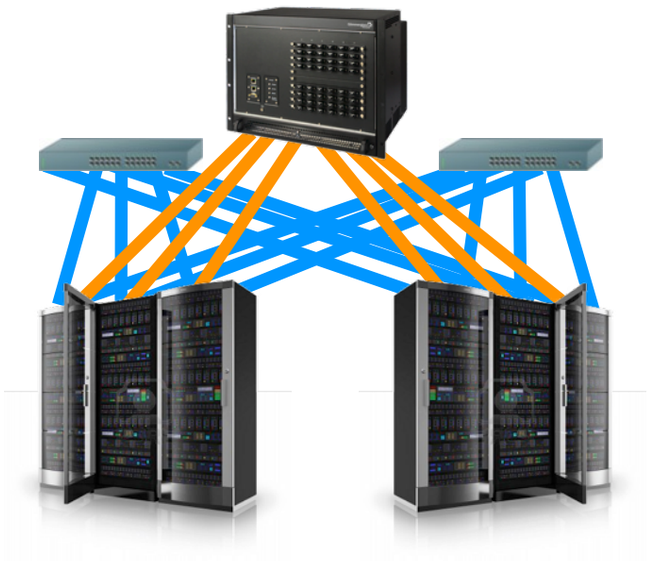
bandwidth
100Mbps

Current: **80 Mbps**

ShareB

bandwidth
100Mbps

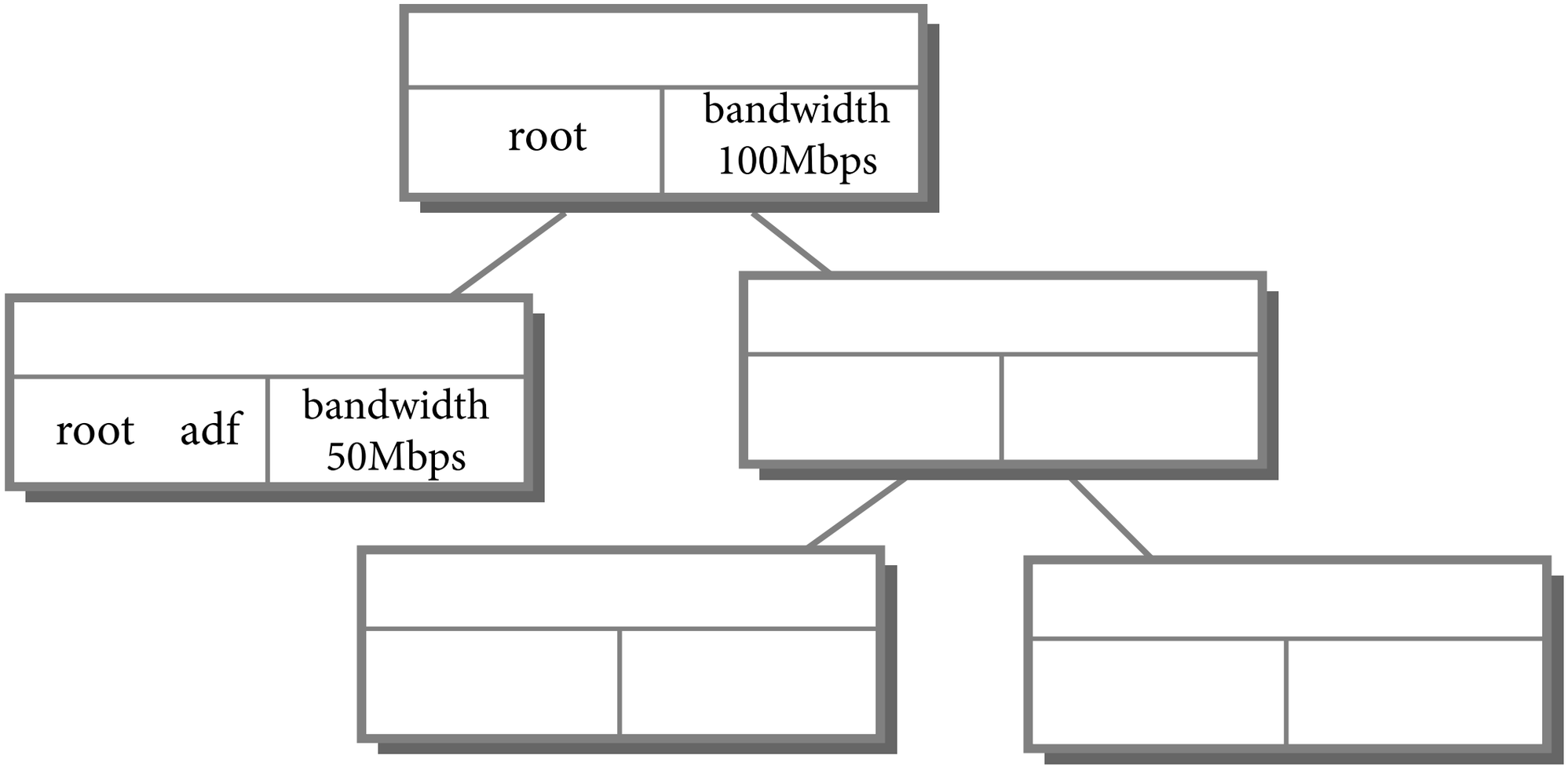
Current: 0 Mbps



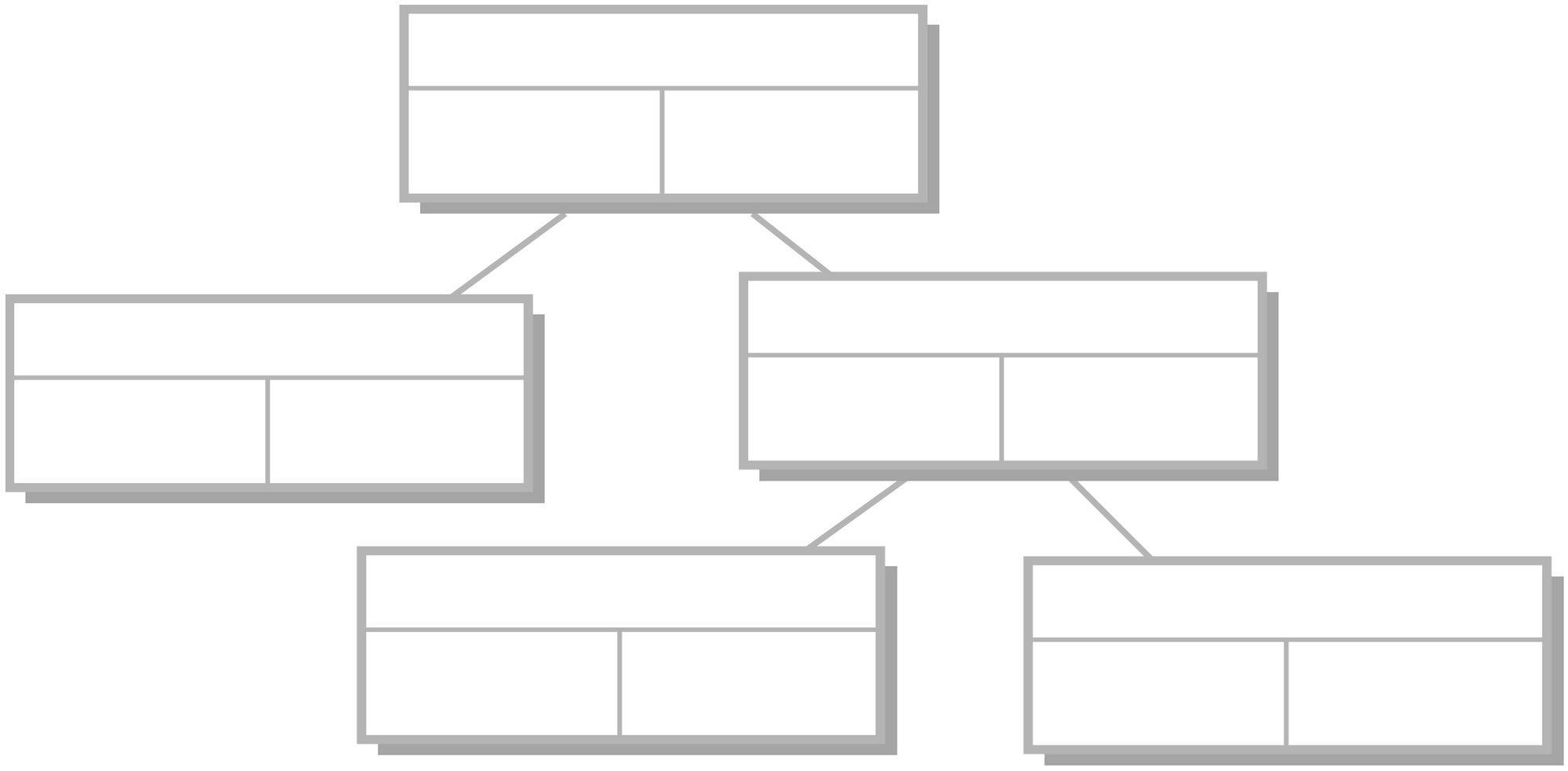
OpenFlow

PANE

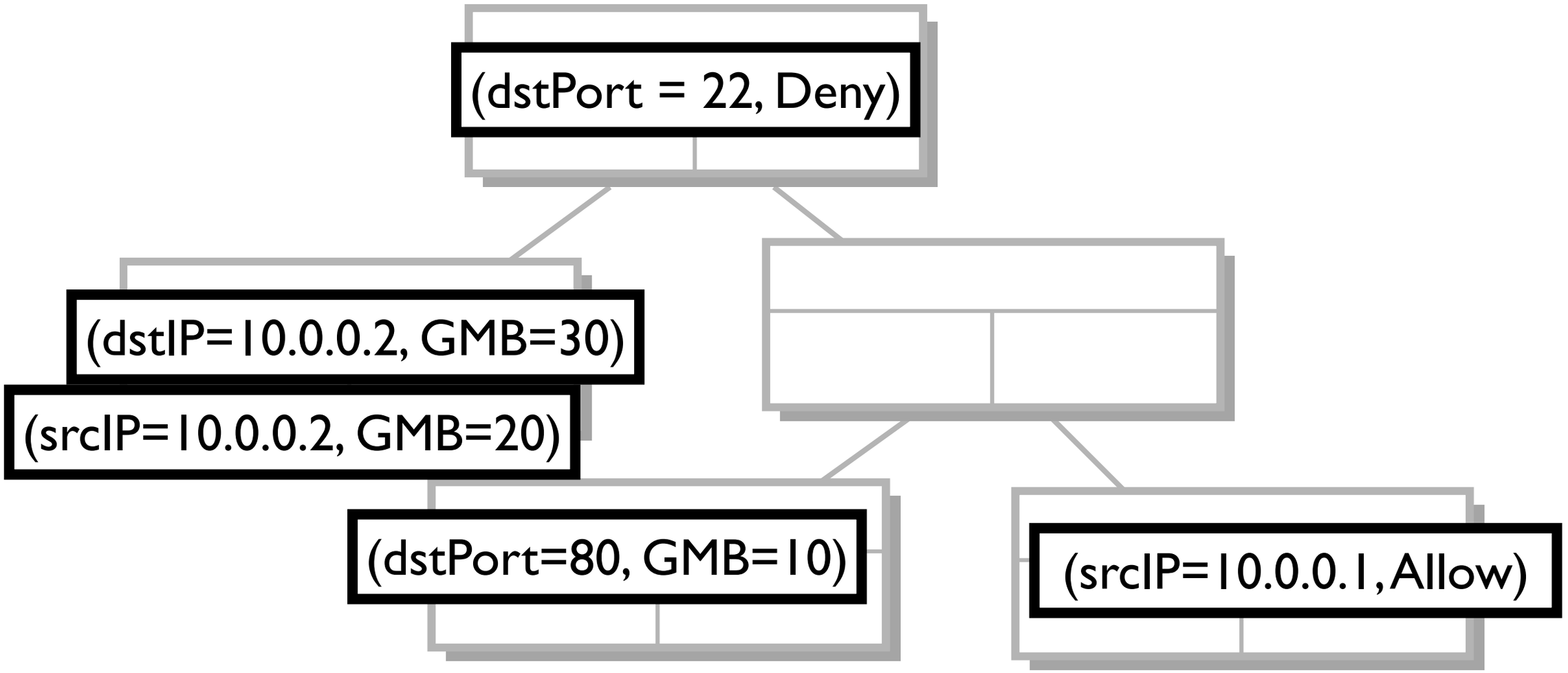
Resolving Conflicts



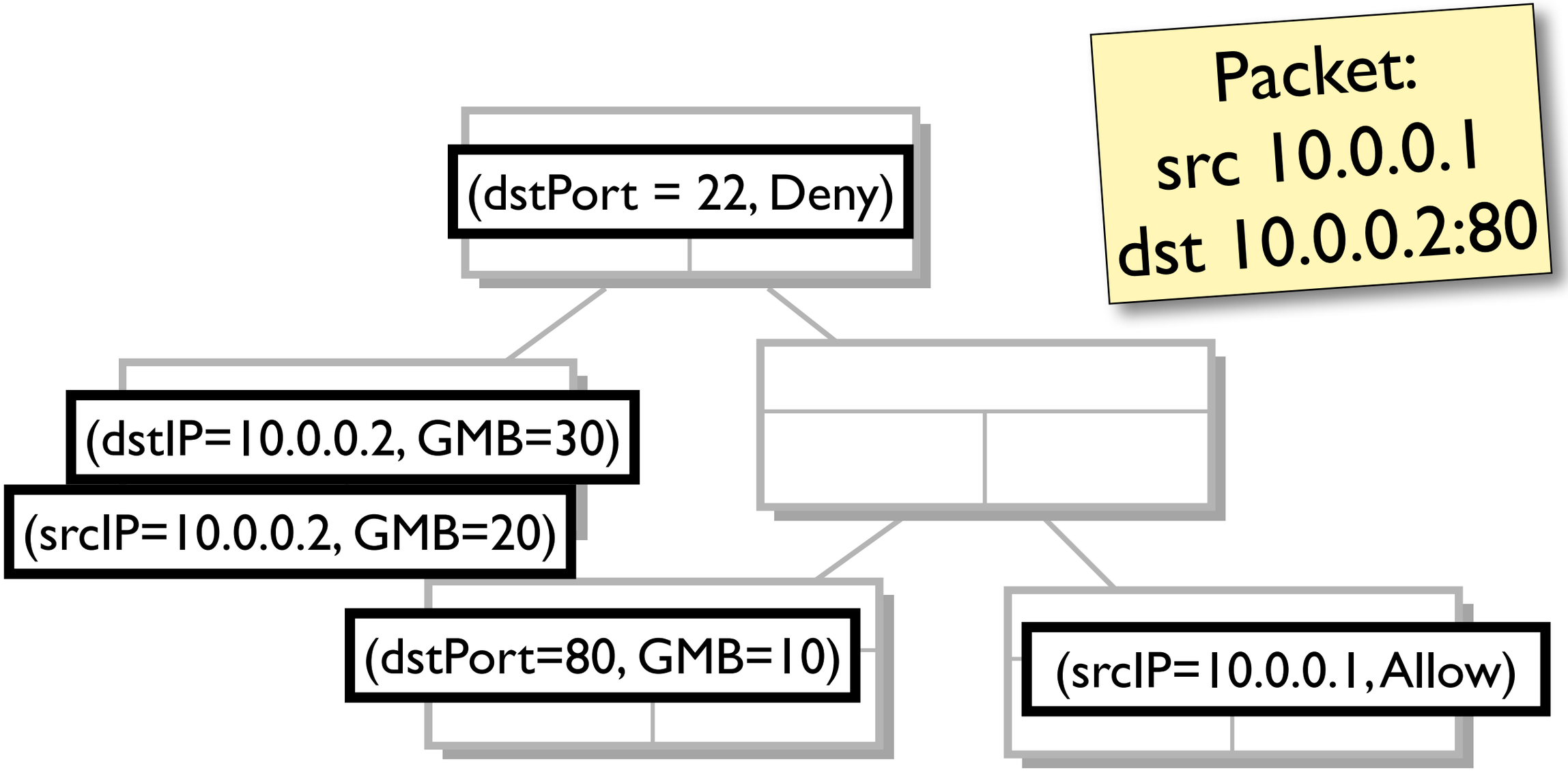
Share Tree



Policy Trees

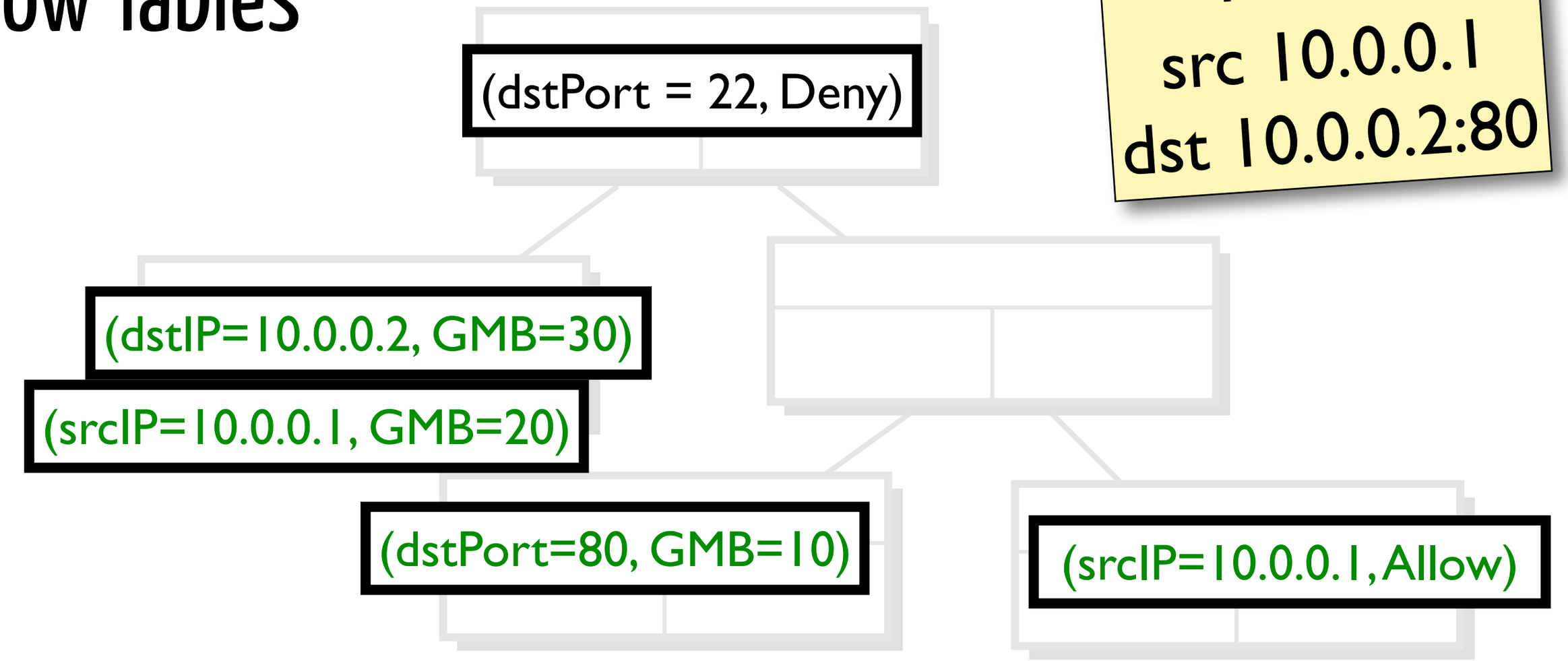


Policy Trees



Policy Trees

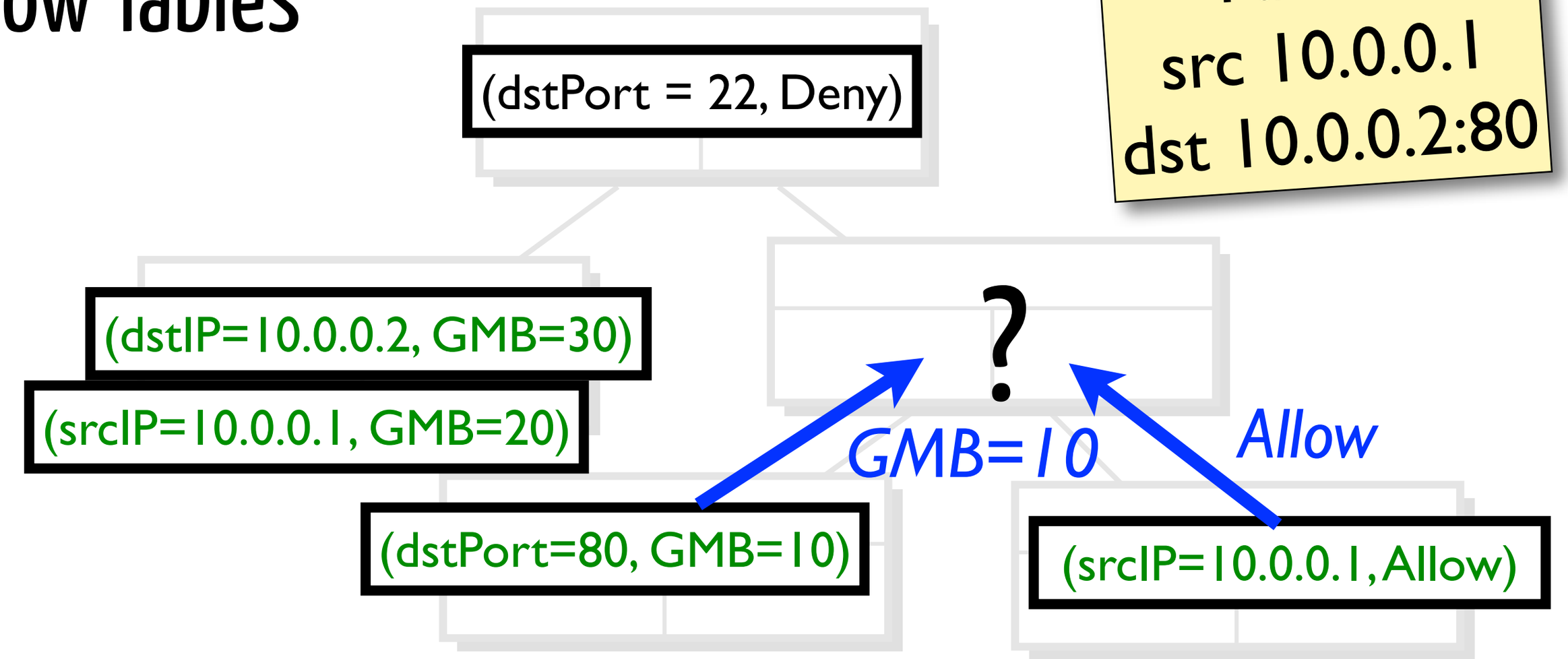
Hierarchical Flow Tables



Packet:
src 10.0.0.1
dst 10.0.0.2:80

Packet Evaluation

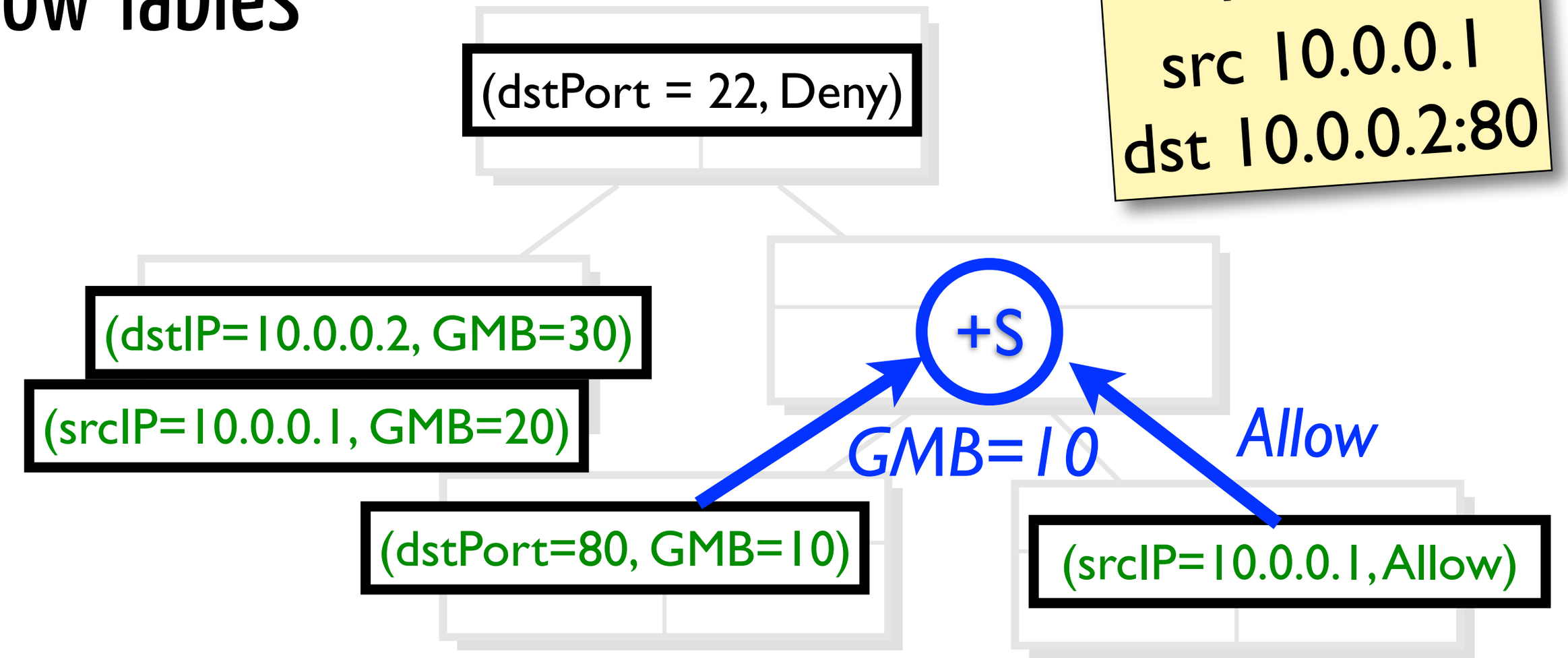
Hierarchical Flow Tables



Packet:
src 10.0.0.1
dst 10.0.0.2:80

Packet Evaluation

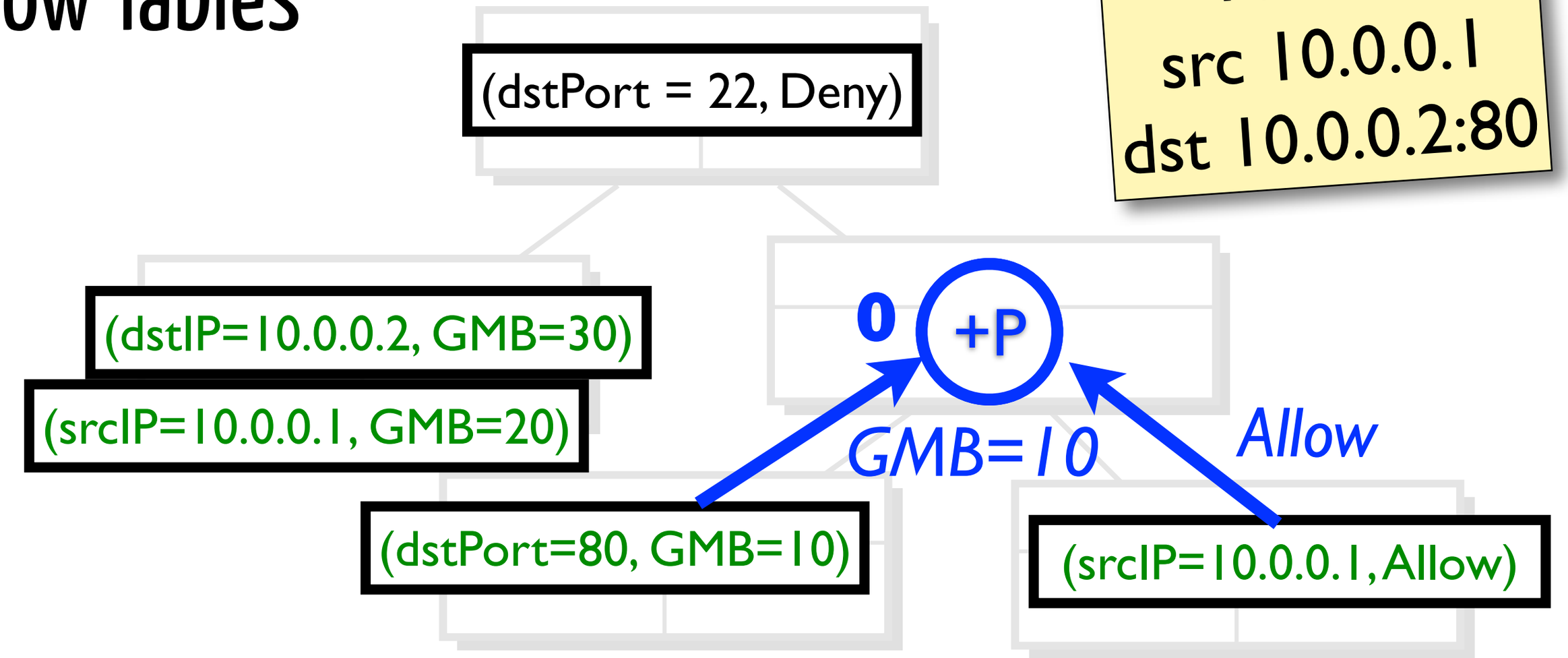
Hierarchical Flow Tables



Packet:
src 10.0.0.1
dst 10.0.0.2:80

Packet Evaluation

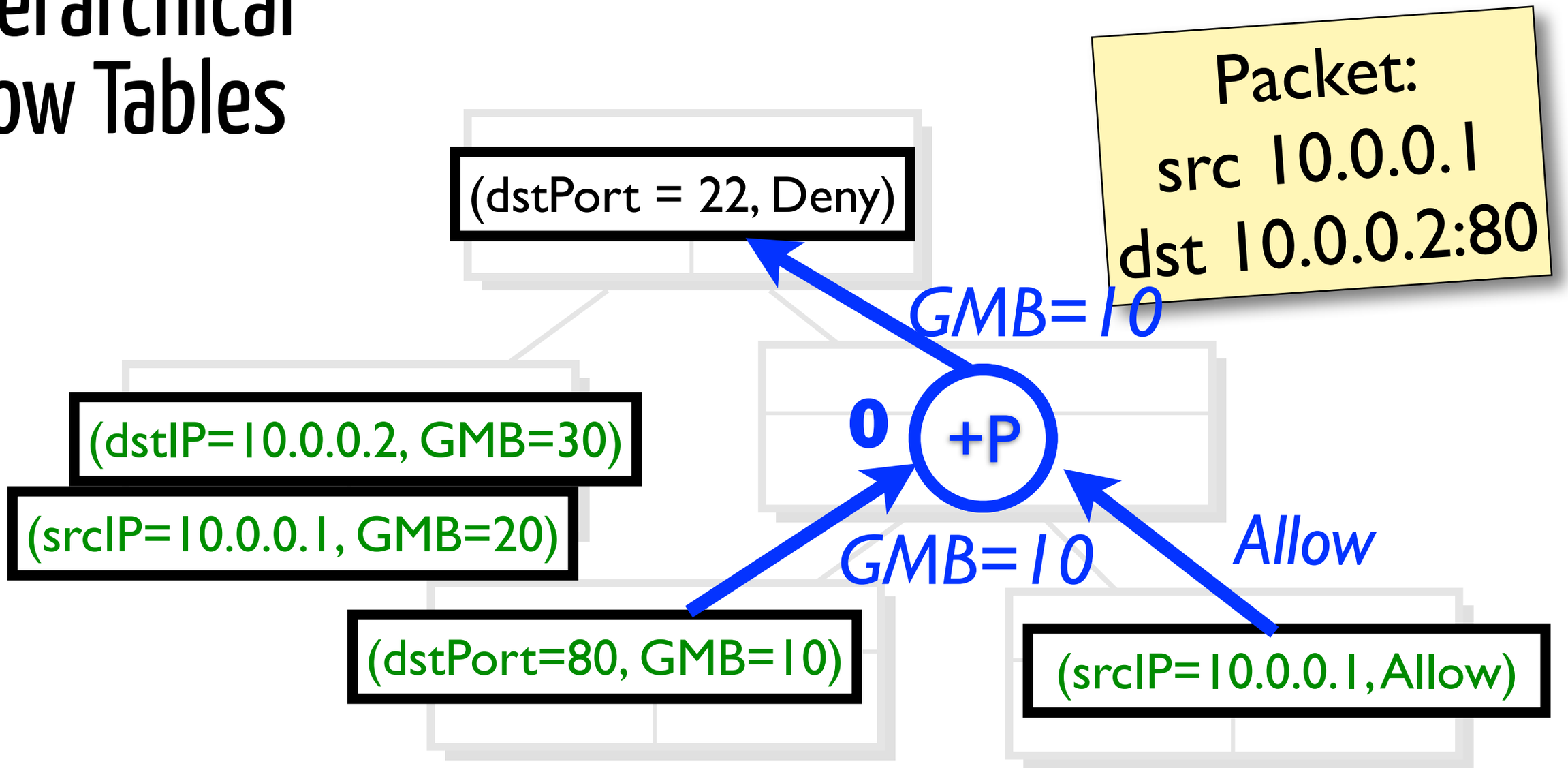
Hierarchical Flow Tables



Packet:
src 10.0.0.1
dst 10.0.0.2:80

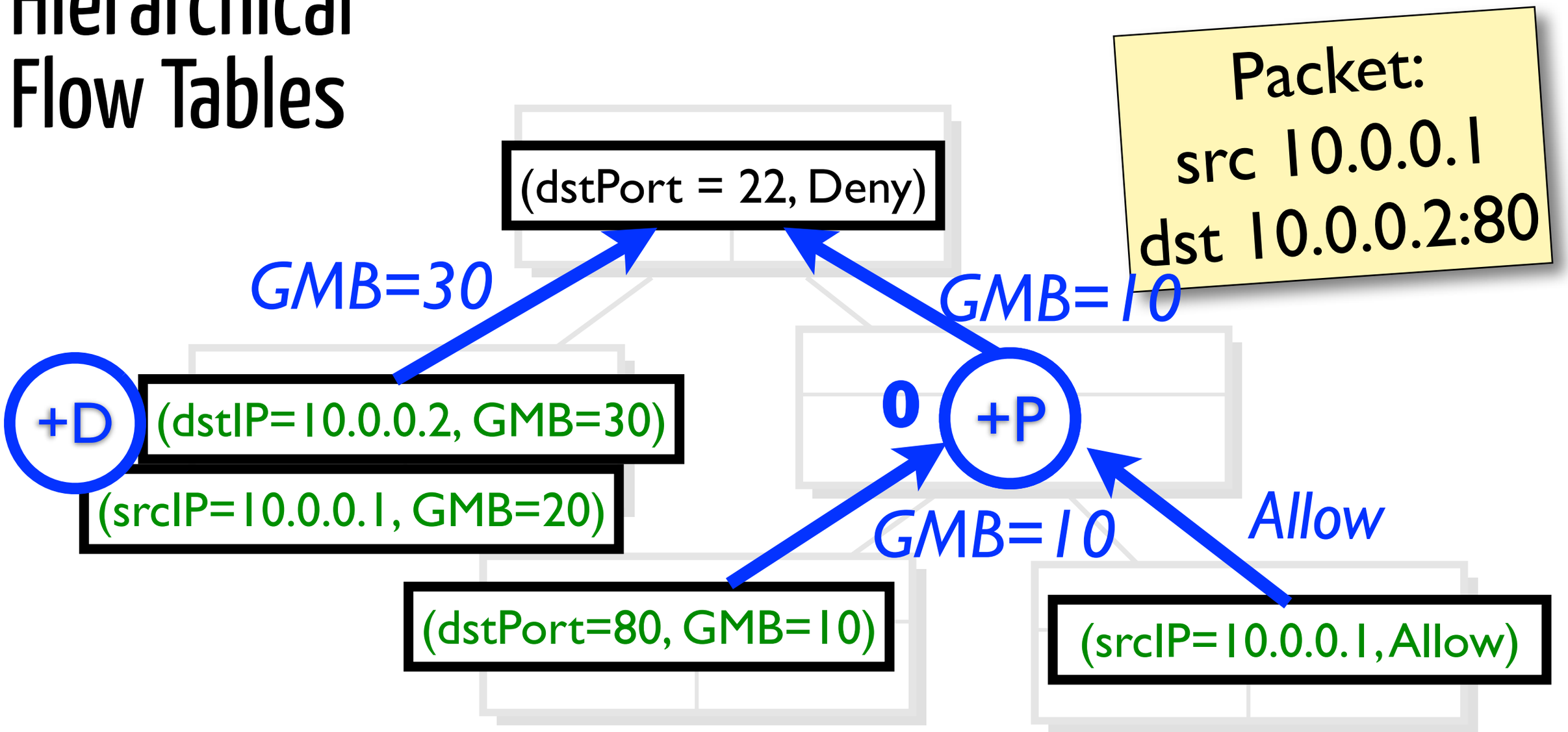
Packet Evaluation

Hierarchical Flow Tables



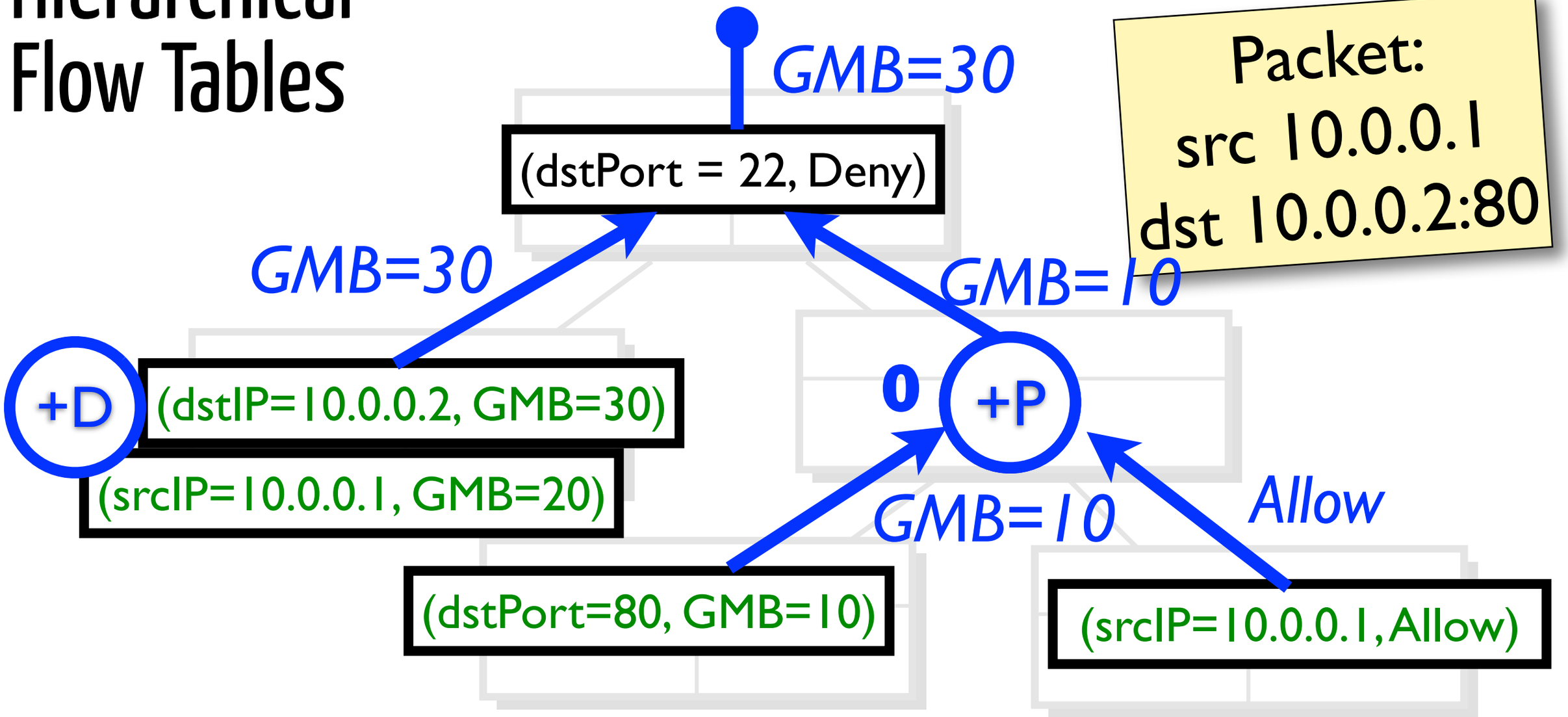
Packet Evaluation

Hierarchical Flow Tables



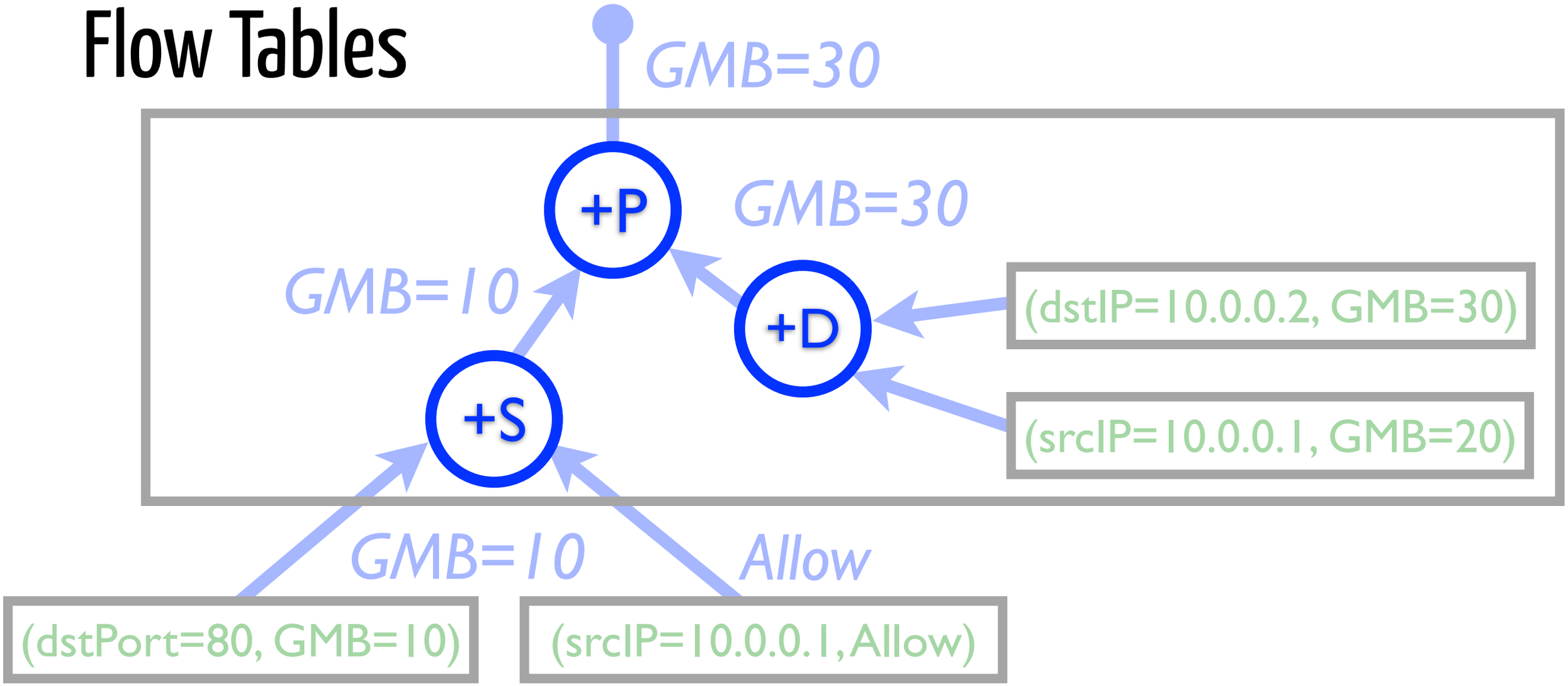
Packet Evaluation

Hierarchical Flow Tables



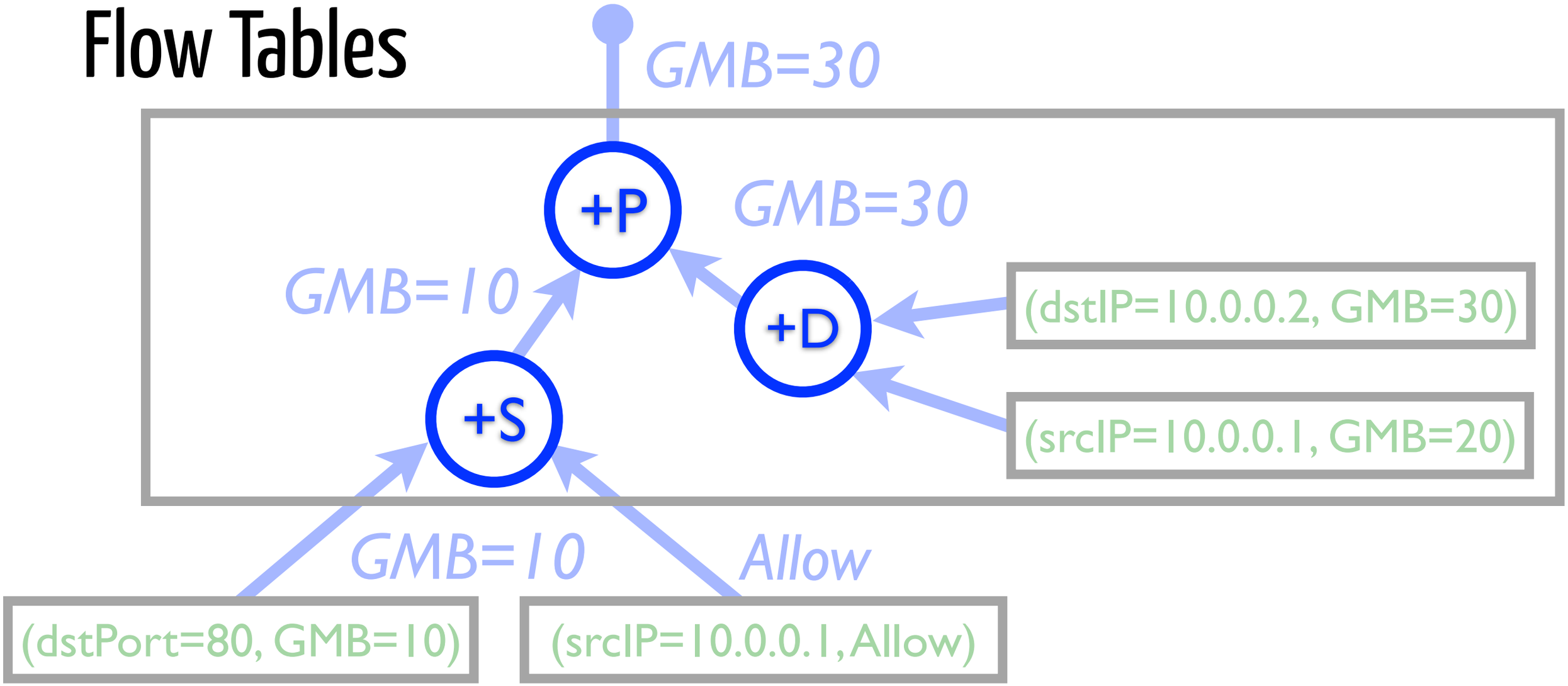
Packet Evaluation

Hierarchical Flow Tables



Conflict Resolution

Hierarchical Flow Tables



Conflict Resolution

Only Requirements:
Associative, **0**-identity

+D *In node*

D and S identical.
Deny overrides Allow.
GMB combines as **max**
Rate-limit combines as **min**

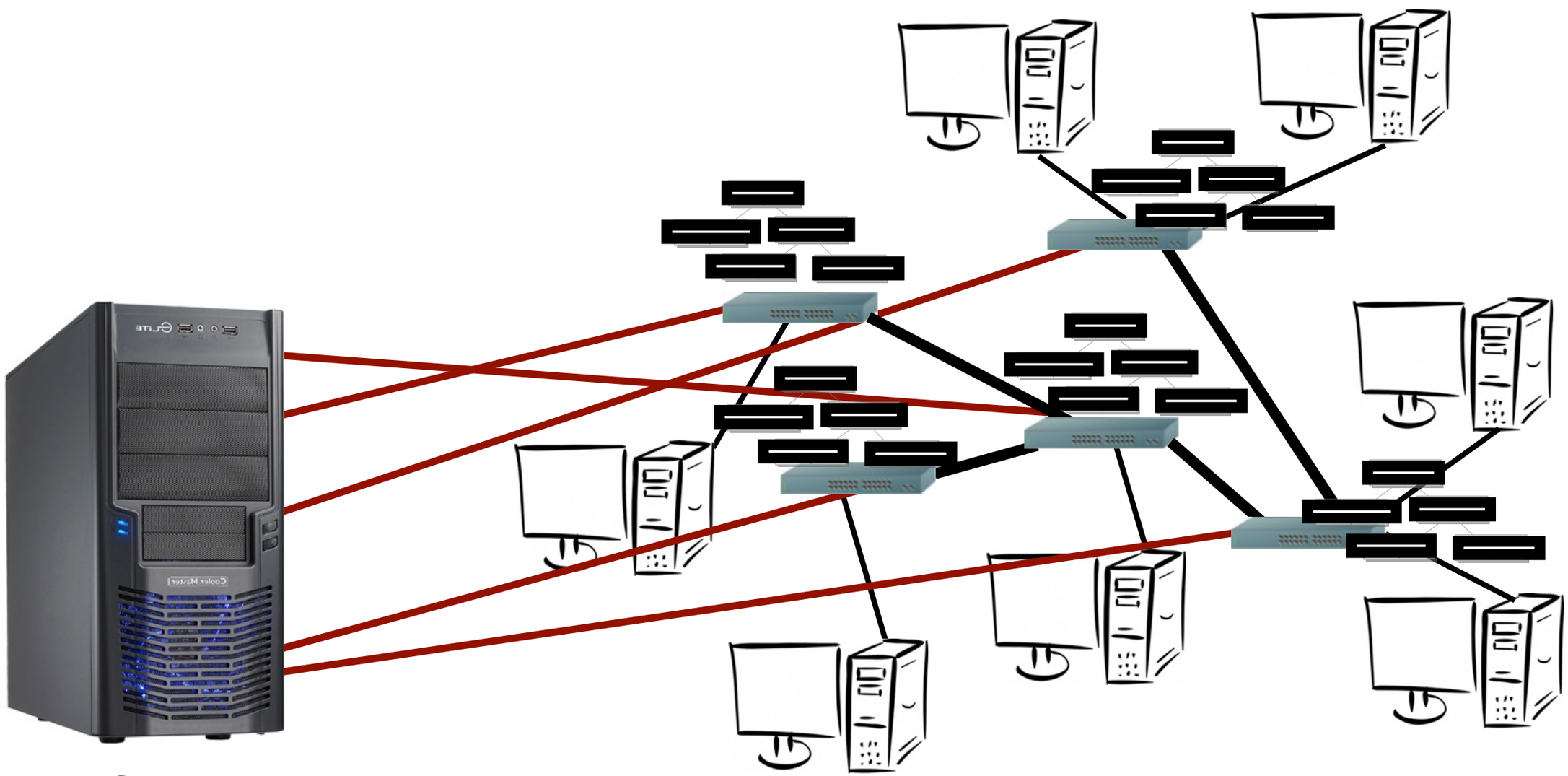
+S *Sibling*

+P *Parent-Sibling*

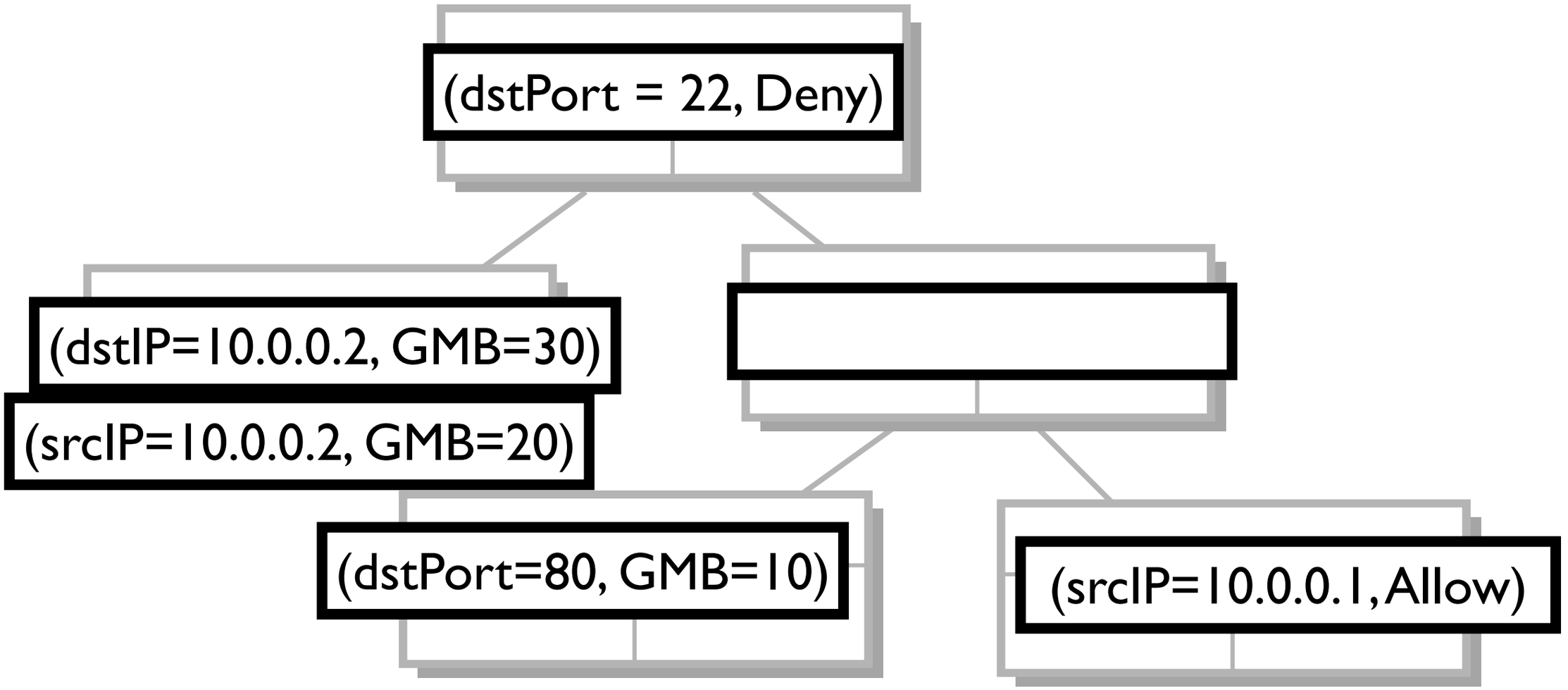
Child overrides Parent
for Access Control
GMB combines as **max**
Rate-limit combines as **min**

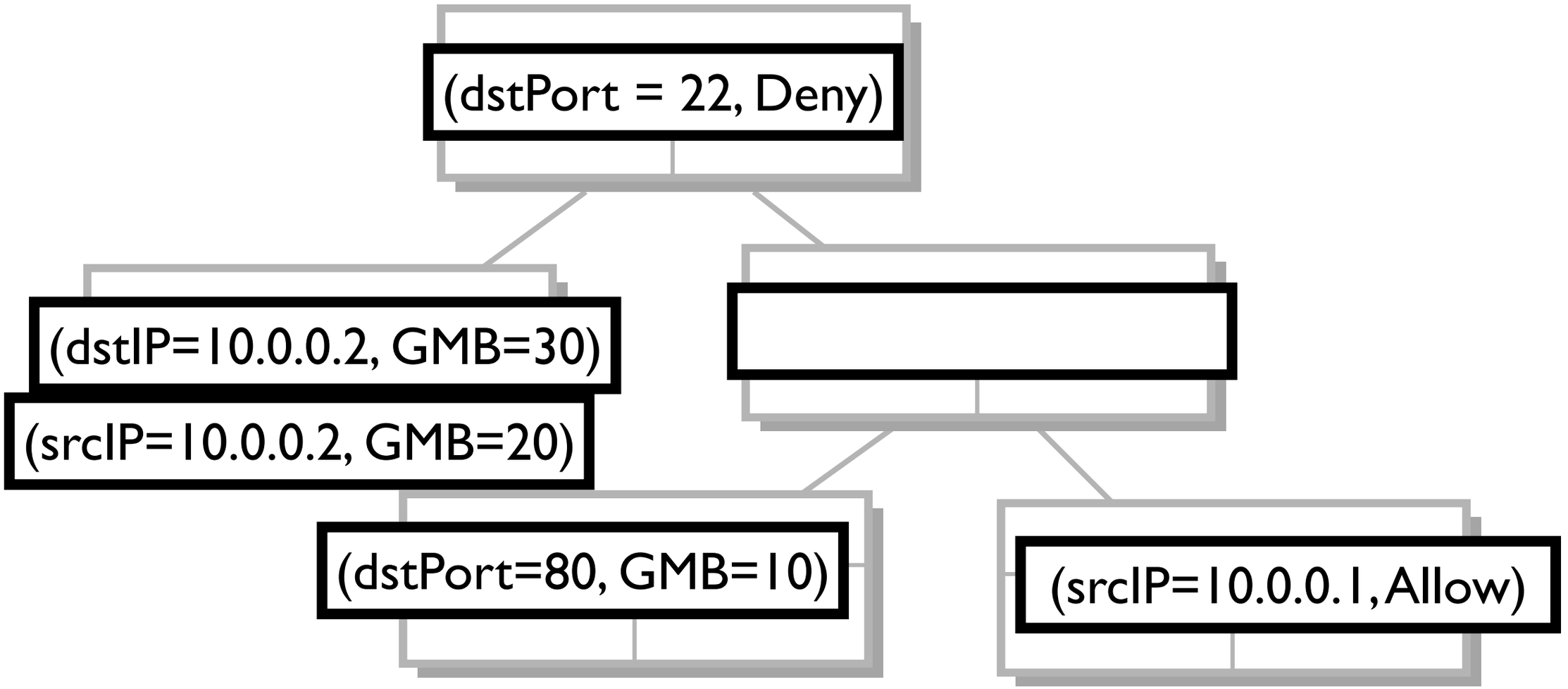
PANE's Conflict Resolution Operators

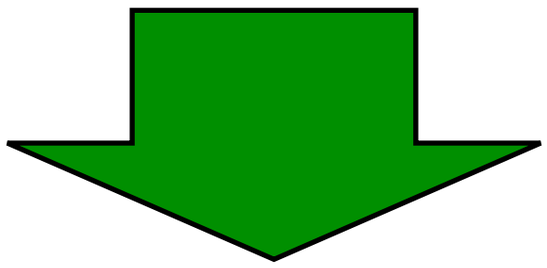
Implementation



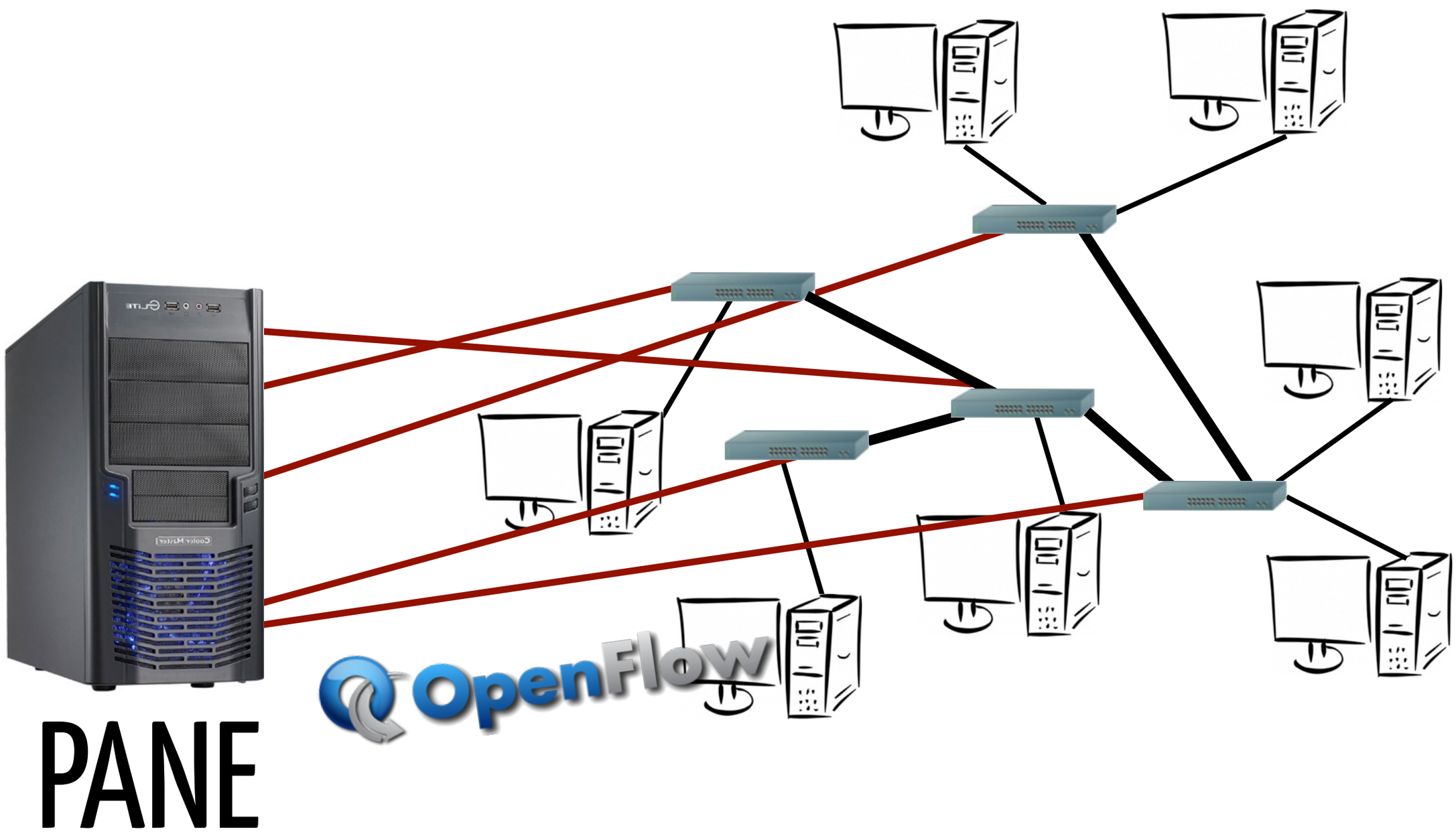
PANE


(dstPort = 22, Deny)
(dstIP=10.0.0.2, GMB=30)
(srcIP=10.0.0.2, GMB=20)
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)

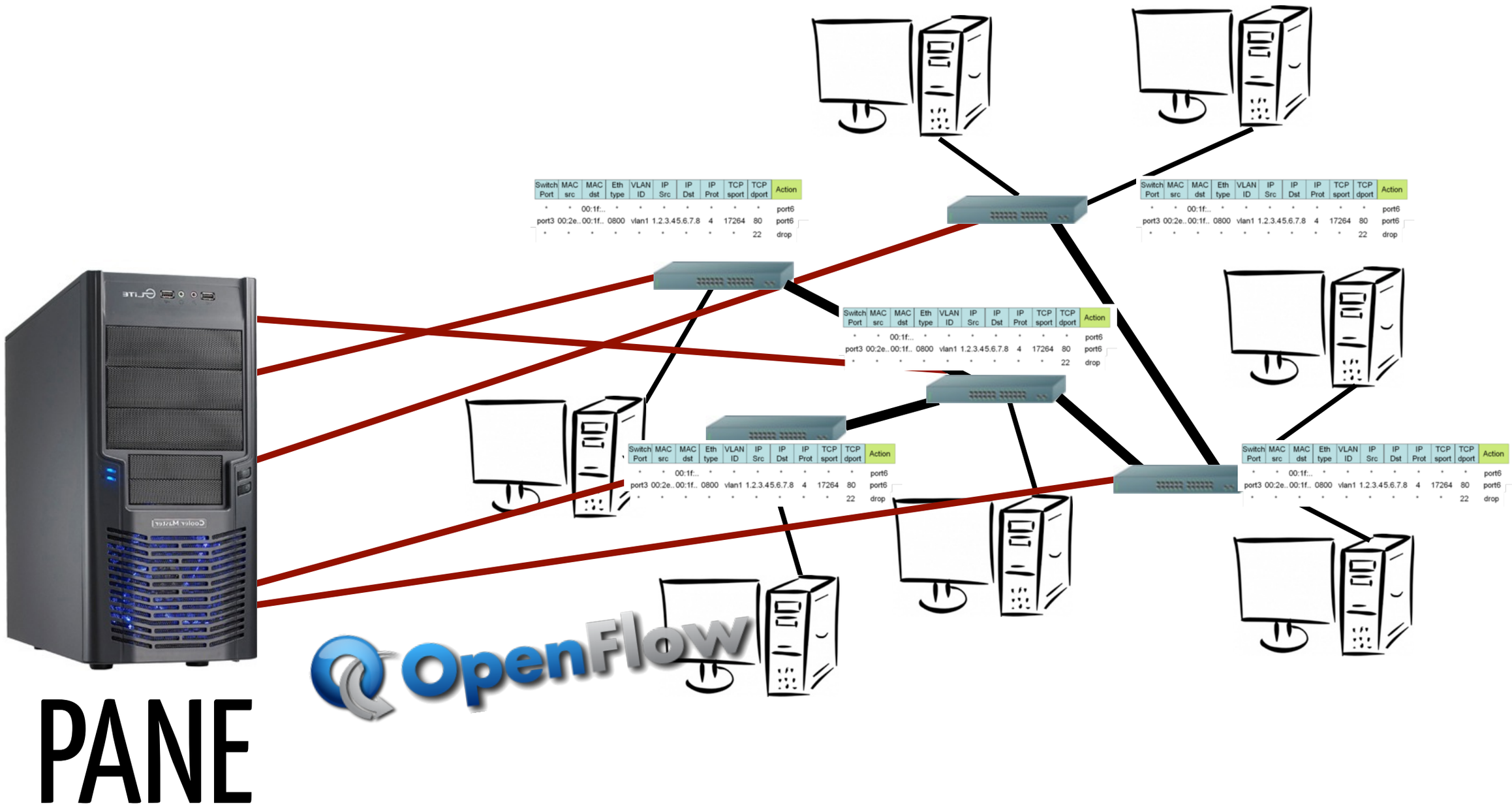




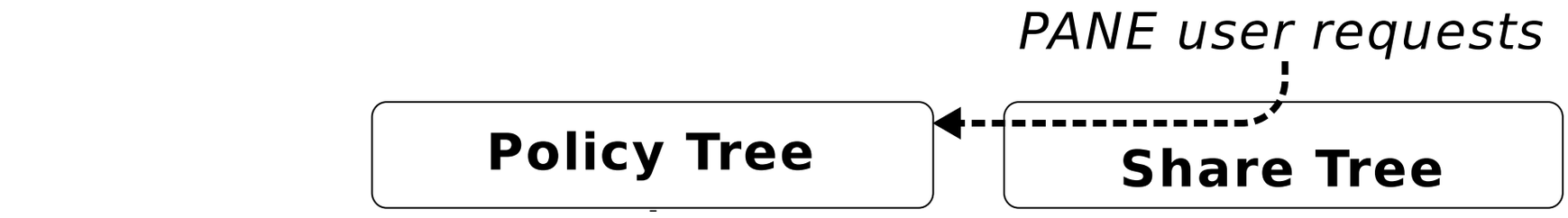
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6
*	*	*	*	*	*	*	*	*	22	drop

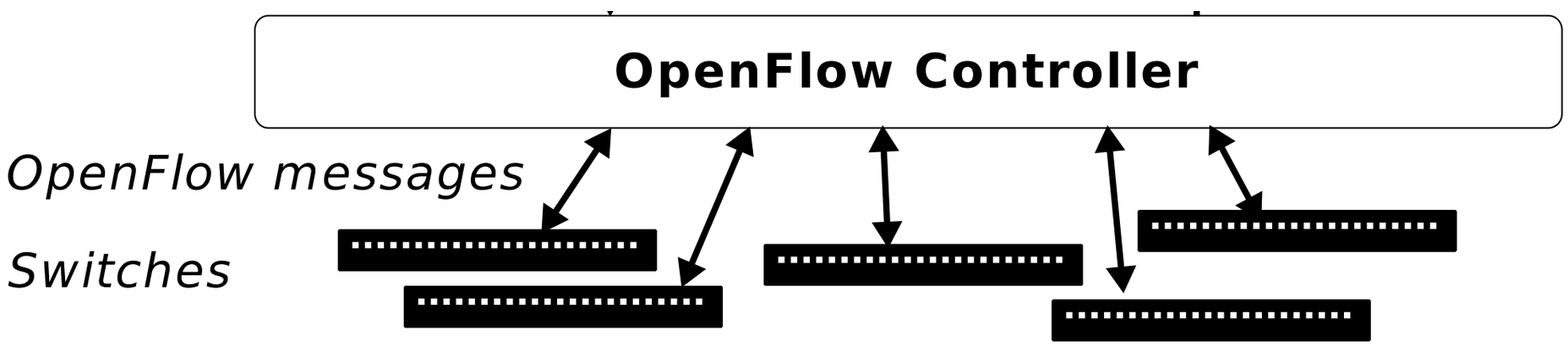


PANE

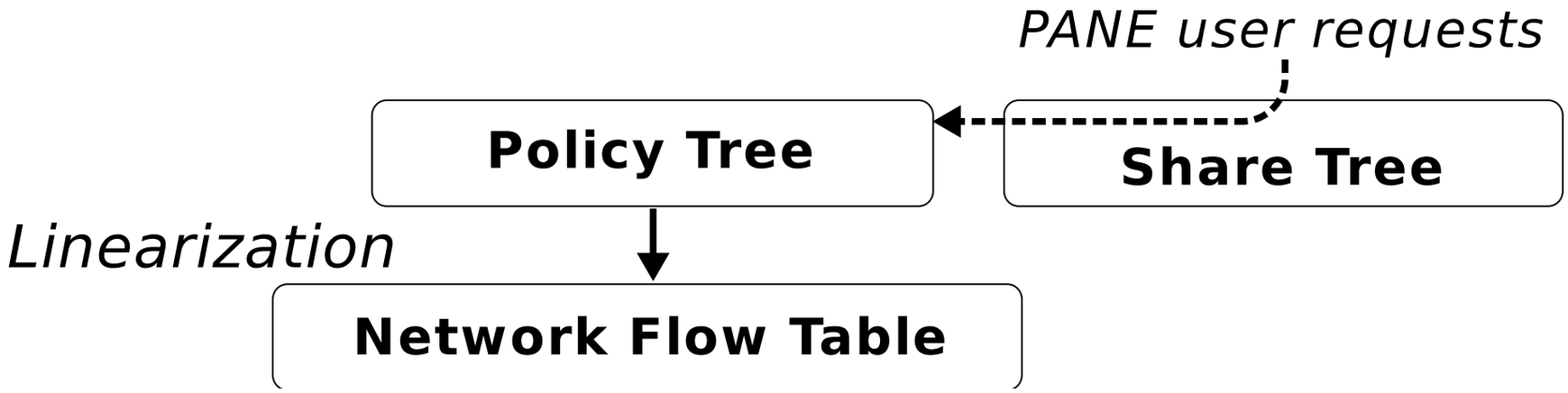

Switch Port | MAC src | MAC dst | Eth type | VLAN ID | IP Src | IP Dst | IP Prot | TCP sport | TCP dport | Action
* * 00:1f:.. * * * * * * * port6
port3 00:2e.. 00:1f.. 0800 vlan1 1.2.3.4 5.6.7.8 4 17264 80 port6
* * * * * * * * * 22 drop
OpenFlow
PANE

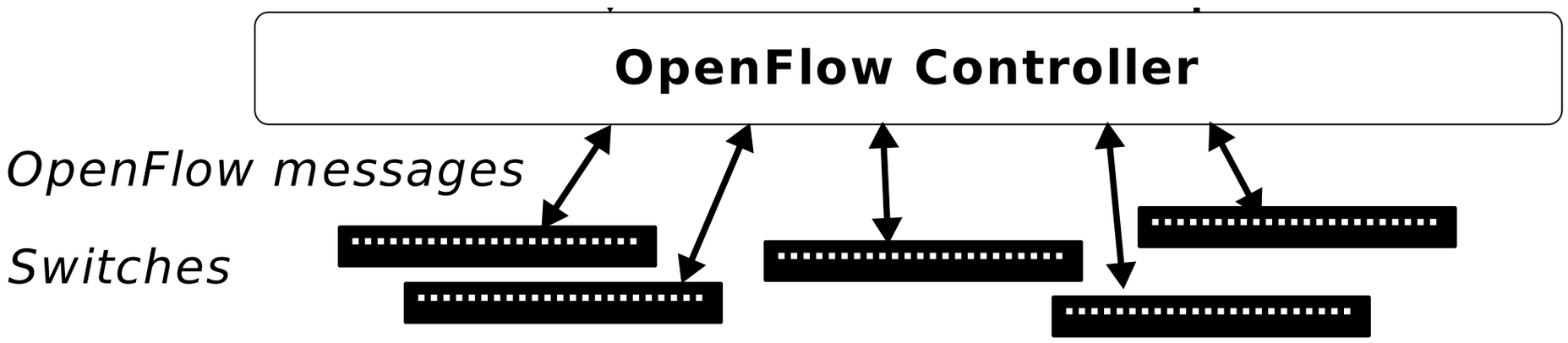
PANE


PANE user requests
Policy Tree
Share Tree

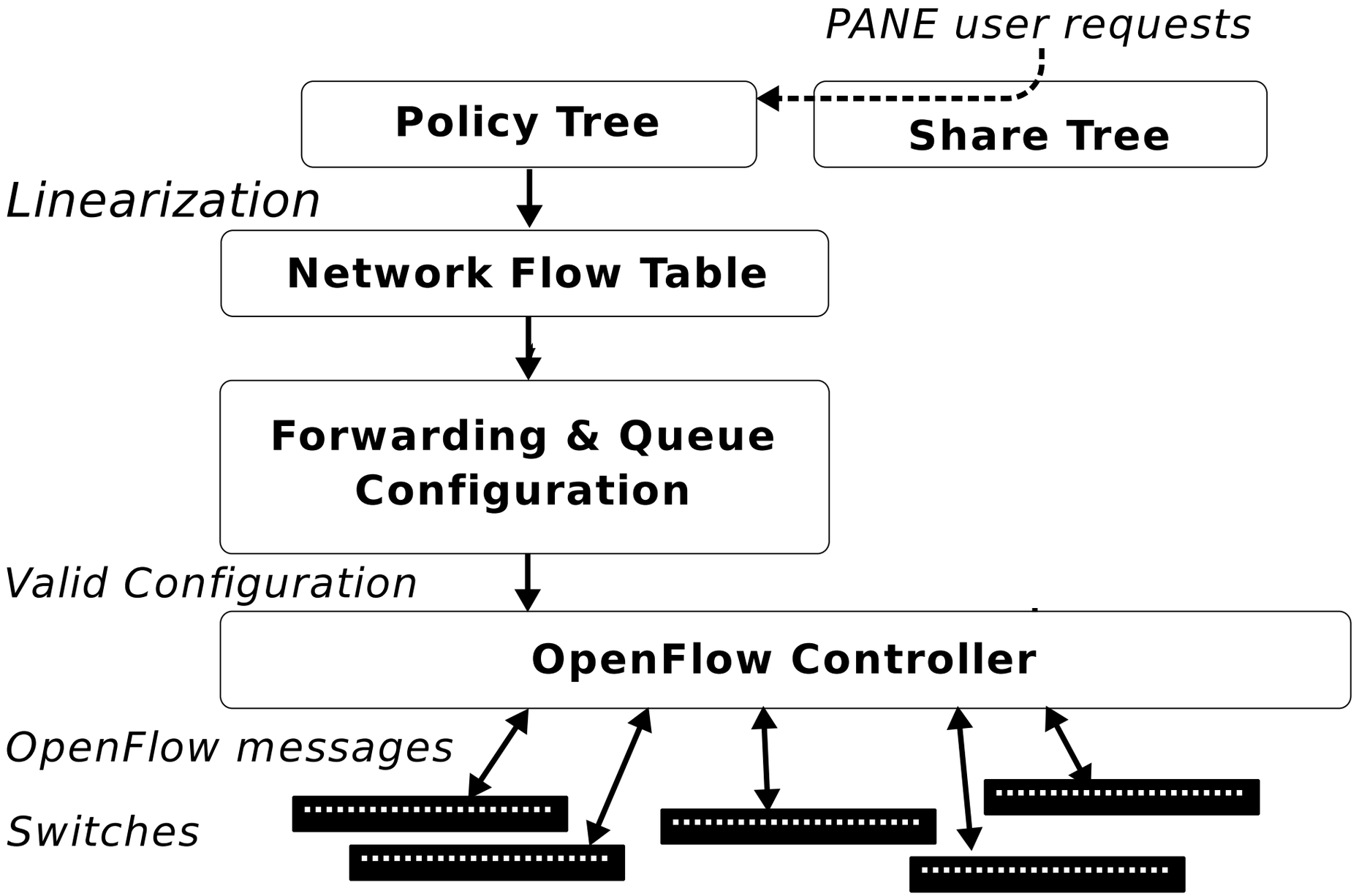

OpenFlow Controller
OpenFlow messages
Switches

PANE

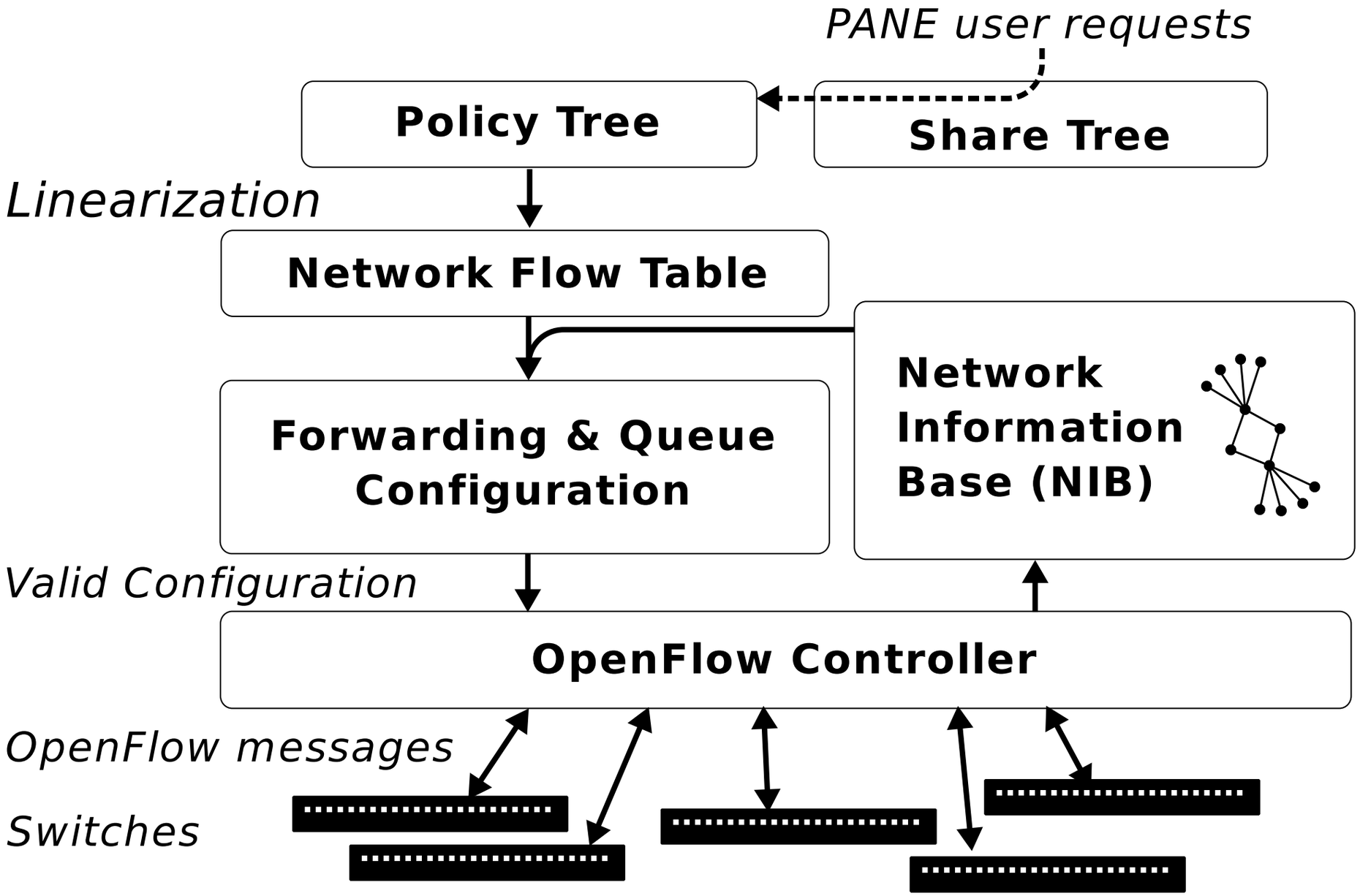

PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table

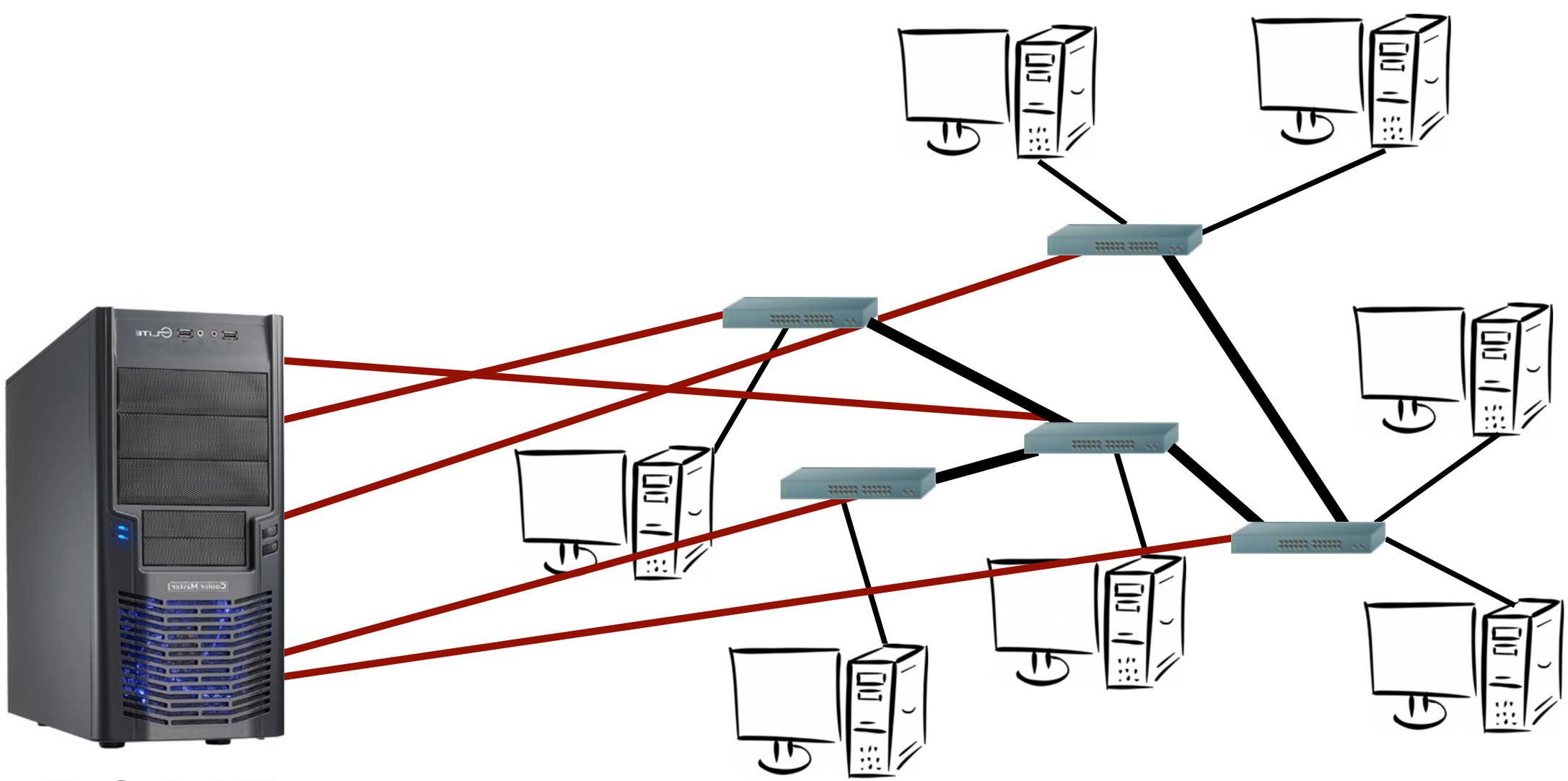

OpenFlow Controller
OpenFlow messages
Switches

PANE

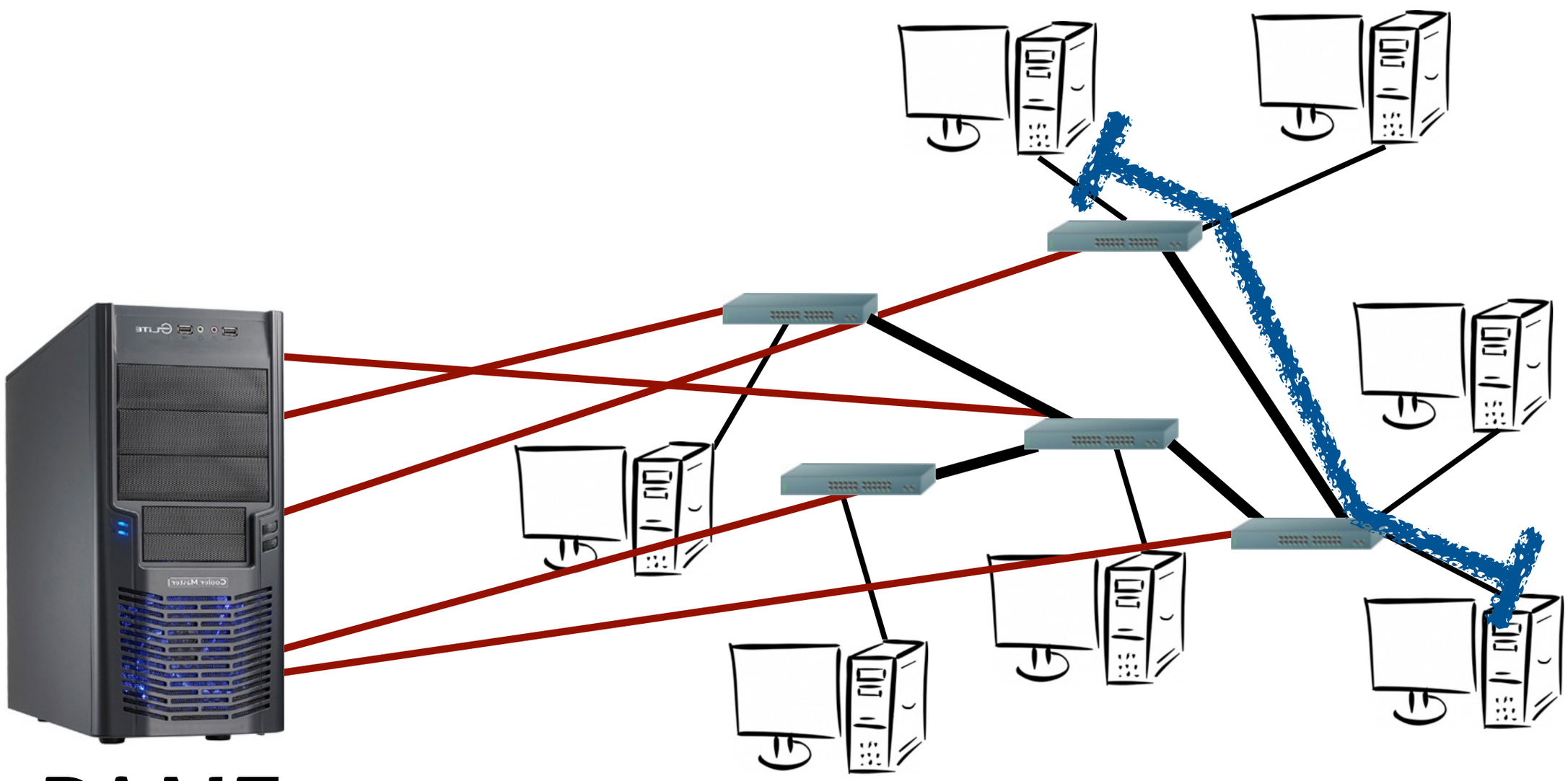

PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches

PANE

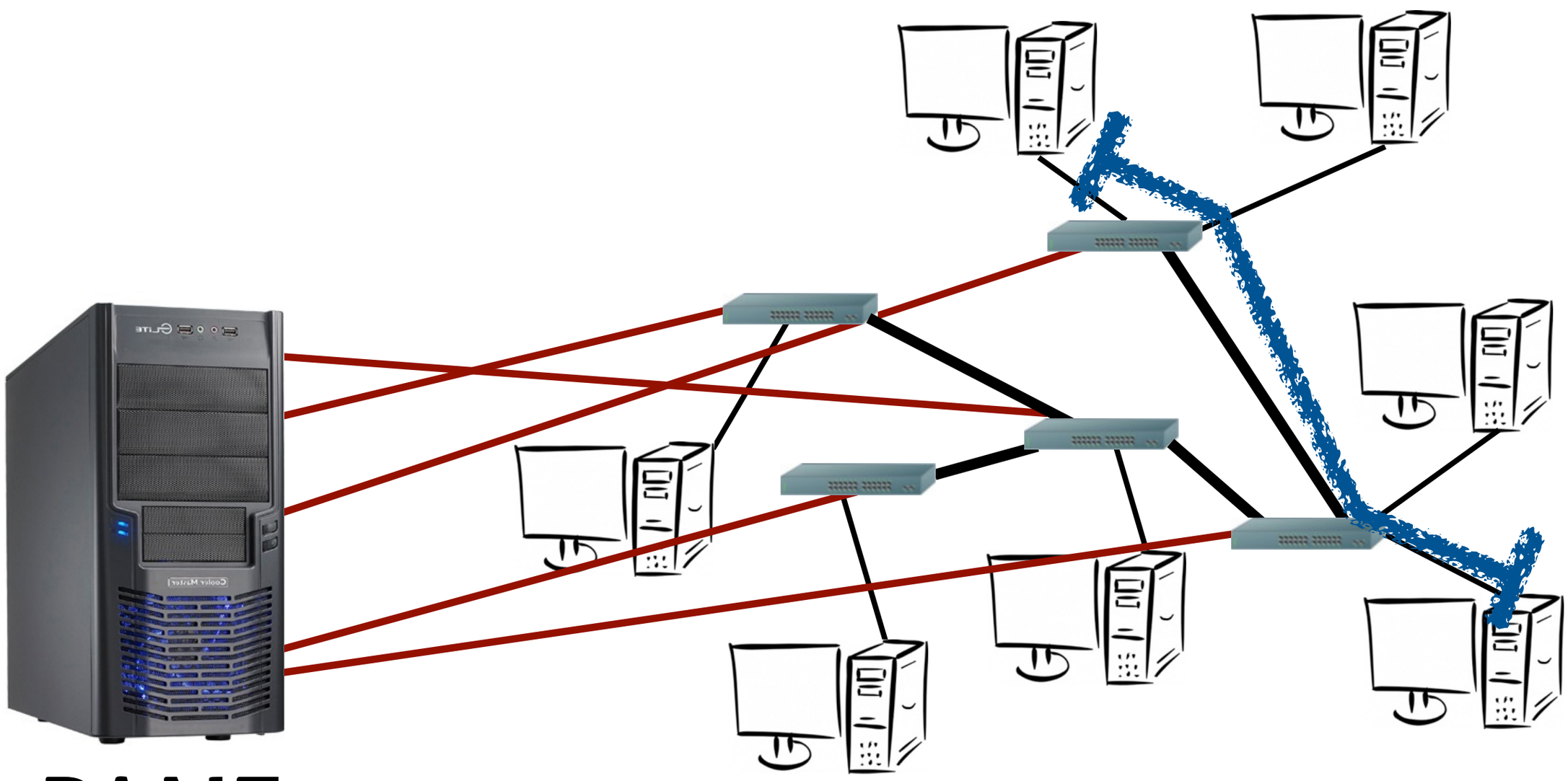

PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Network Information Base (NIB)
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches



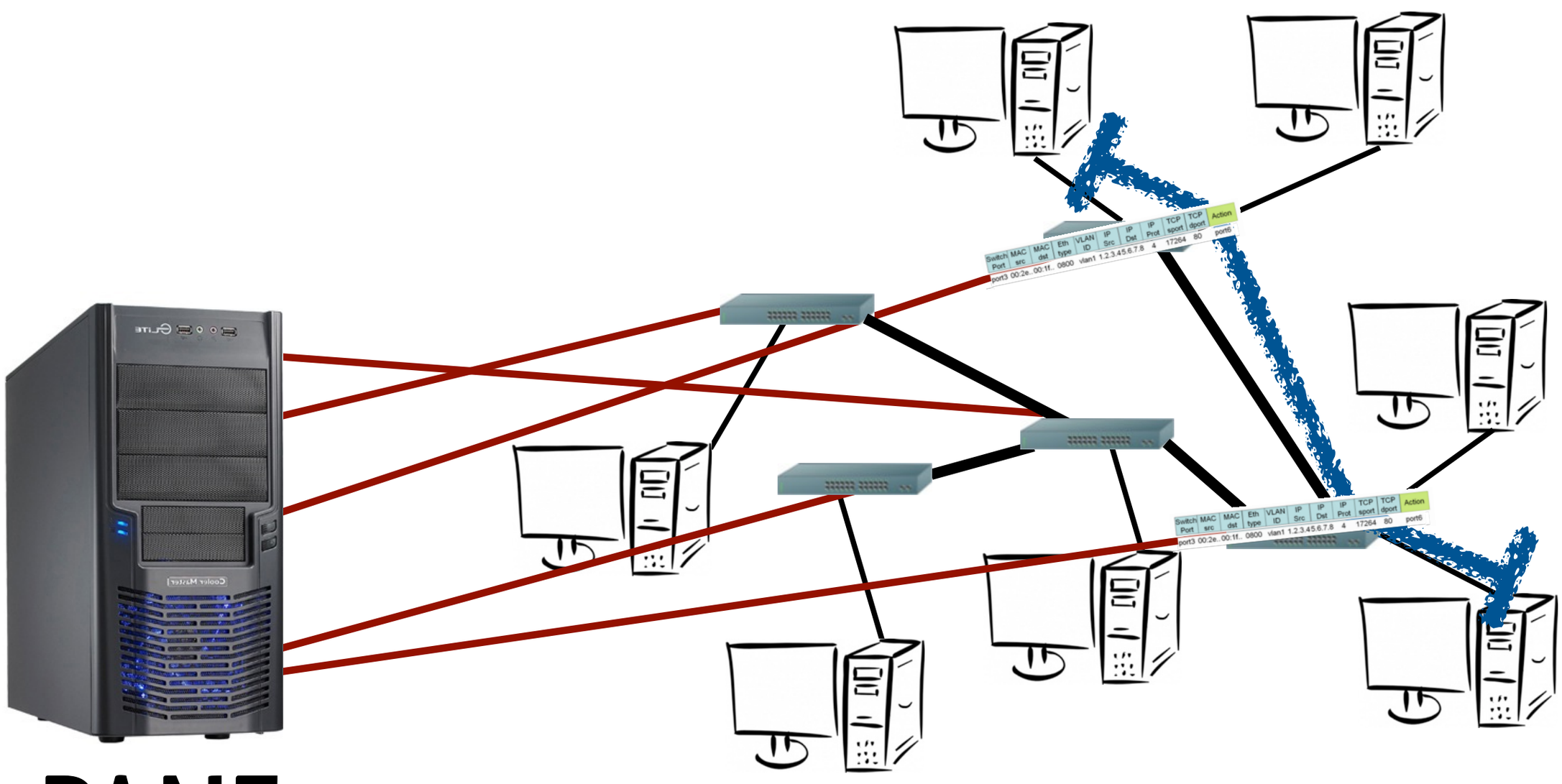
PANE



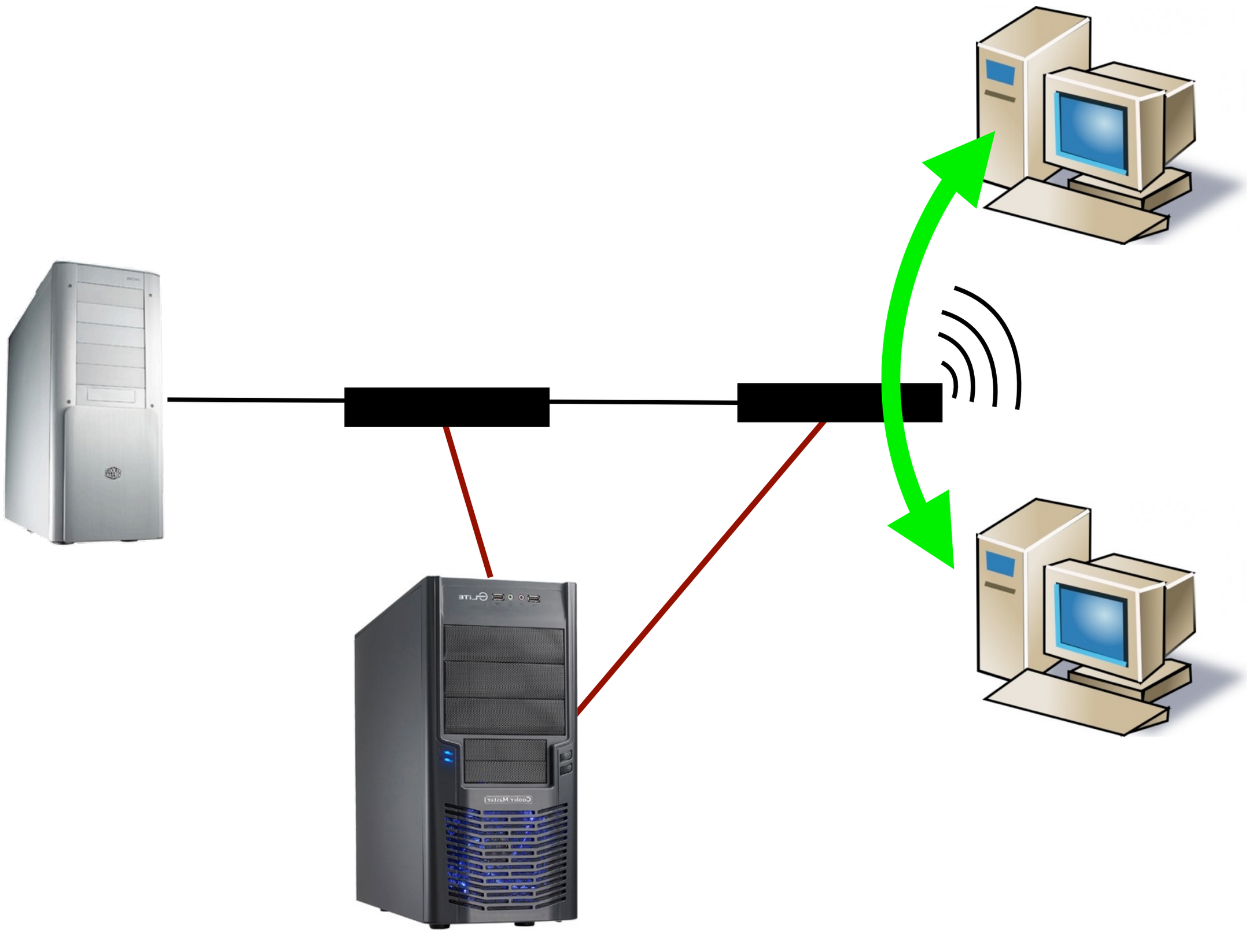
PANE



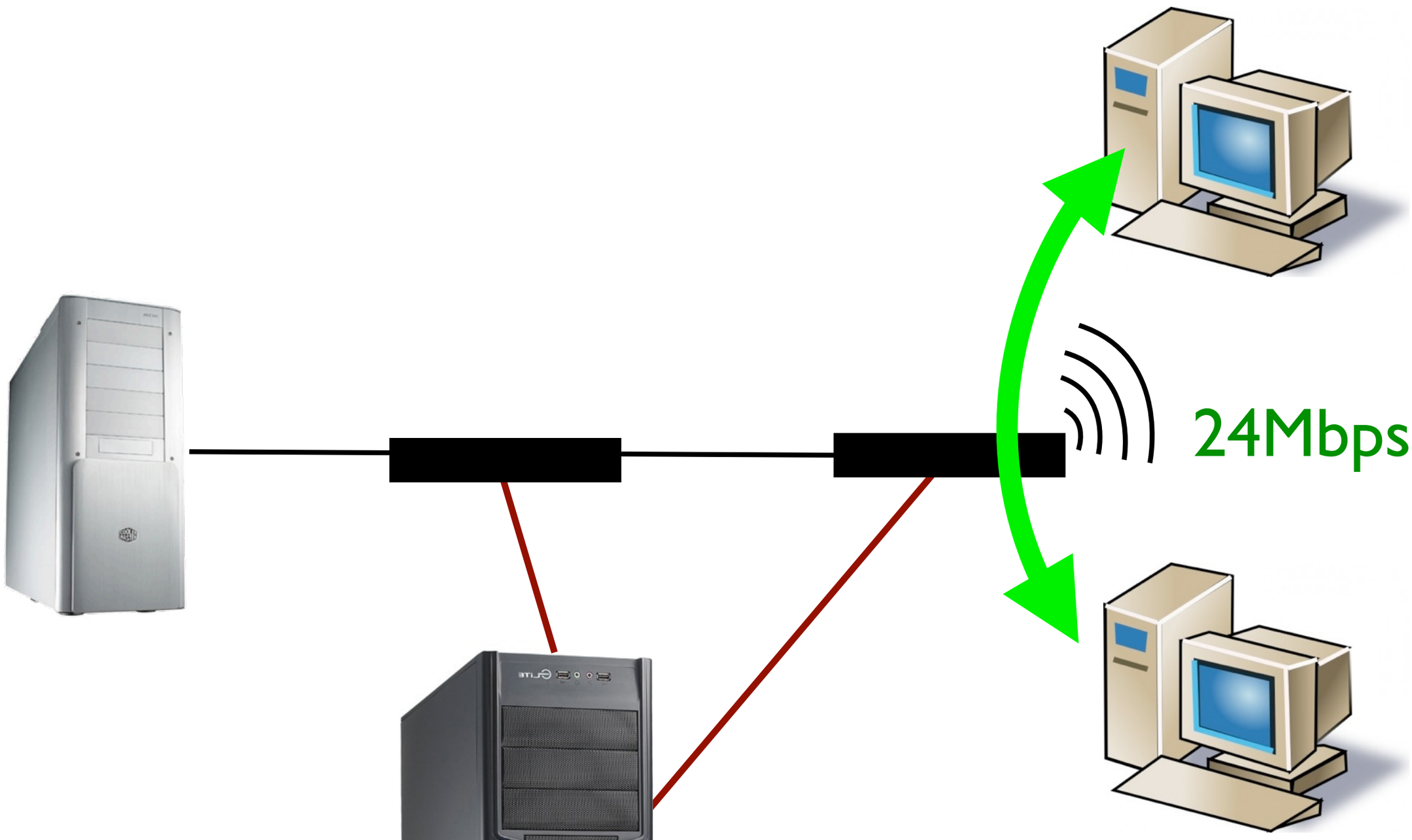
PANE


Switch Port
MAC src
MAC dst
Eth type
VLAN ID
IP Src
IP Dst
IP Prot
TCP sport
TCP dport
Action
port3 00:2e..00:1f.. 0800 vlan1 1.2.3.45.6.7.8 4 17264 80 port6
Switch Port
MAC src
MAC dst
Eth type
VLAN ID
IP Src
IP Dst
IP Prot
TCP sport
TCP dport
Action
port3 00:2e..00:1f.. 0800 vlan1 1.2.3.45.6.7.8 4 17264 80 port6

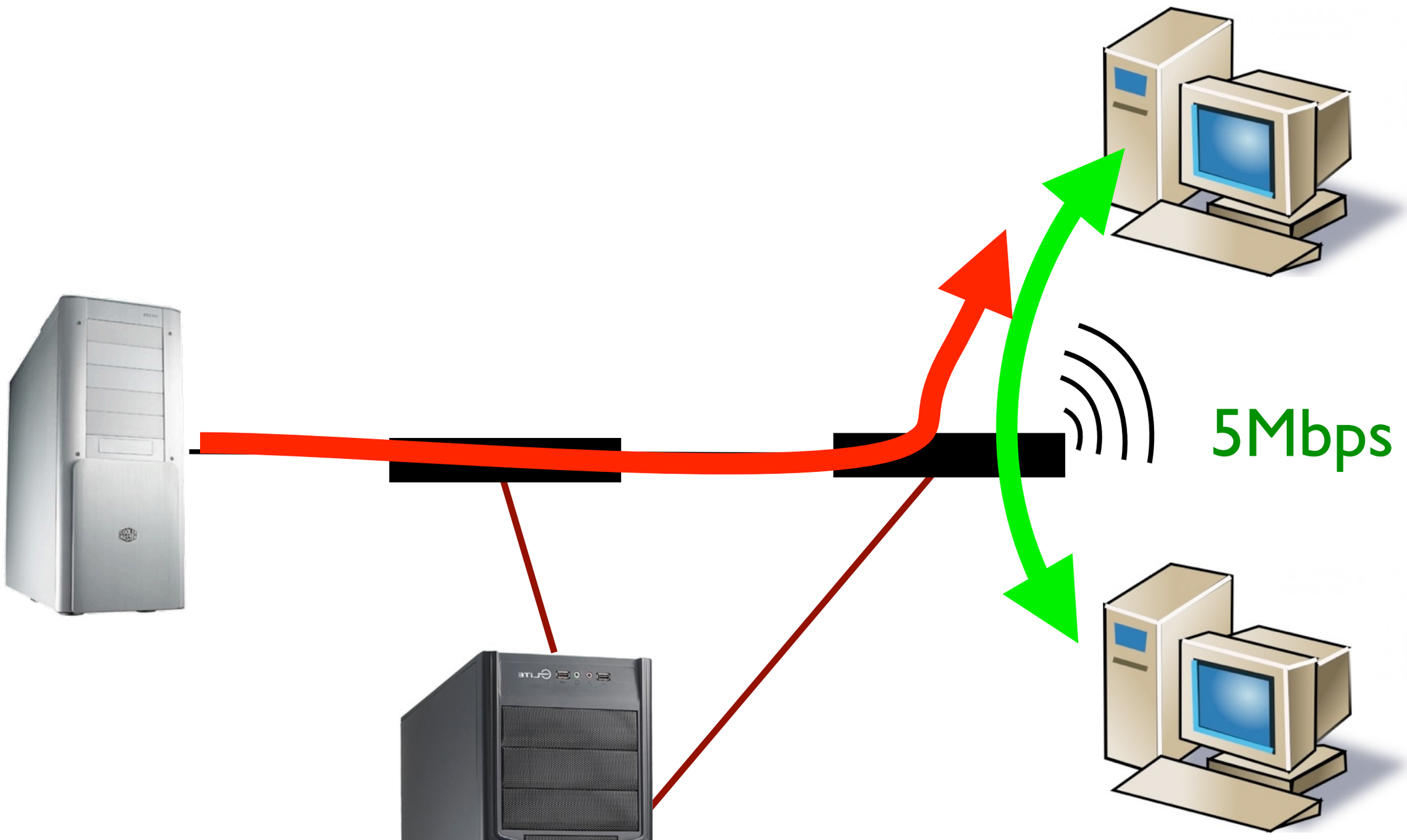
PANE



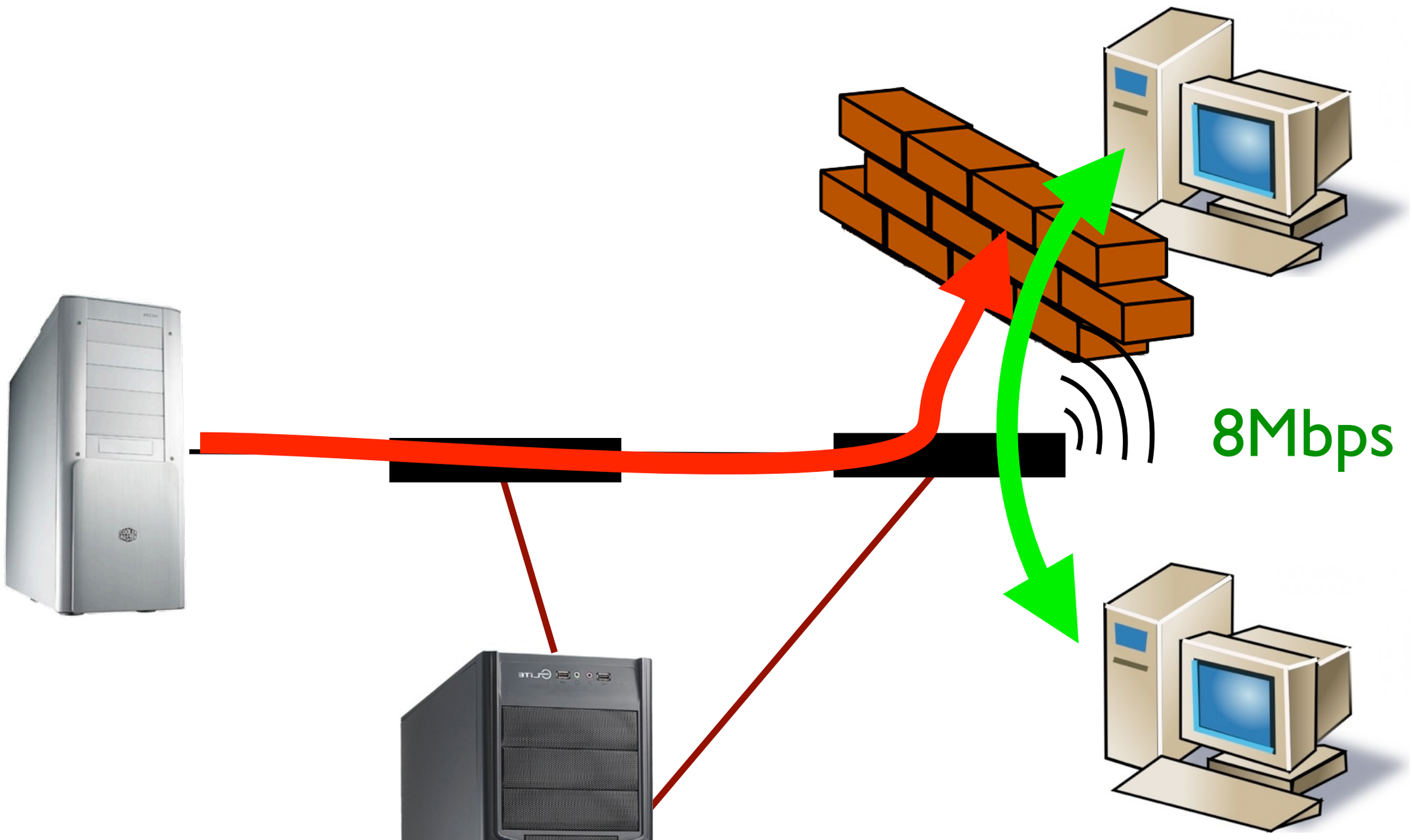
PANE



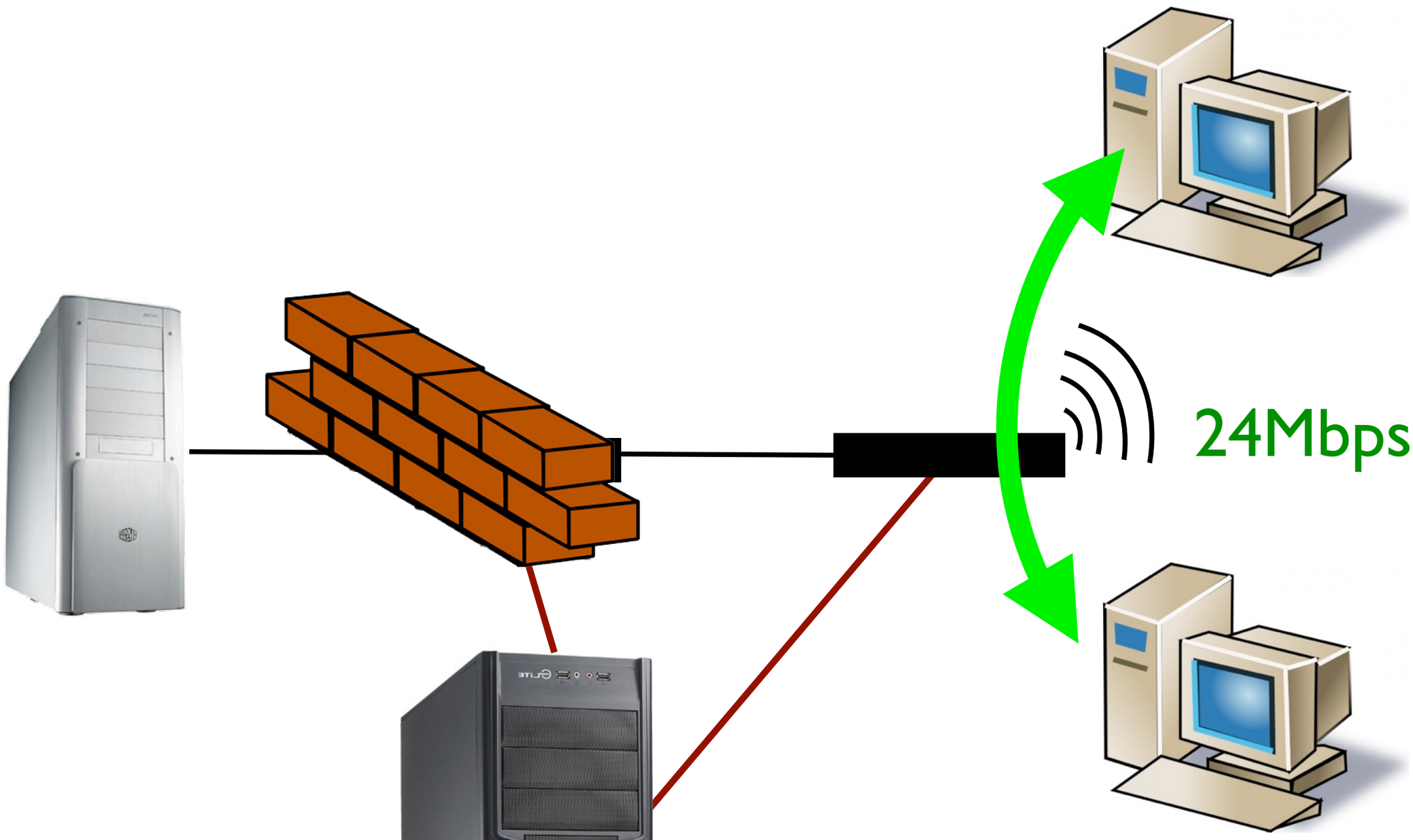
PANE



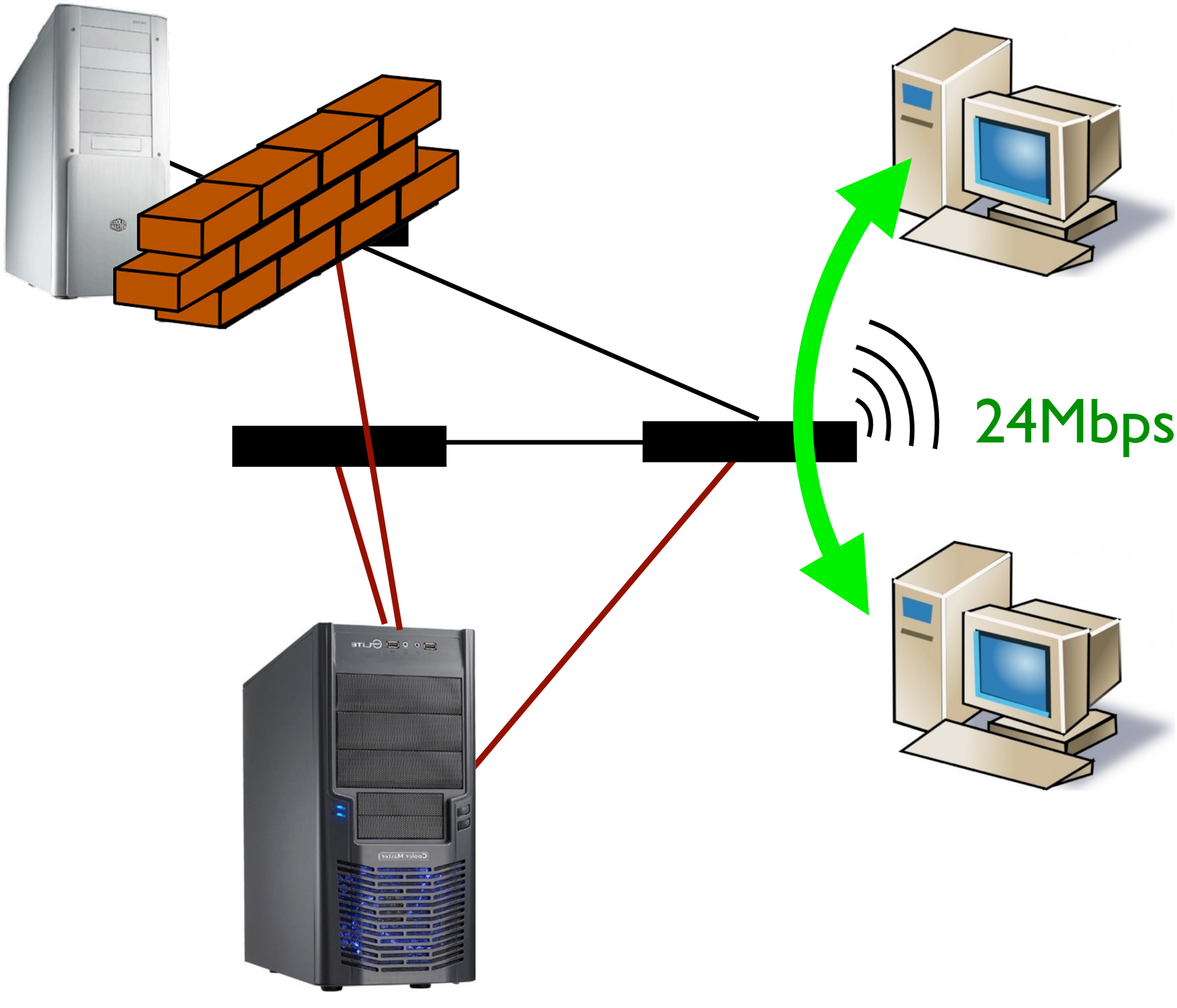
PANE



PANE

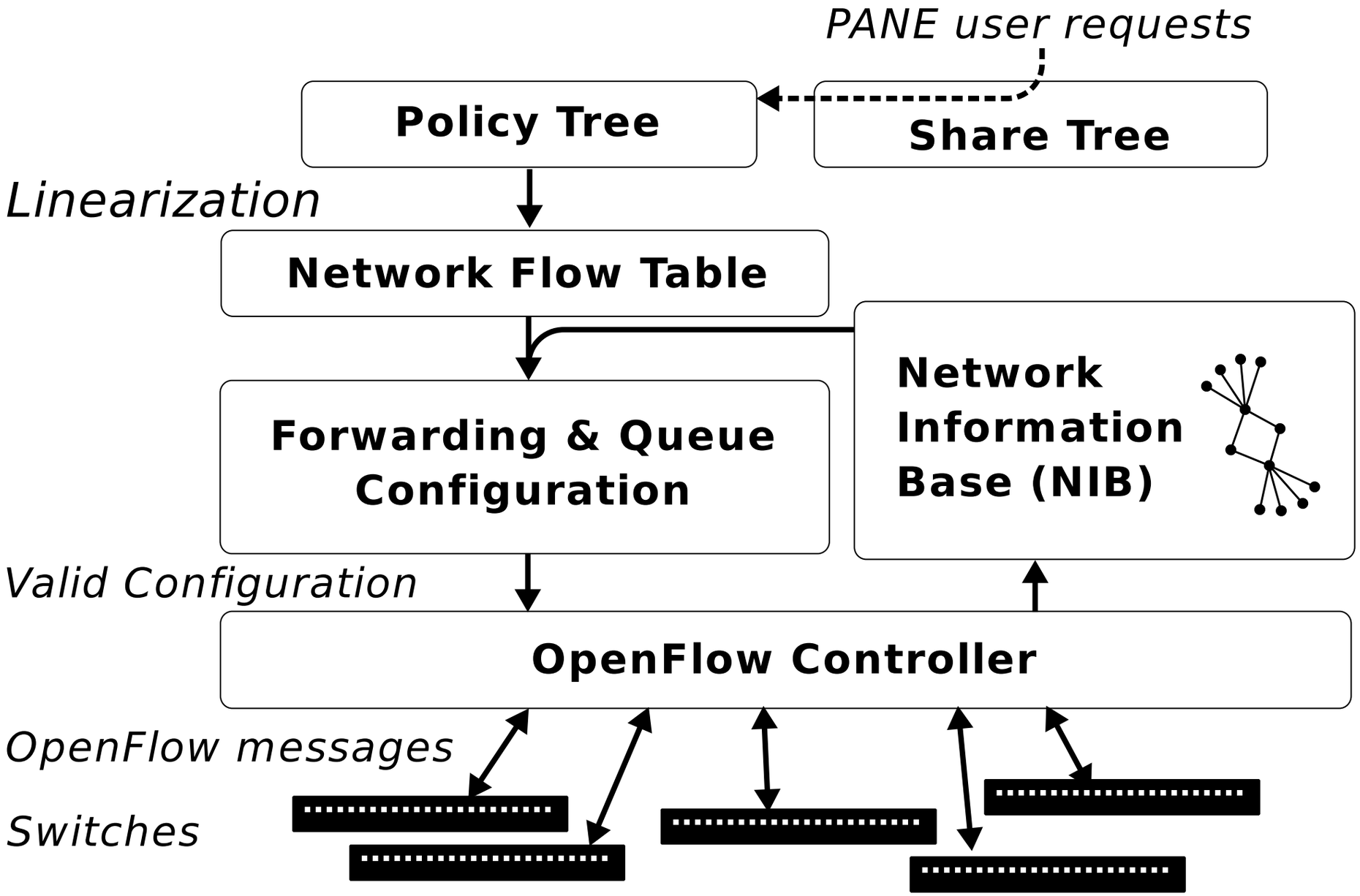


PANE

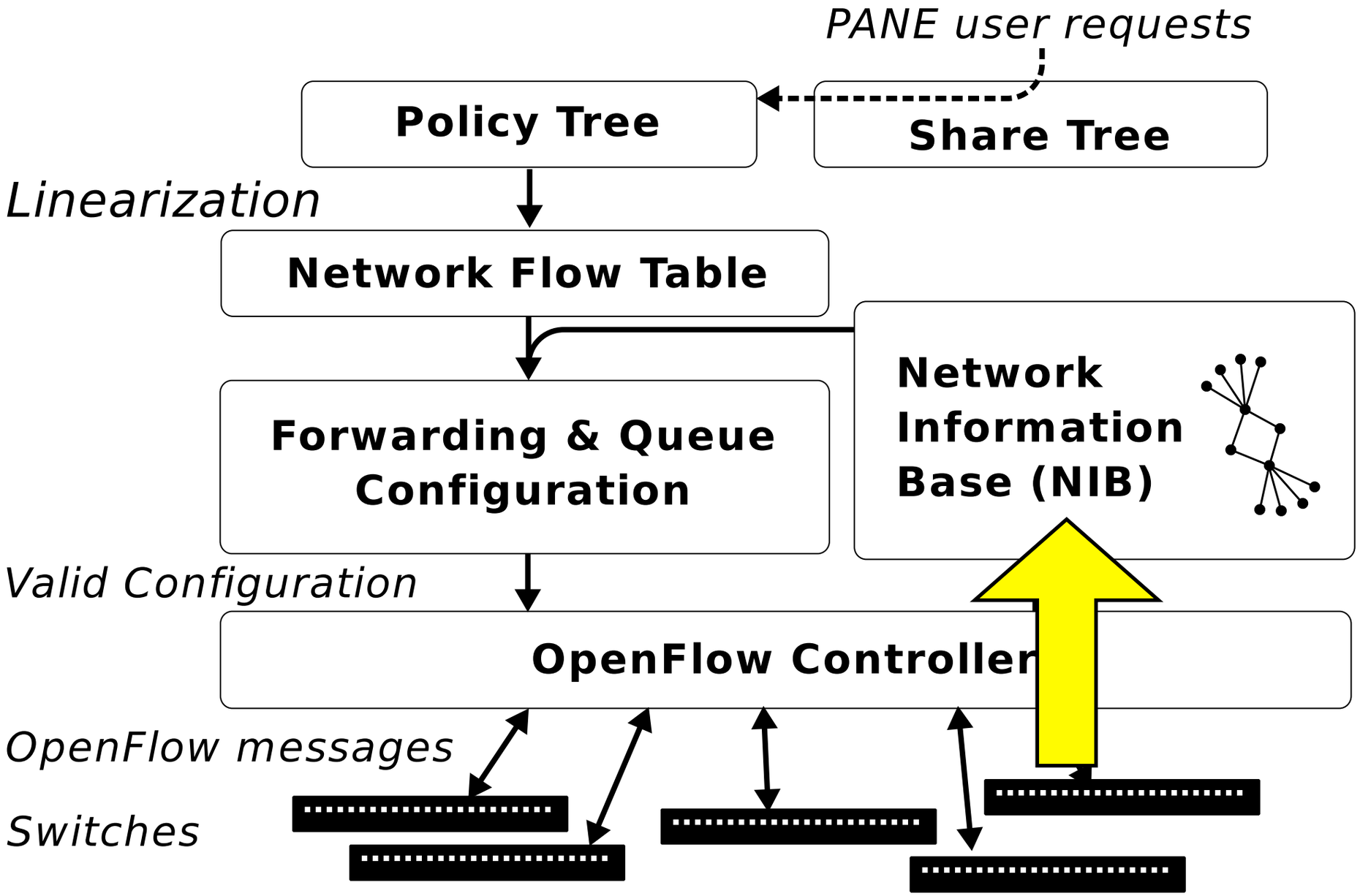


PANE

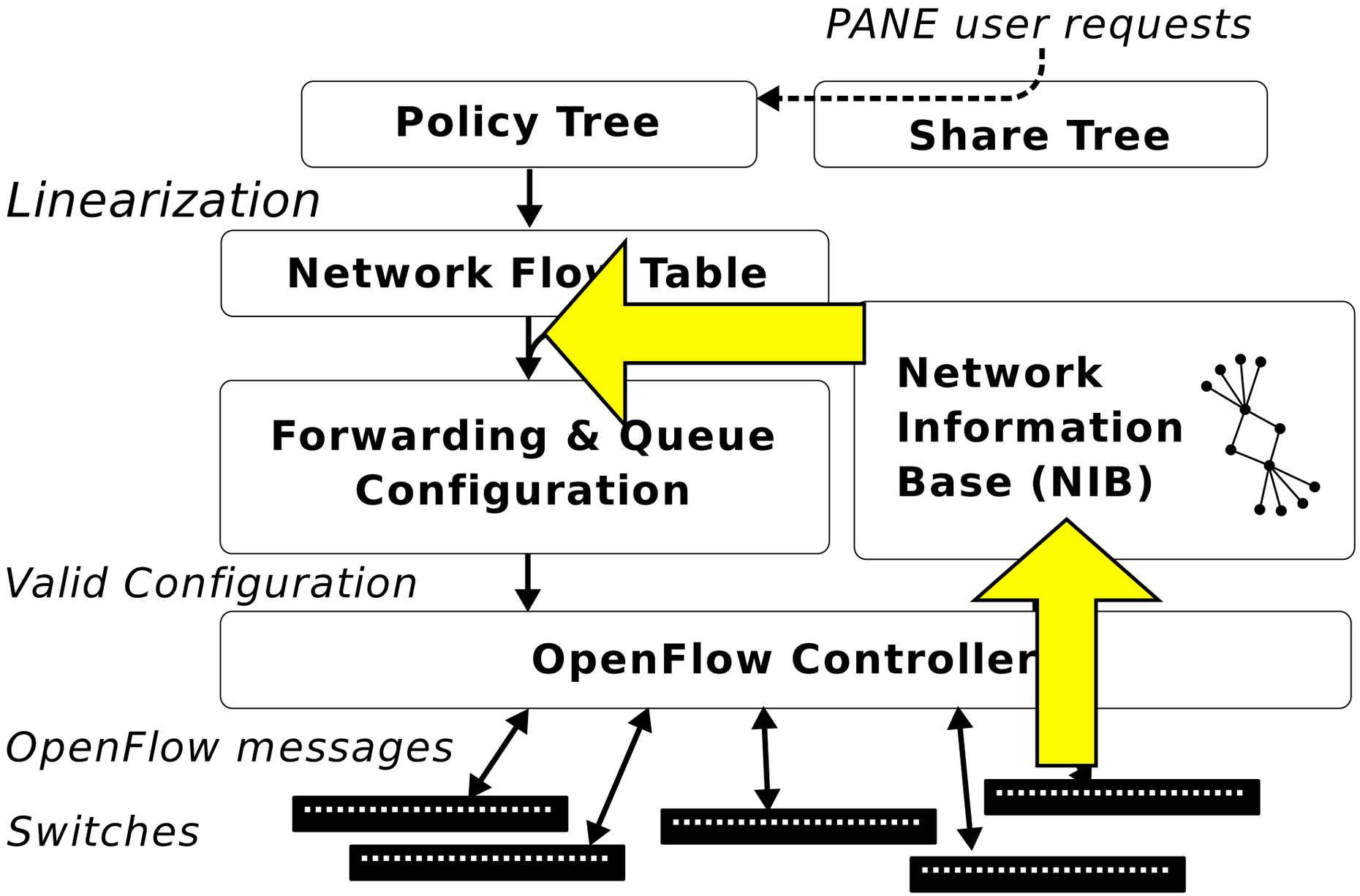
PANE


PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Network Information Base (NIB)
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches

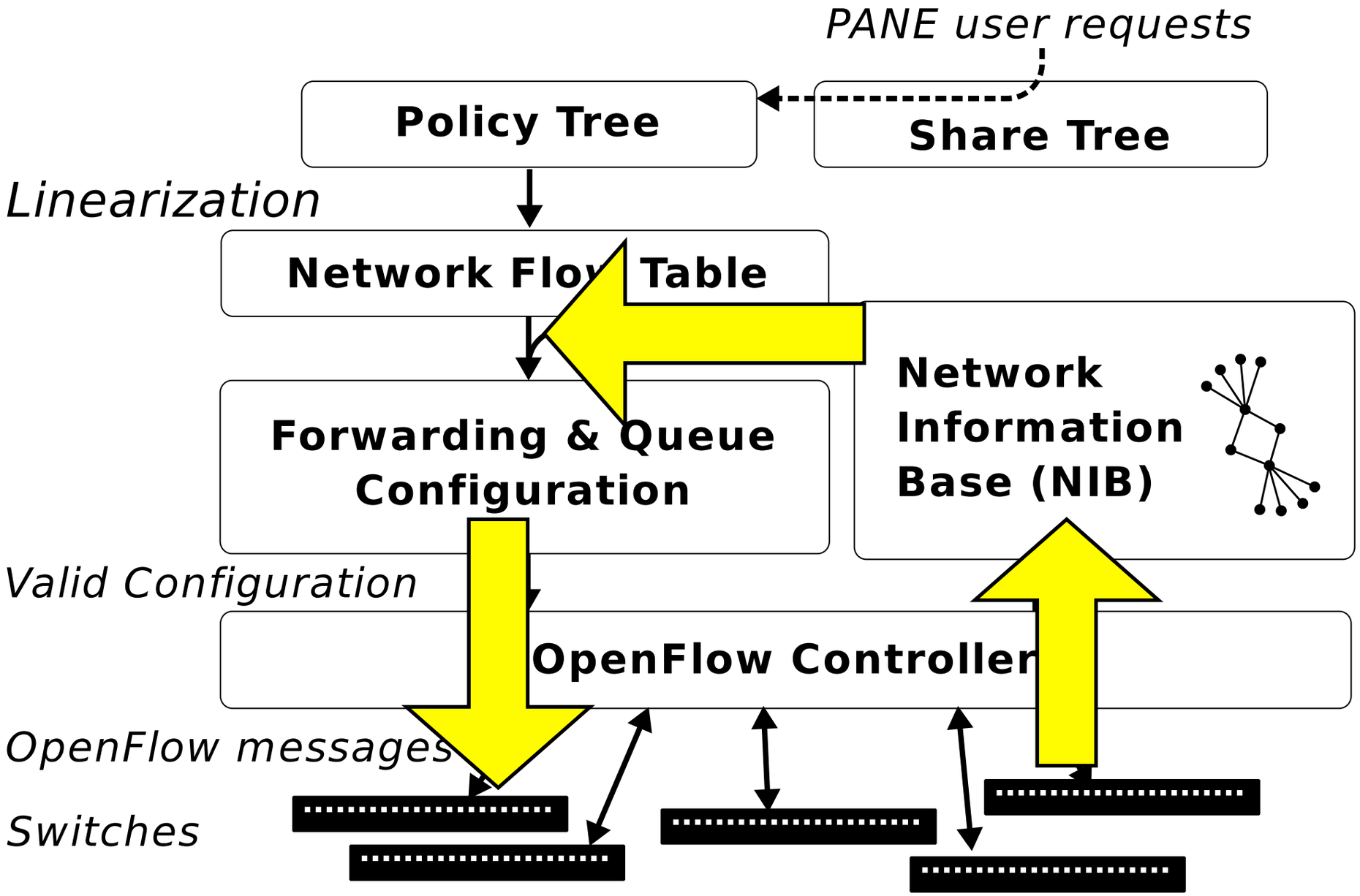
PANE


PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Network Information Base (NIB)
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches

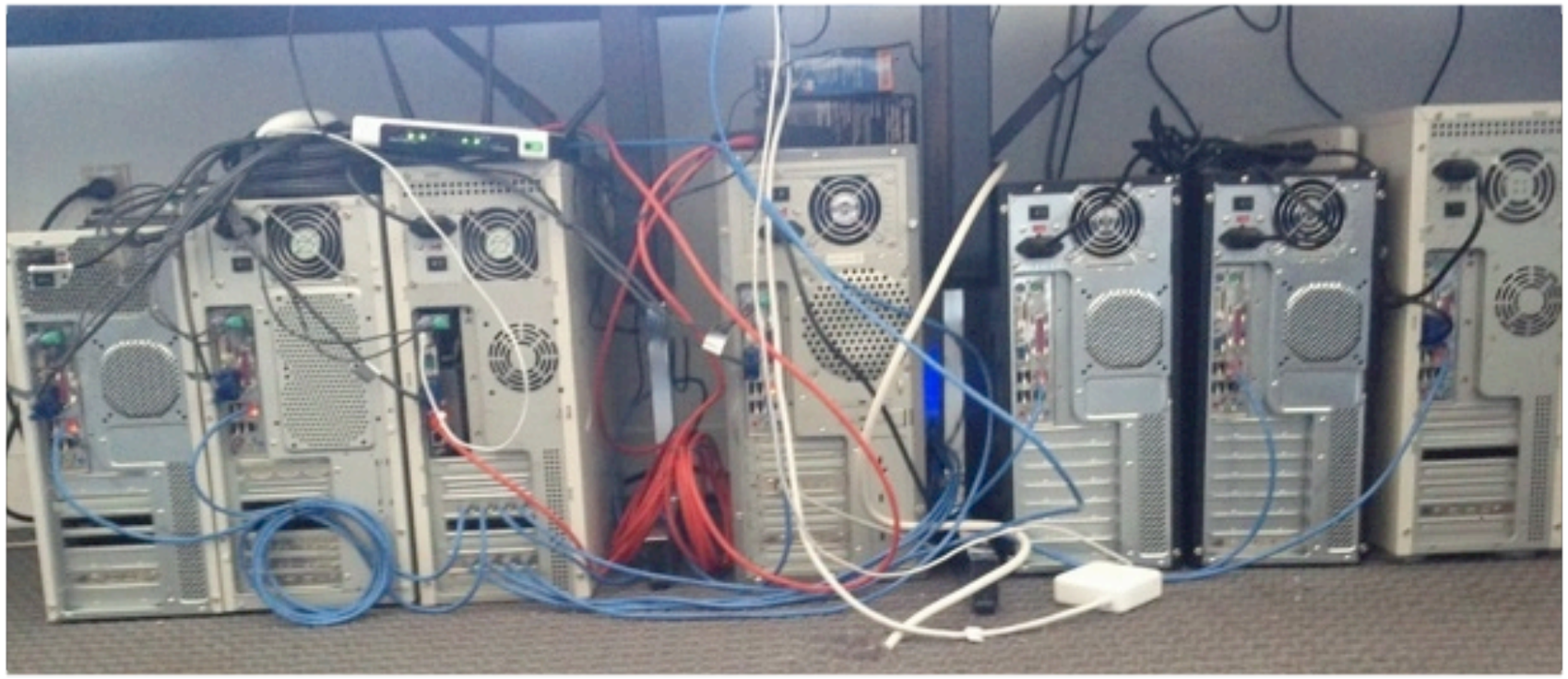
PANE

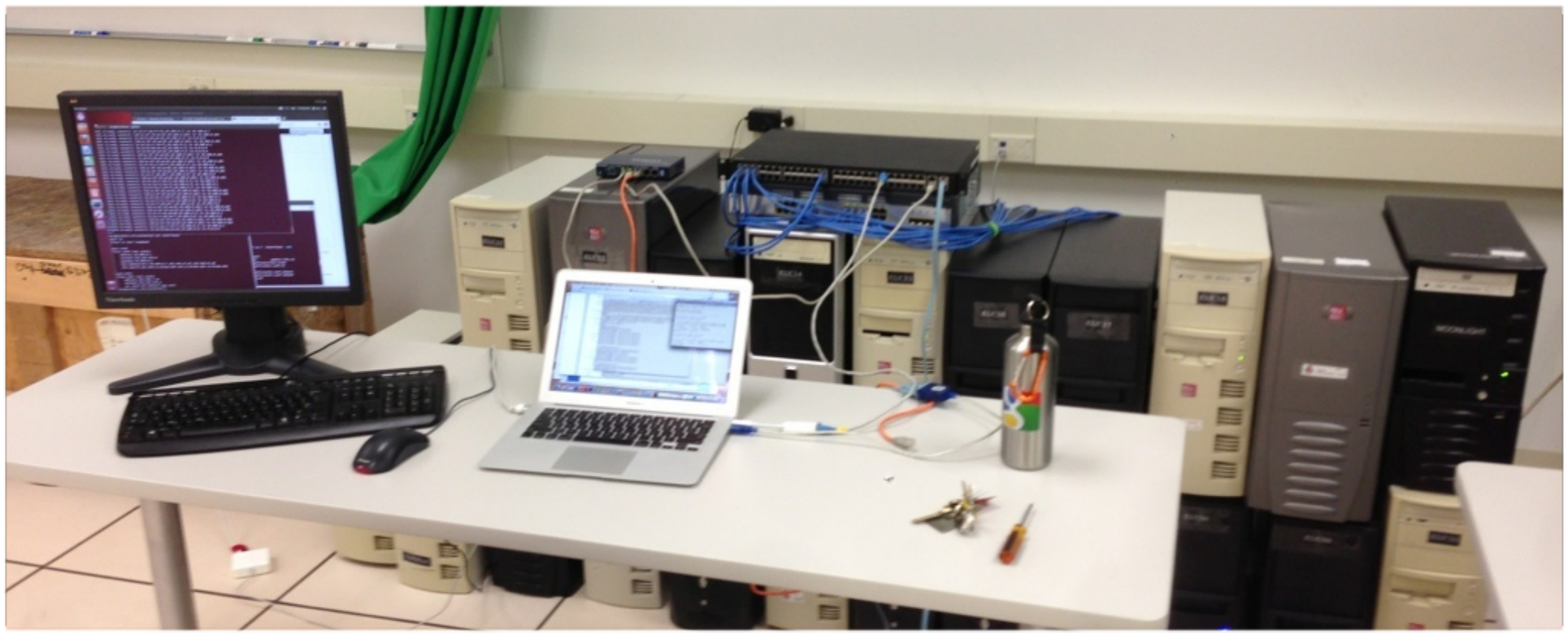

PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Network Information Base (NIB)
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches

PANE


PANE user requests
Policy Tree
Share Tree
Linearization
Network Flow Table
Forwarding & Queue Configuration
Network Information Base (NIB)
Valid Configuration
OpenFlow Controller
OpenFlow messages
Switches

Evaluation





1. SSHGuard	access control
2. Ekiga	bandwidth reservations
3. ZooKeeper	queues for low latency
4. Hadoop	centralized traffic weights

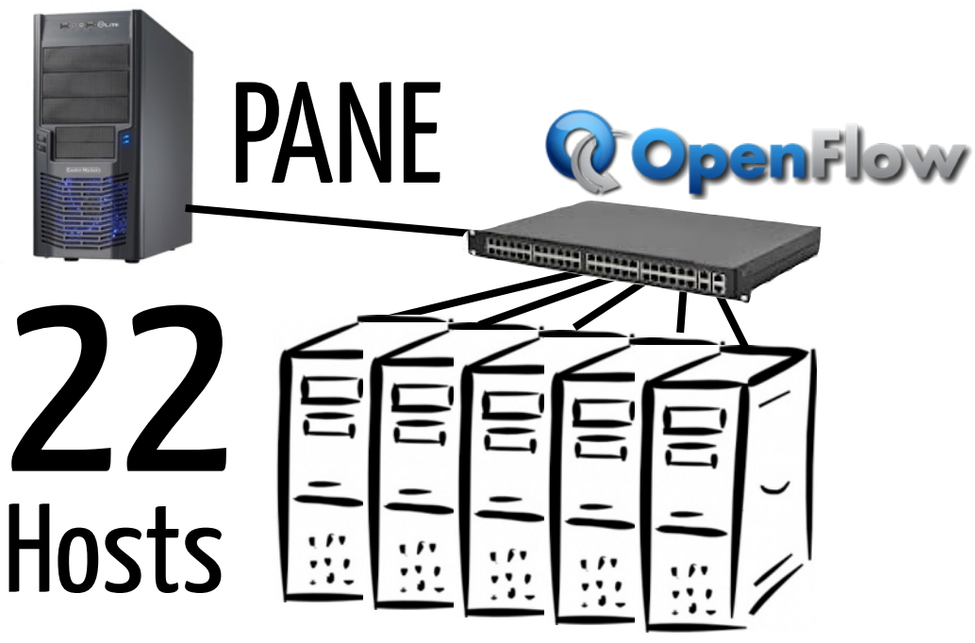
Evaluation

Three equal-sized sort jobs:

- Two Low Priority with 25% weight
- One High Priority with 50% weight

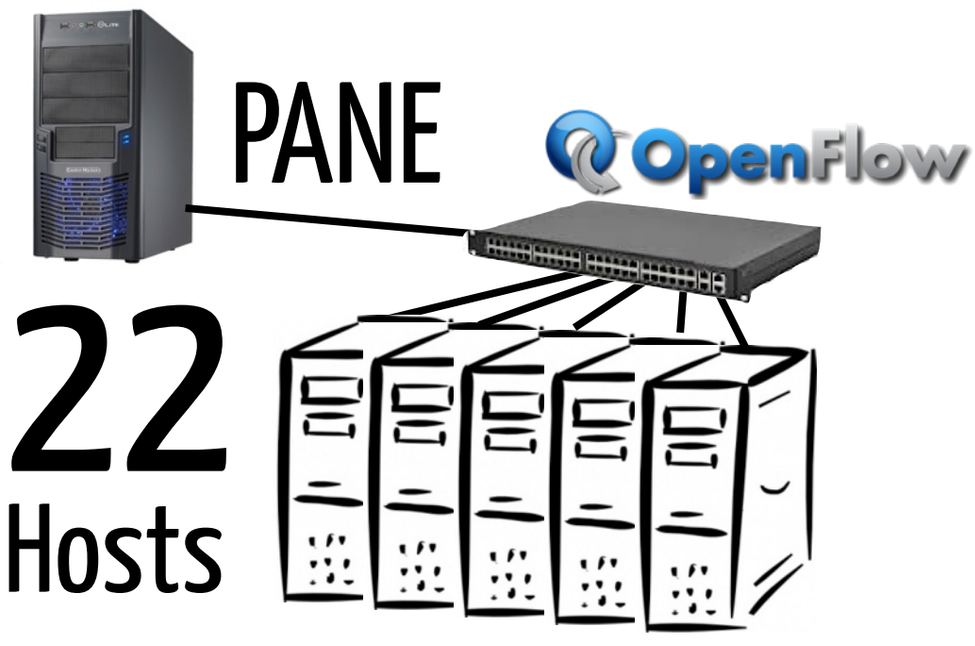
Three equal-sized sort jobs:

- Two Low Priority with 25% weight
- One High Priority with 50% weight

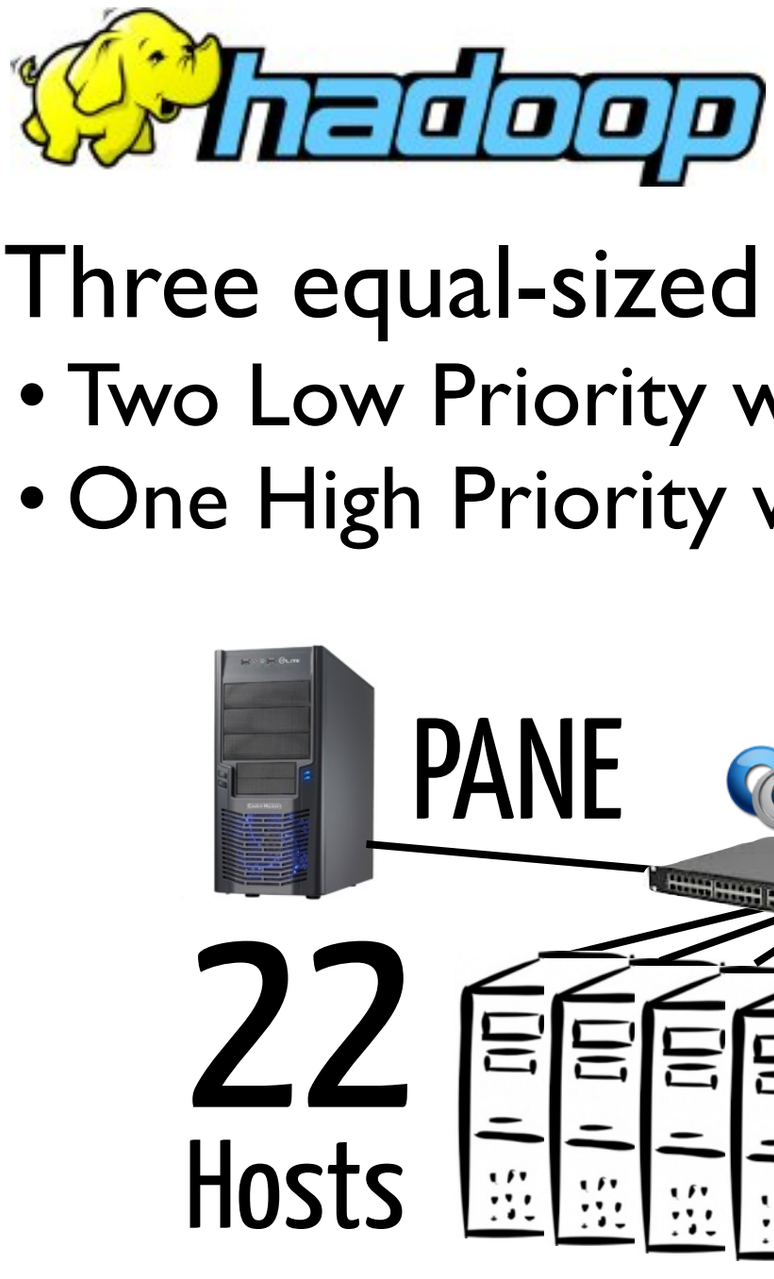


Three equal-sized sort jobs:

- Two Low Priority with 25% weight
- One High Priority with 50% weight

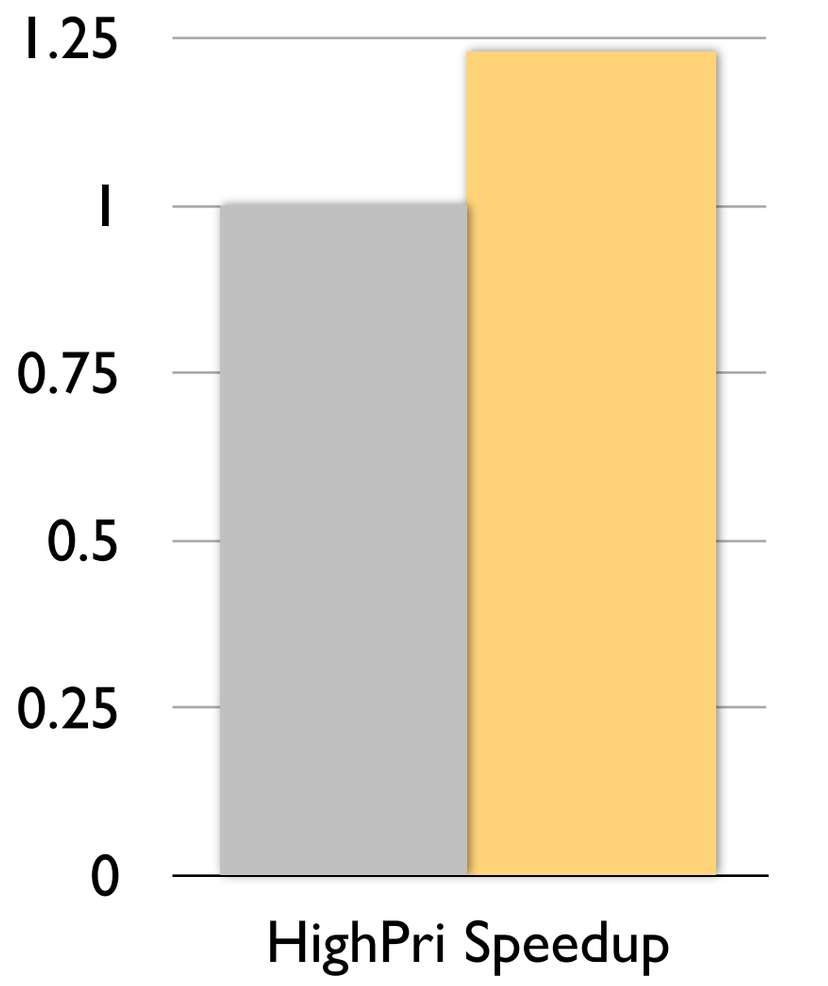


Dynamically apply QoS to High Priority flows using PANE.



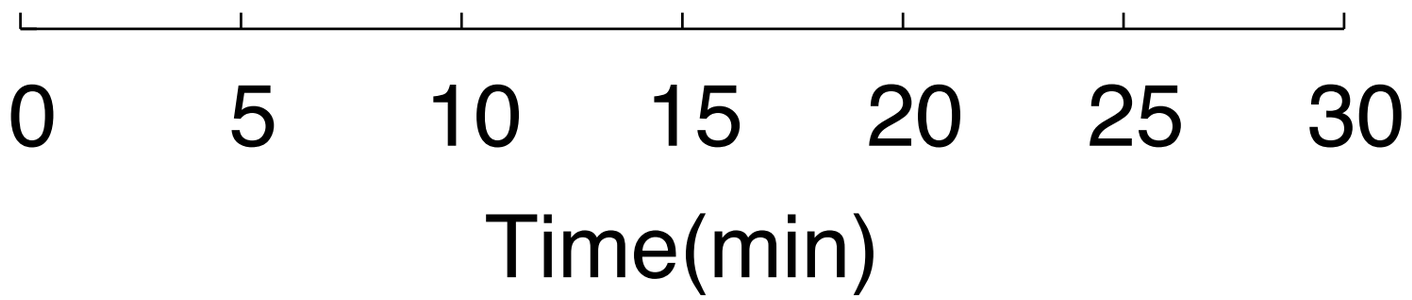
Three equal-sized sort jobs:

- Two Low Priority with 25% weight
- One High Priority with 50% weight

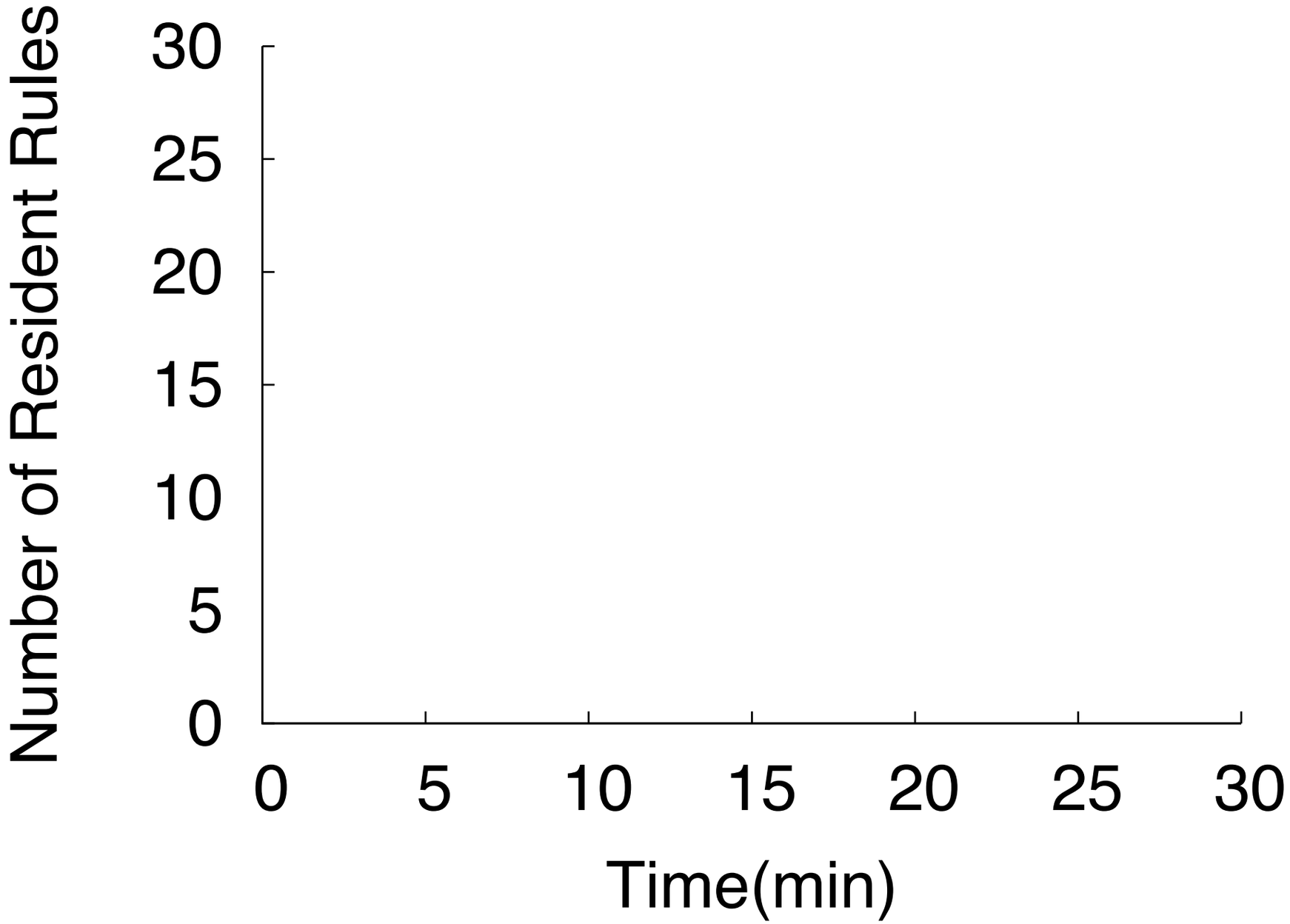


Dynamically apply QoS to High Priority flows using PANE.

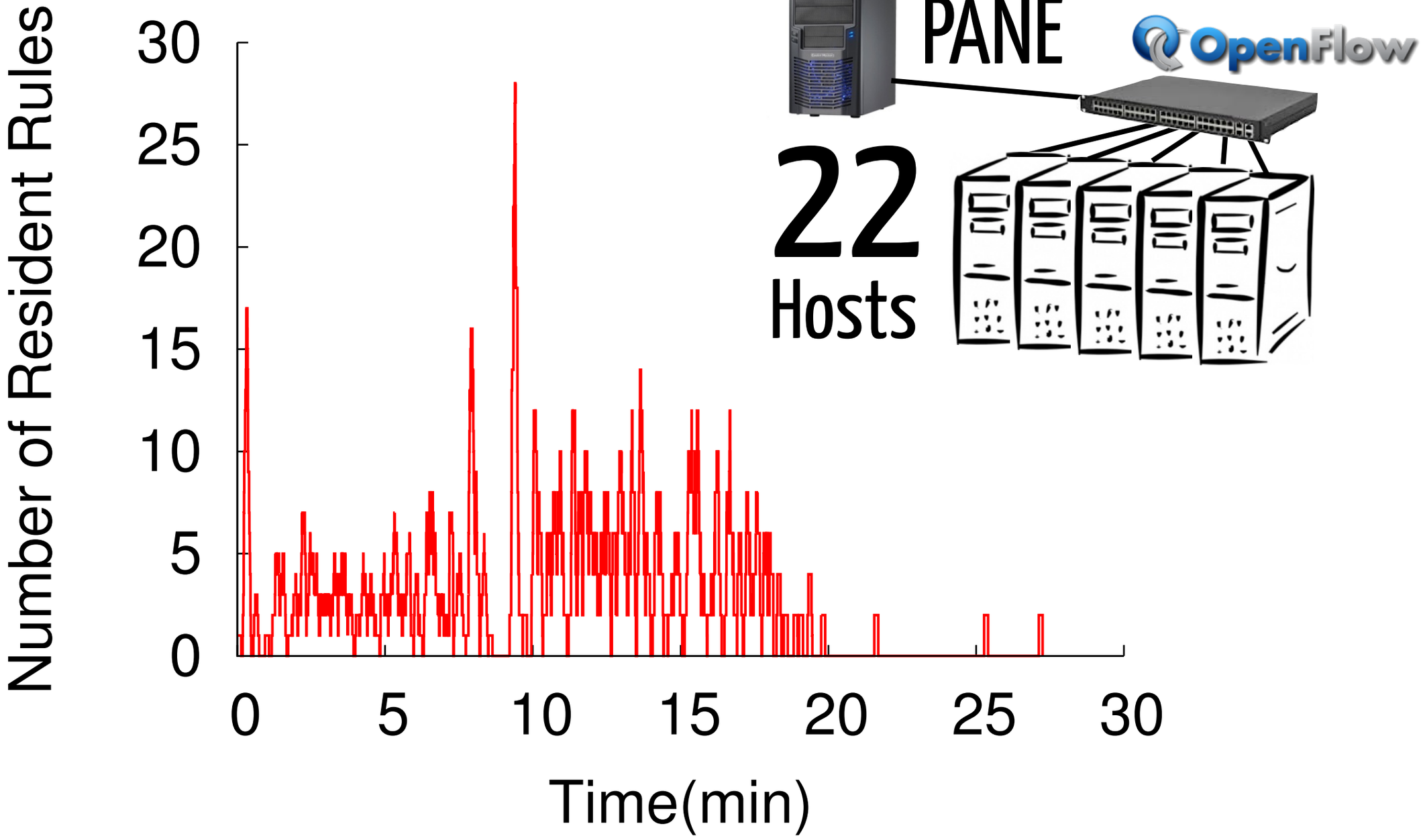
Hadoop's OpenFlow rules



Hadoop's OpenFlow rules



Hadoop's OpenFlow rules



Hadoop's OpenFlow rules

1. For applications that know what they want from the network
2. Allows these applications to co-exist

Conclusion

pane.cs.brown.edu

Andrew Ferguson
adf@cs.brown.edu

Co-authors

- **Arjun Guha**
  Brown ↦ Cornell ↦ UMass Amherst
- **Chen Liang**
  Brown ↦ Duke
- **Rodrigo Fonseca**
  Brown
- **Shriram Krishnamurthi**
  Brown



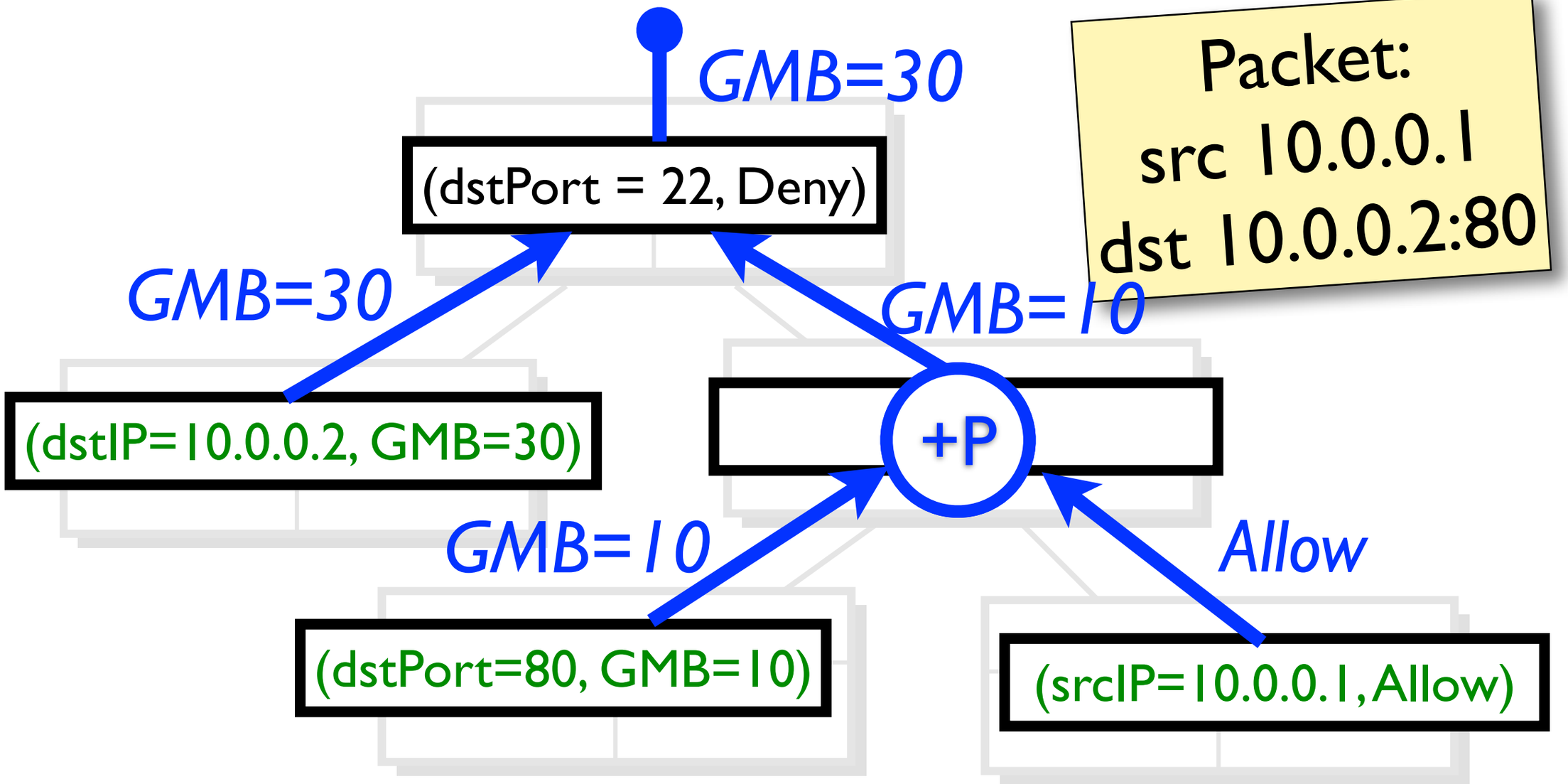

pane.cs.brown.edu

Andrew Ferguson
adf@cs.brown.edu

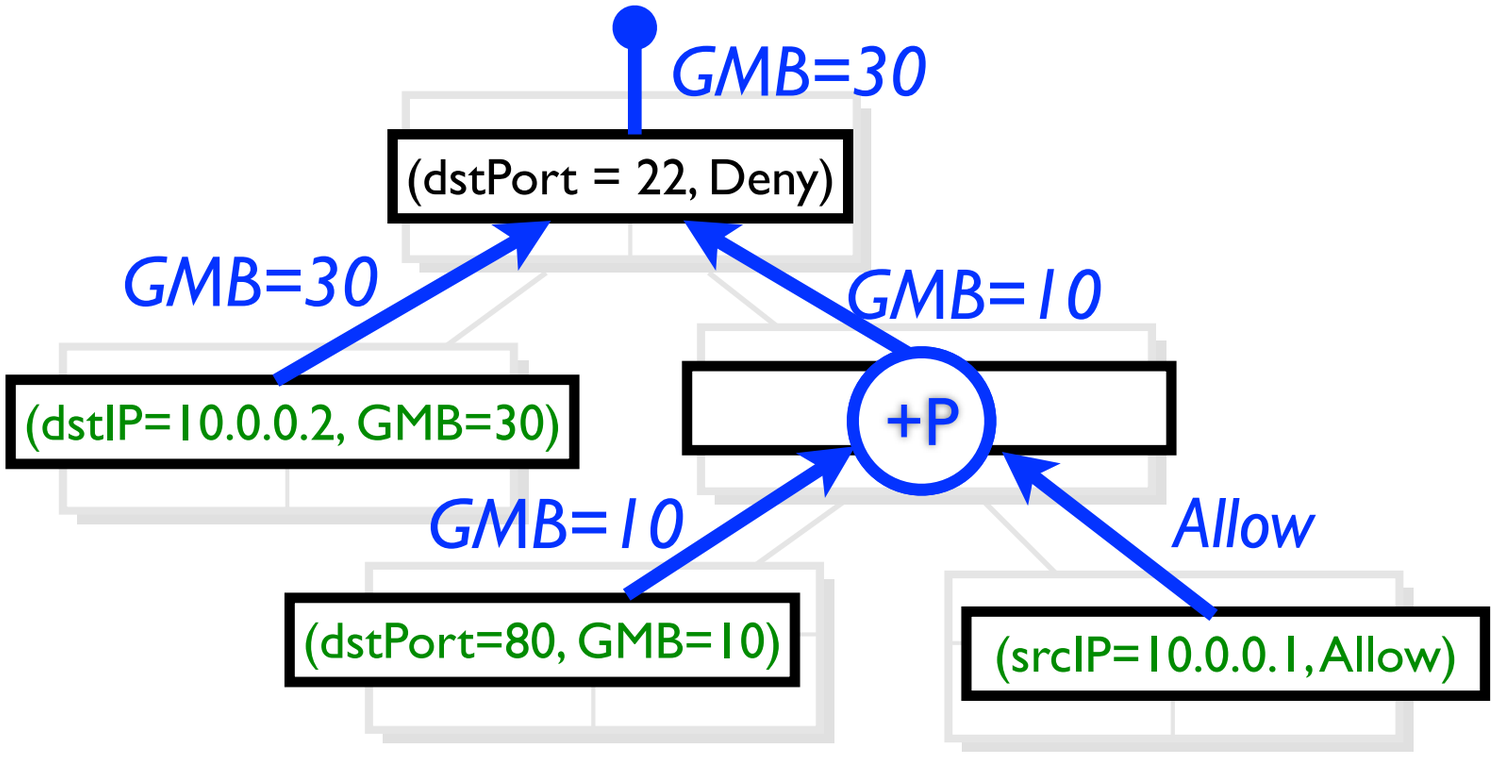
Backup

Slides

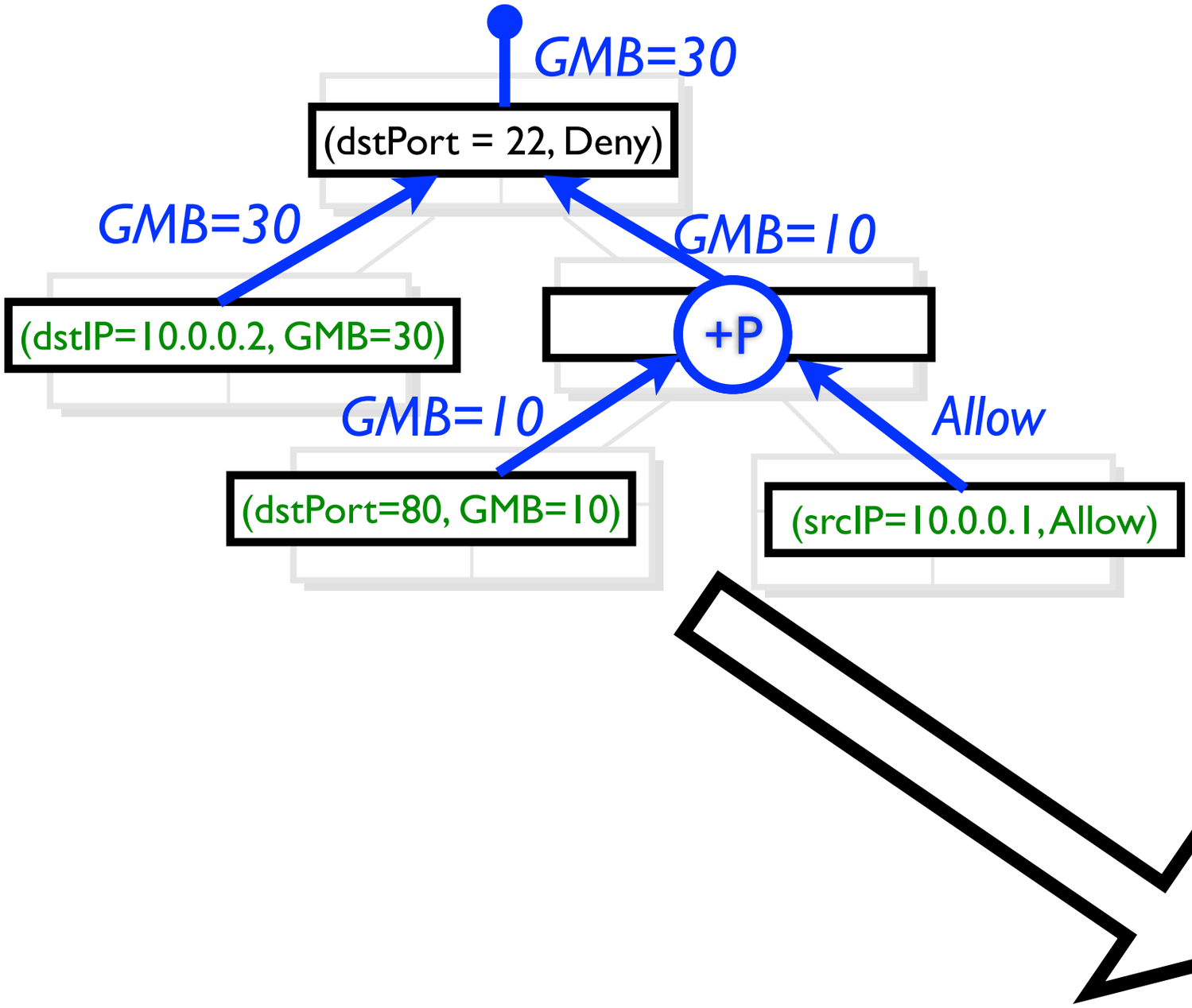
Proof of Correctness



Hierarchical Flow Tables

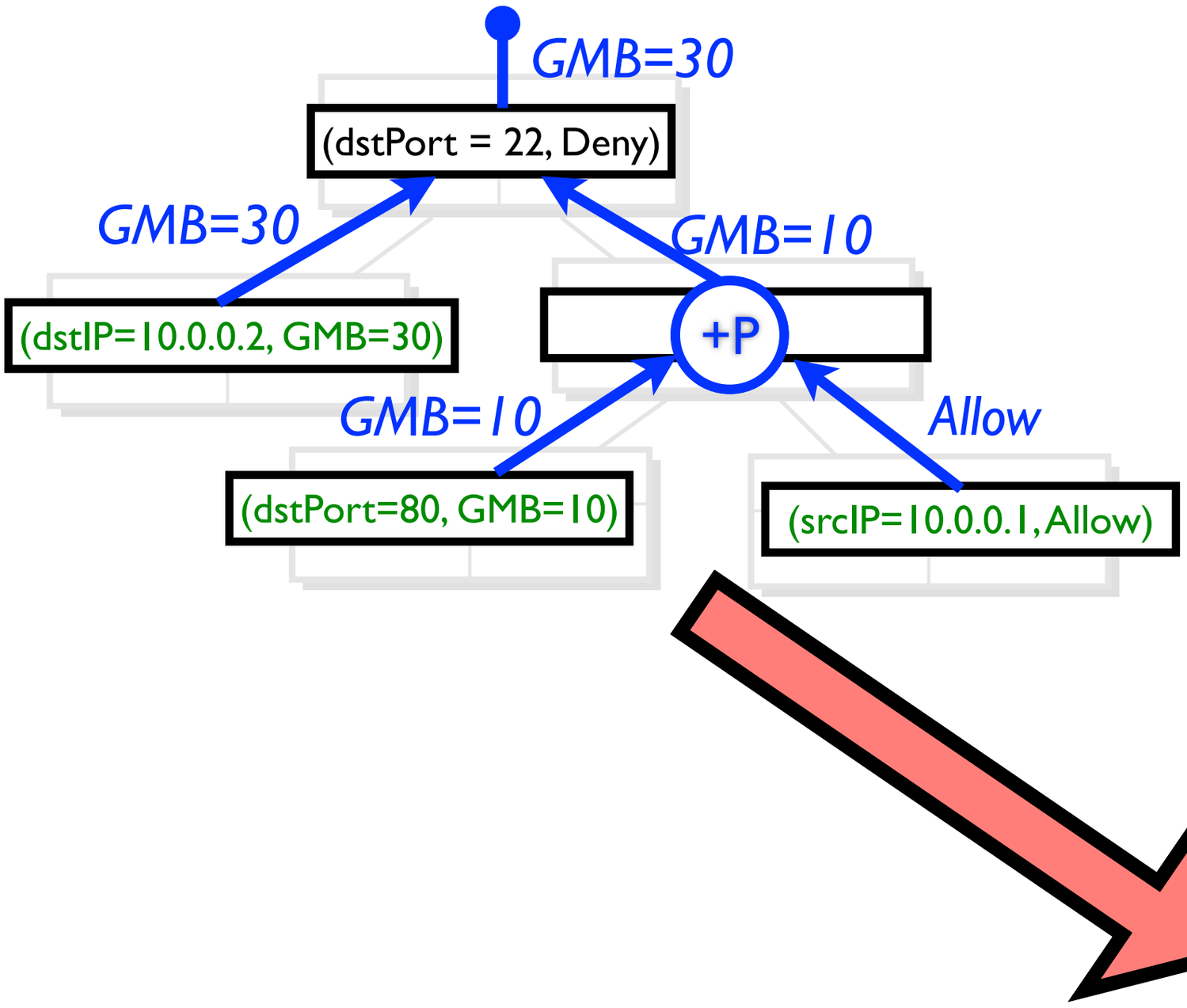


Compiler Correctness



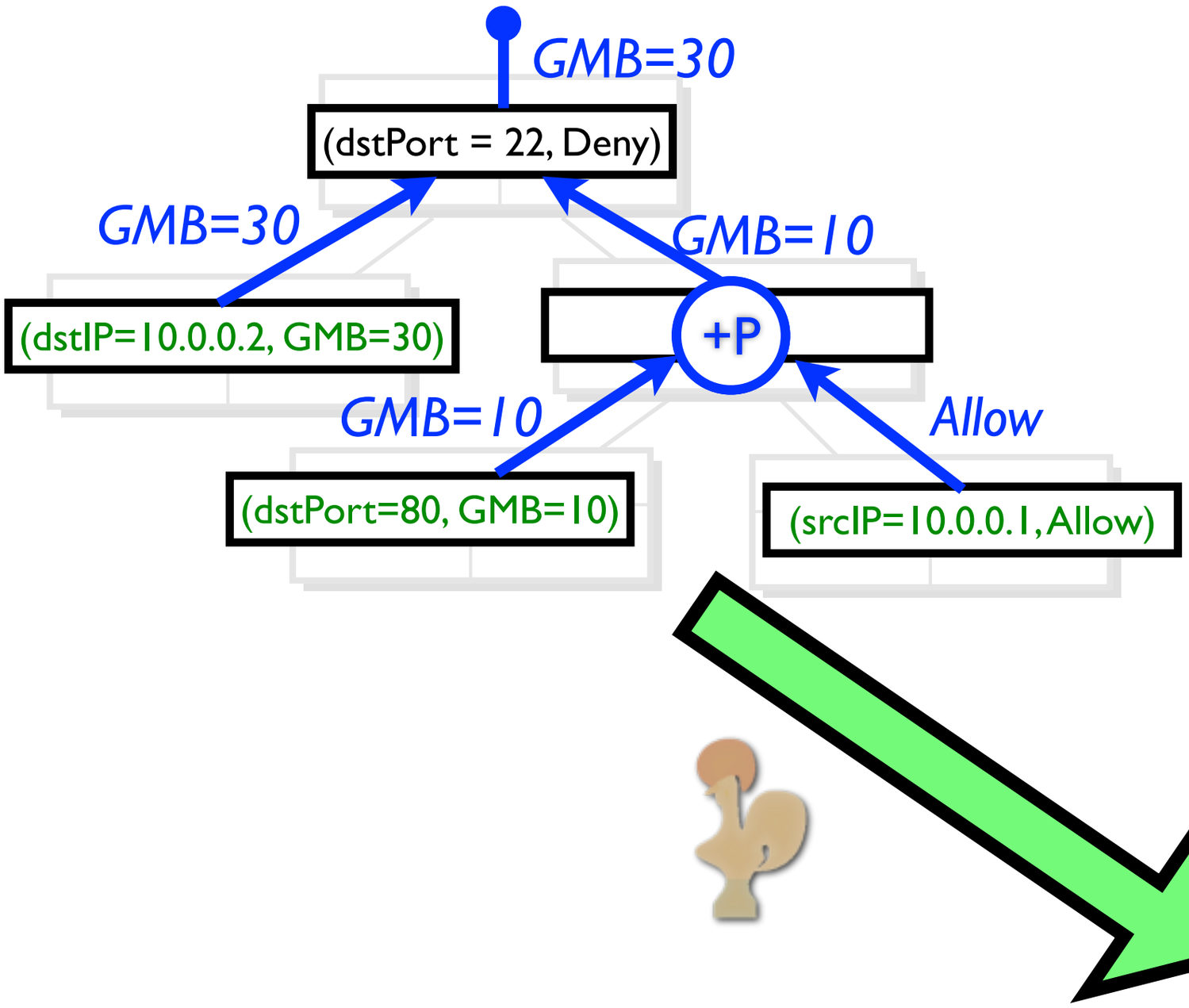
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6
*	*	*	*	*	*	*	*	*	22	drop

Compiler Correctness



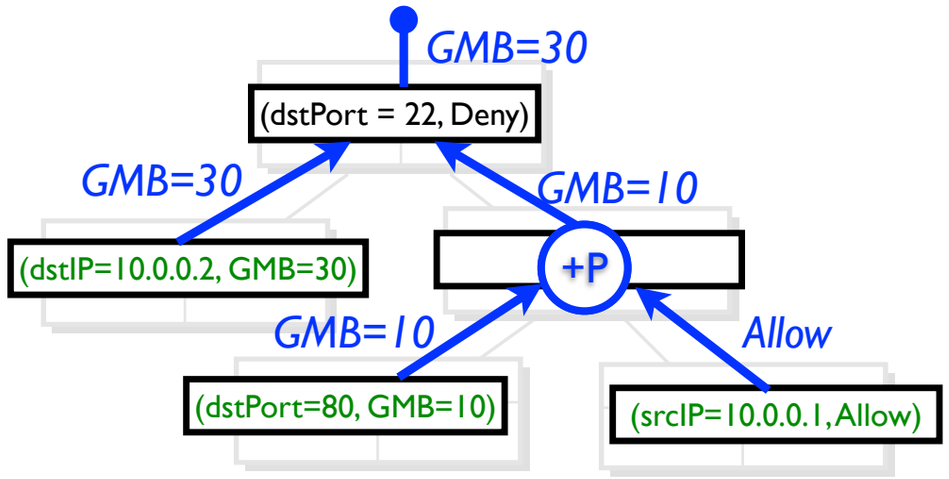
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6
*	*	*	*	*	*	*	*	*	22	drop

Compiler Correctness



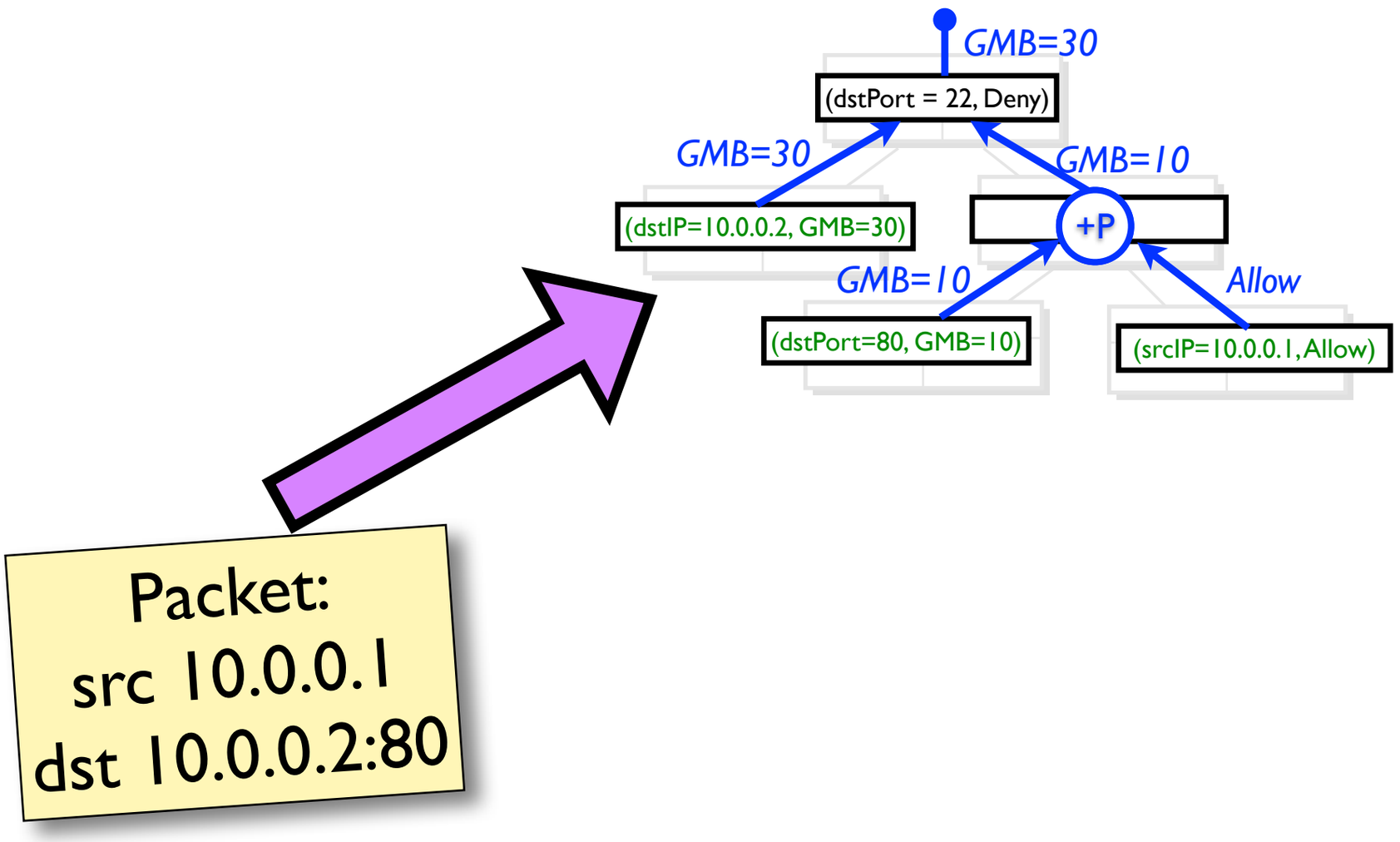
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6
*	*	*	*	*	*	*	*	*	22	drop

Coq Proof Assistant

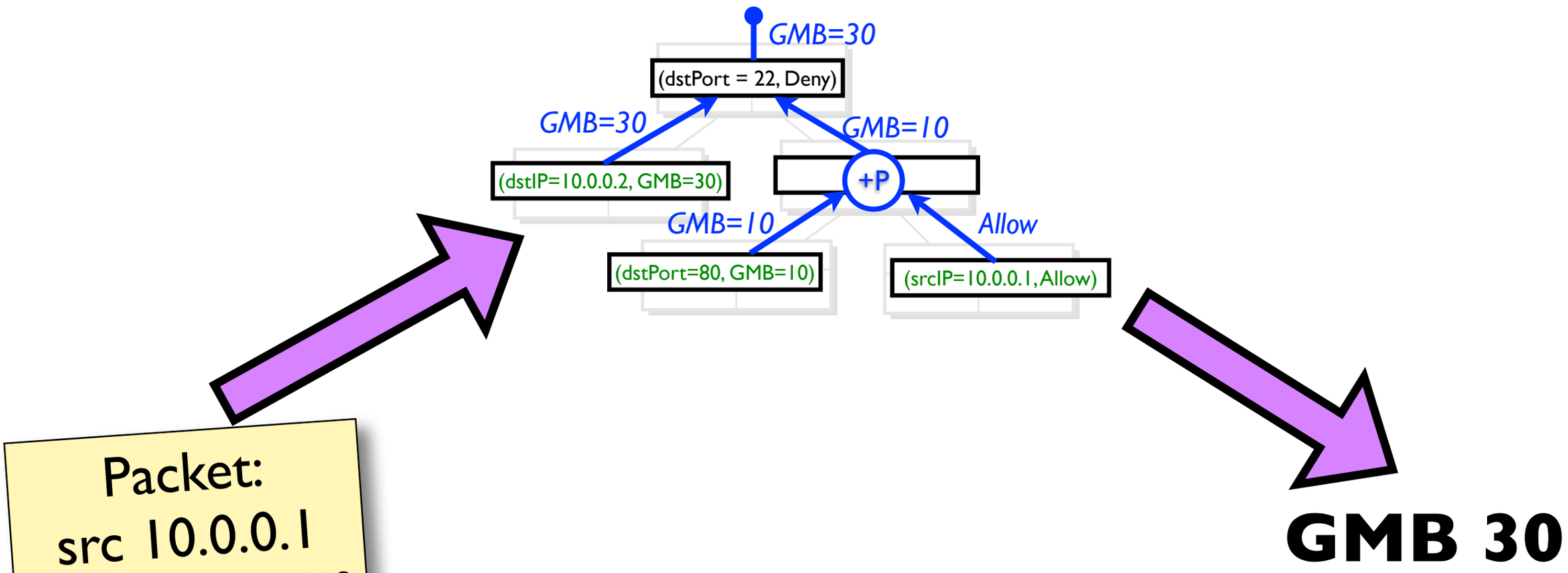


Packet:
src 10.0.0.1
dst 10.0.0.2:80

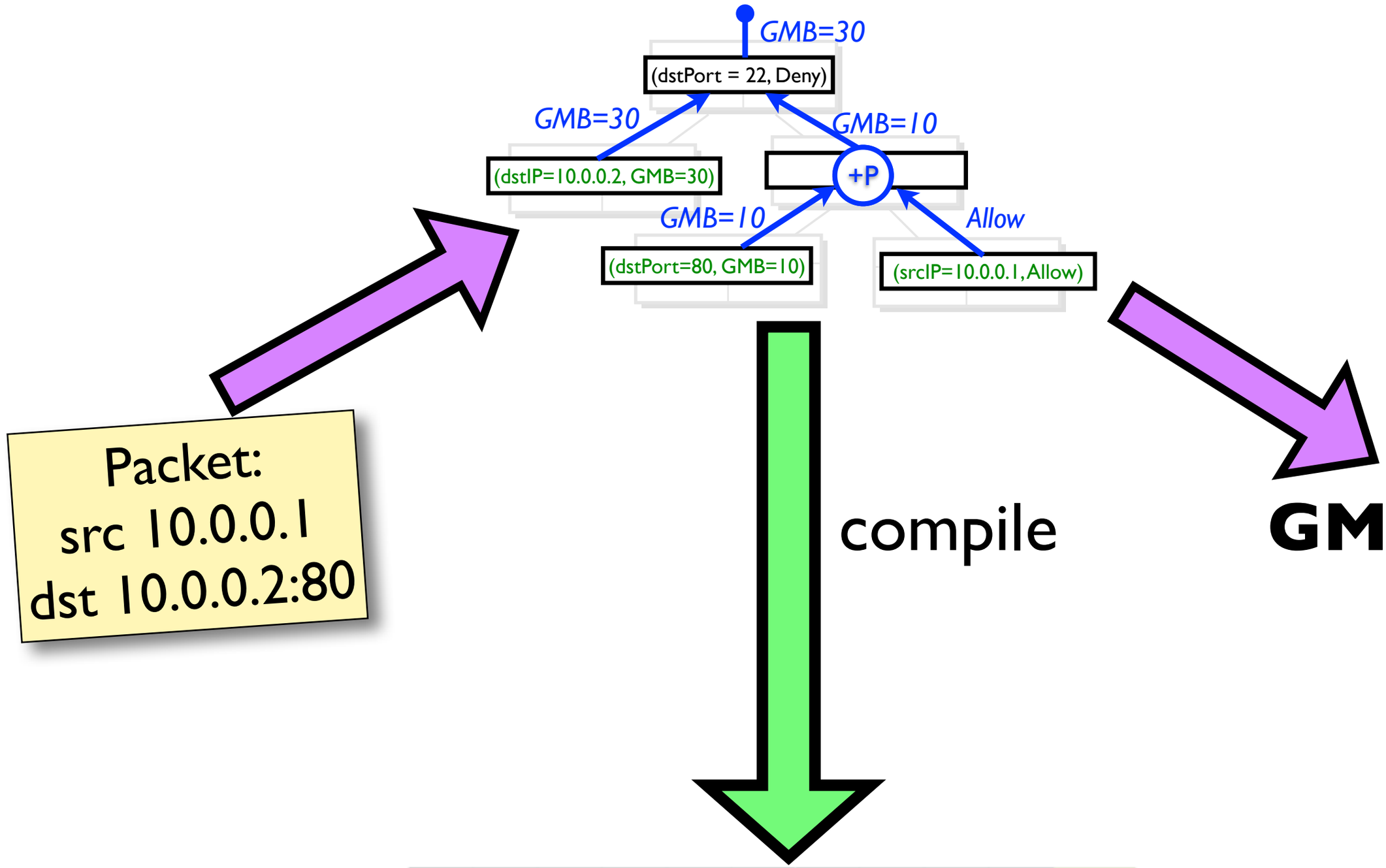
Theorem


GMB=30
(dstPort = 22, Deny)
GMB=30
GMB=10
(dstIP=10.0.0.2, GMB=30)
+P
GMB=10
Allow
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)
Packet:
src 10.0.0.1
dst 10.0.0.2:80

Theorem

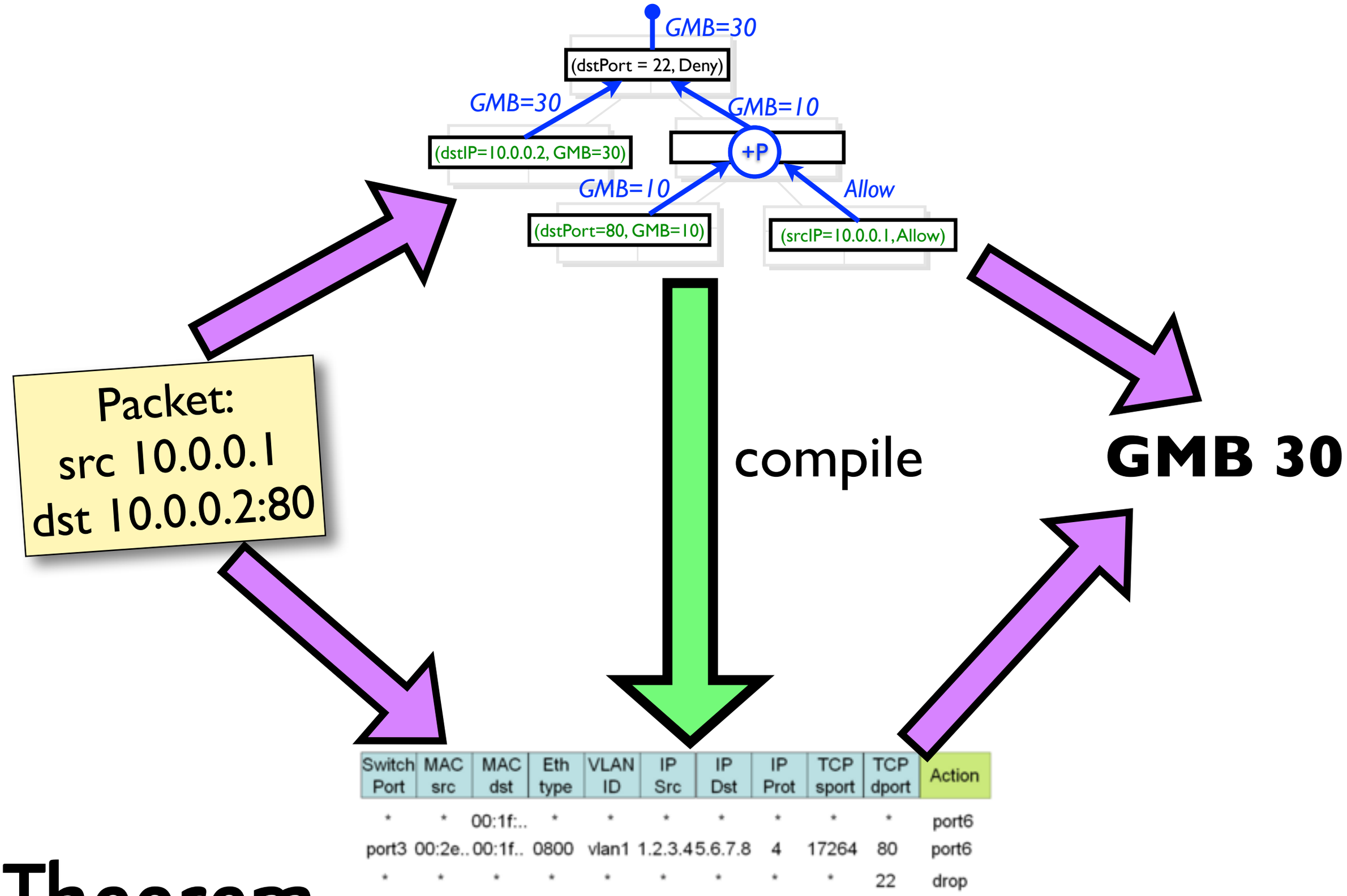

GMB=30
(dstPort = 22, Deny)
GMB=30
GMB=10
(dstIP=10.0.0.2, GMB=30)
+P
GMB=10
Allow
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)
Packet:
src 10.0.0.1
dst 10.0.0.2:80
GMB 30

Theorem



Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6
port3	00:2e..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6
*	*	*	*	*	*	*	*	*	22	drop

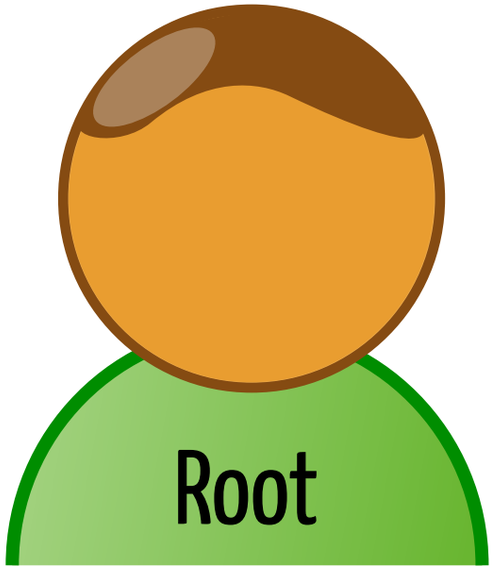
Theorem


GMB=30
(dstPort = 22, Deny)
GMB=30
GMB=10
(dstIP=10.0.0.2, GMB=30)
+P
GMB=10
Allow
(dstPort=80, GMB=10)
(srcIP=10.0.0.1, Allow)
Packet:
src 10.0.0.1
dst 10.0.0.2:80
compile
GMB 30
Switch Port
MAC src
MAC dst
Eth type
VLAN ID
IP Src
IP Dst
IP Prot
TCP sport
TCP dport
Action
* * 00:1f:.. * * * * * * * port6
port3 00:2e.. 00:1f.. 0800 vlan1 1.2.3.4 5.6.7.8 4 17264 80 port6
* * * * * * * * * 22 drop

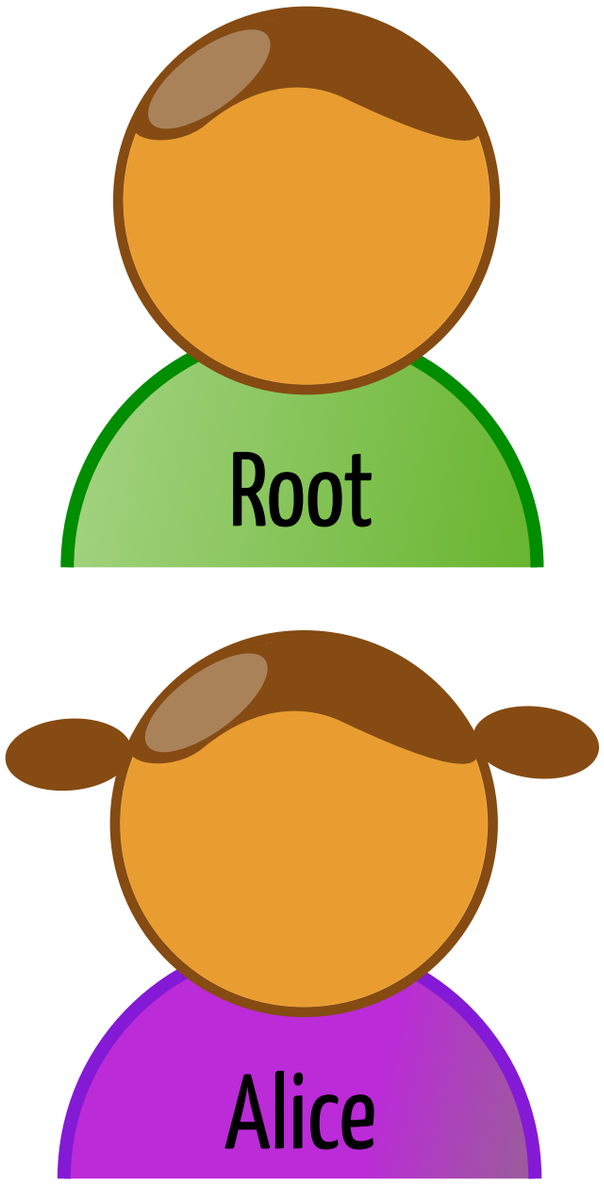
Theorem

Protocol

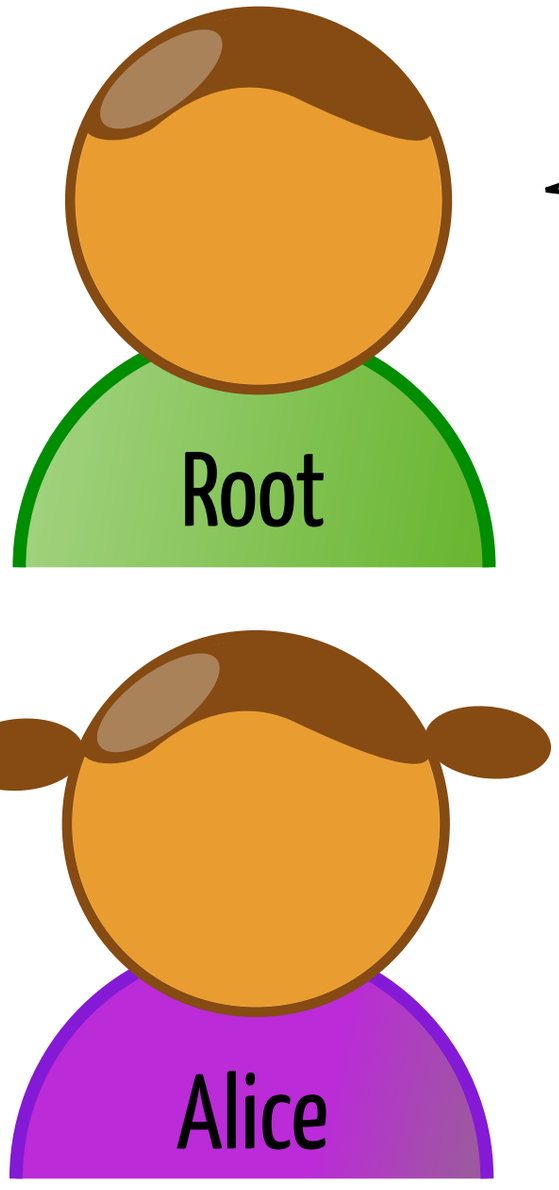
PANE



PANE

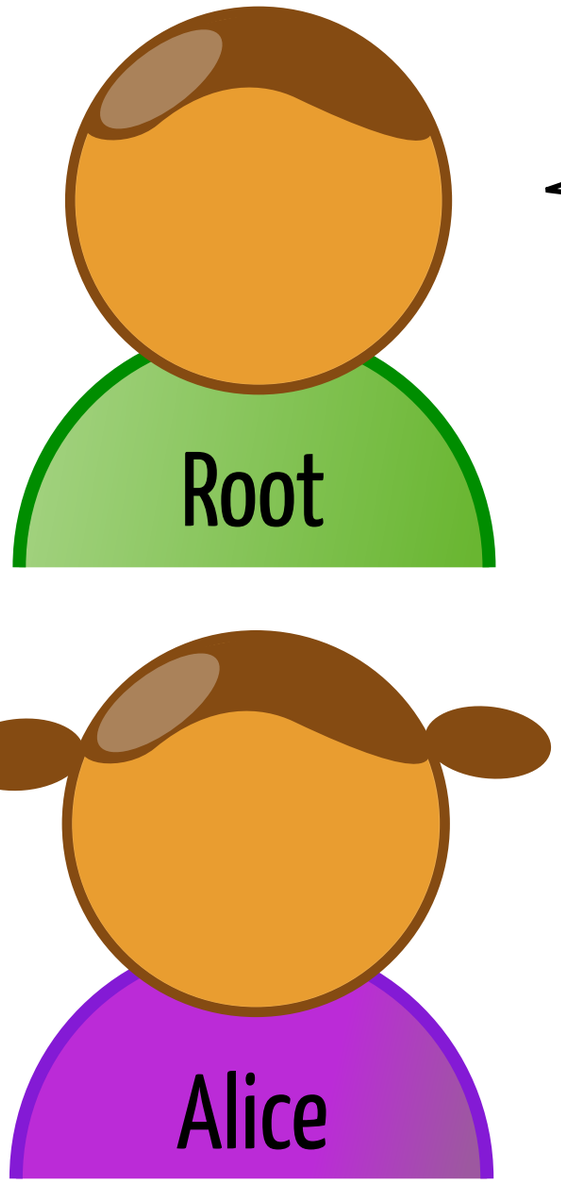


PANE




```
NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.
```

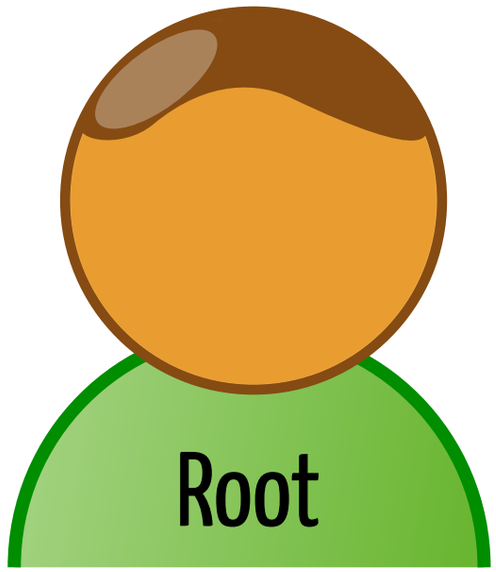
PANE



NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

OK

PANE

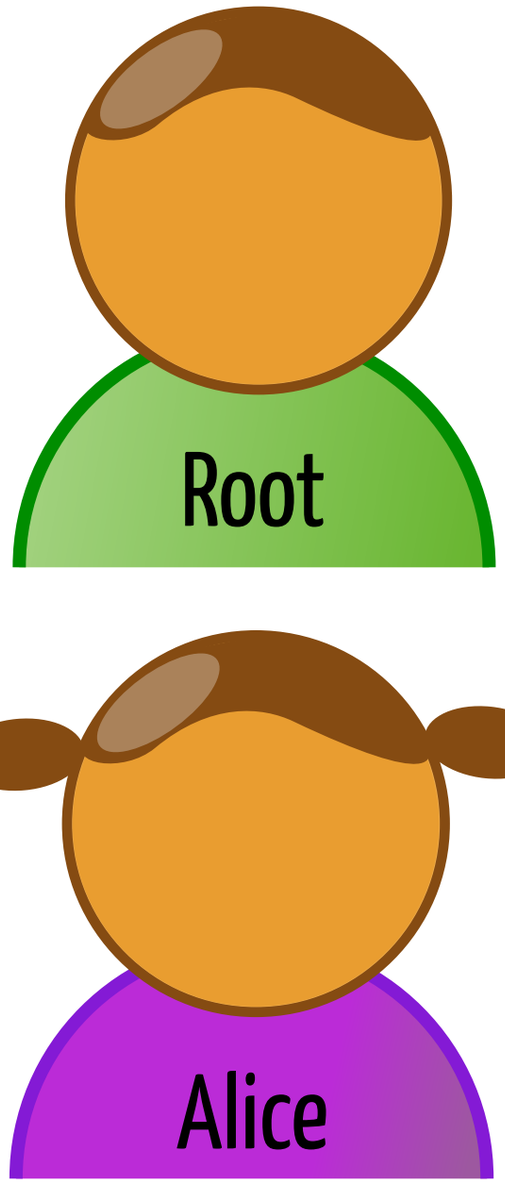


NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

OK

Grant aBW to Alice.

PANE



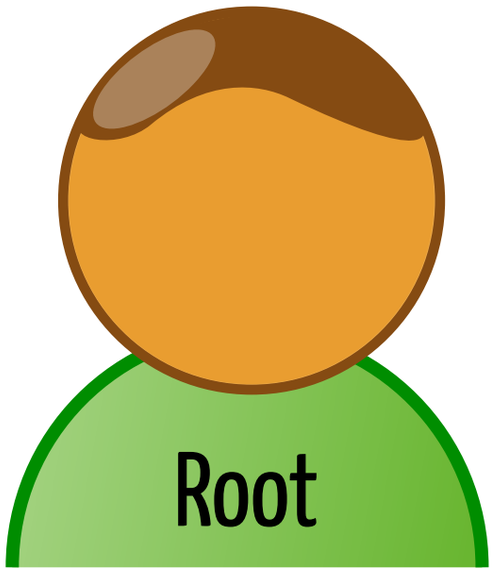
NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

OK

Grant aBW to Alice.

OK

PANE

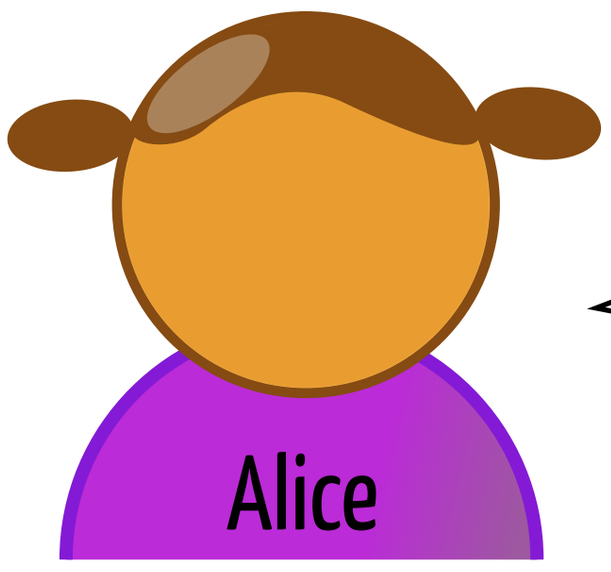


NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

OK

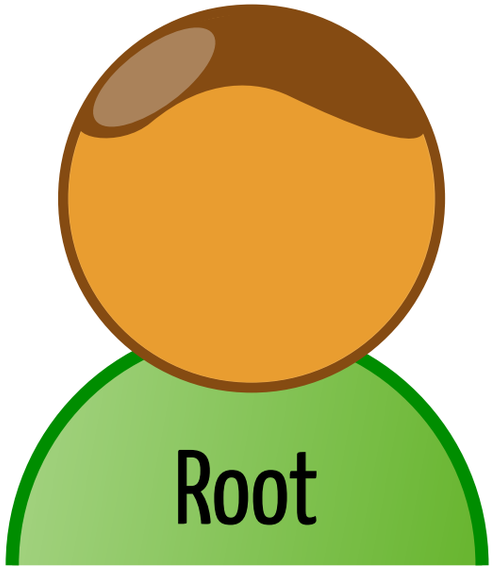
Grant aBW to Alice.

OK



reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.

PANE

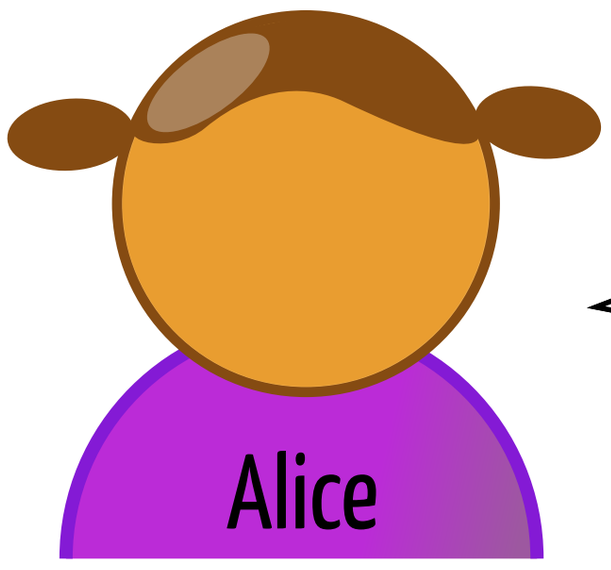


NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

OK

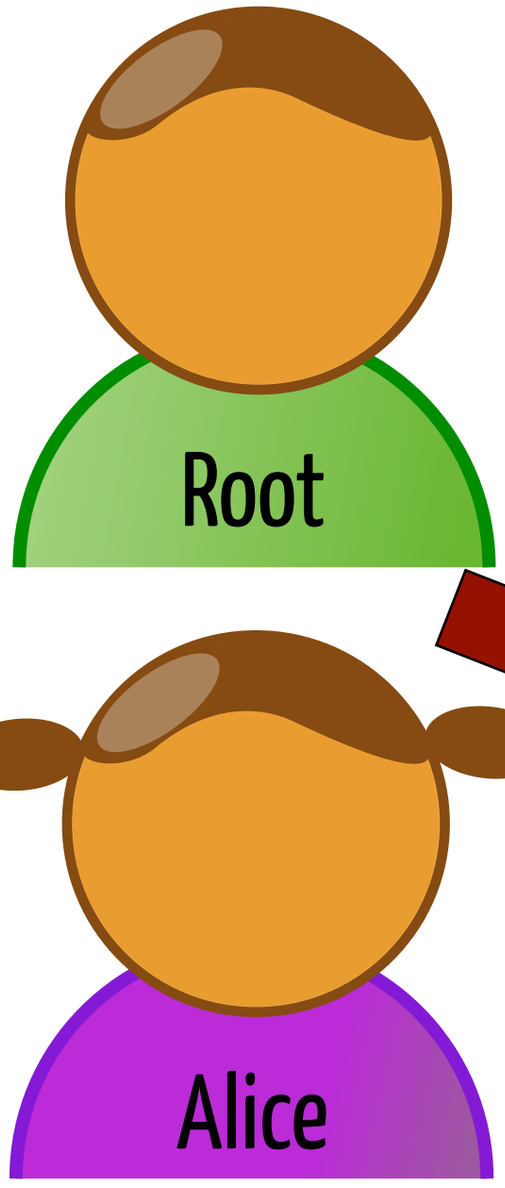
Grant aBW to Alice.

OK



reserve(user=Alice,
dstPort=80) = 5Mb on **aBW**
from now to +10min.

PANE



NewShare aBW for
(user=Alice) [reserve <= 10Mb]
on rootShare.

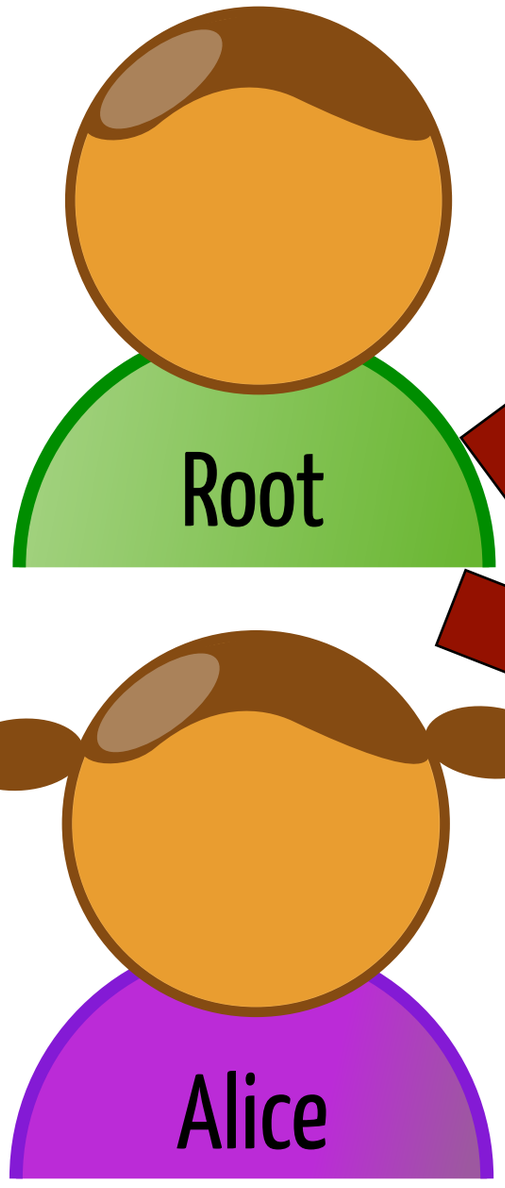
OK

Grant aBW to Alice.

OK

reserve(**user=Alice,**
dstPort=80) = 5Mb on aBW
from now to +10min.

PANE



NewShare aBW for
(**user=Alice**) [reserve <= 10Mb]
on rootShare.

OK

Grant aBW to Alice.

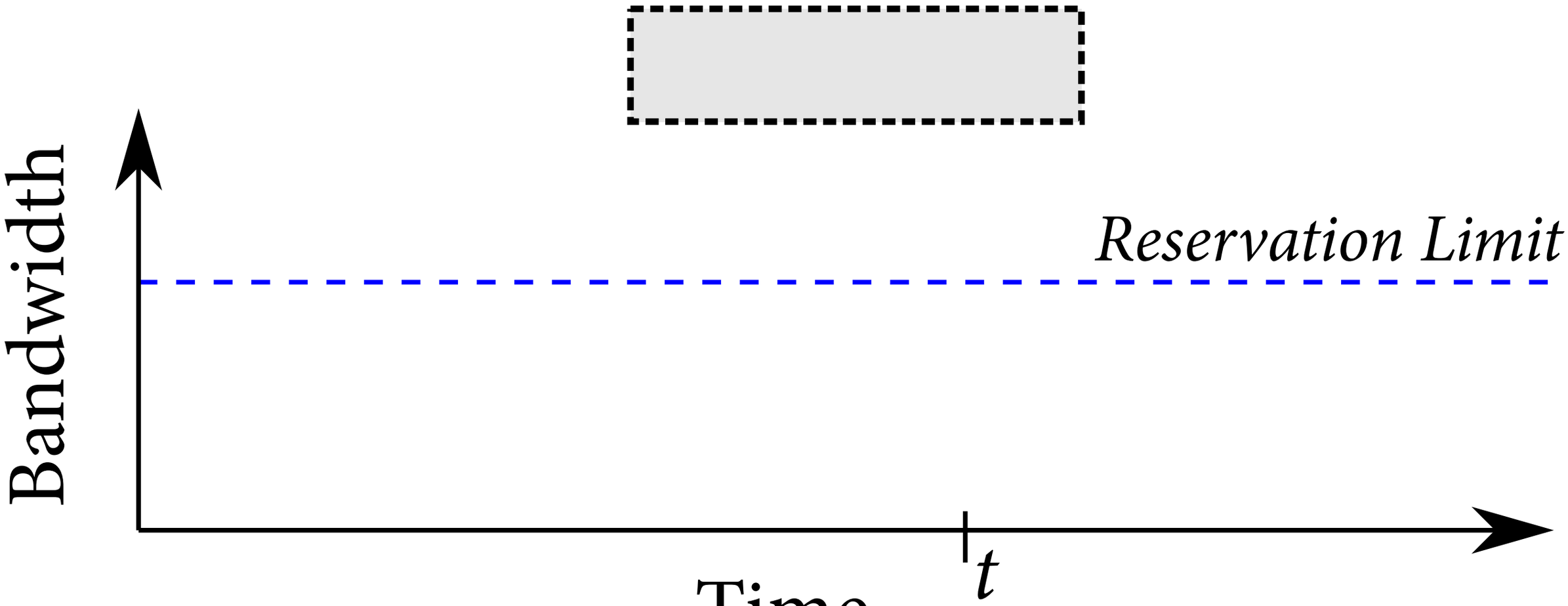
OK

reserve(**user=Alice,**
dstPort=80) = 5Mb on aBW
from now to +10min.

PANE

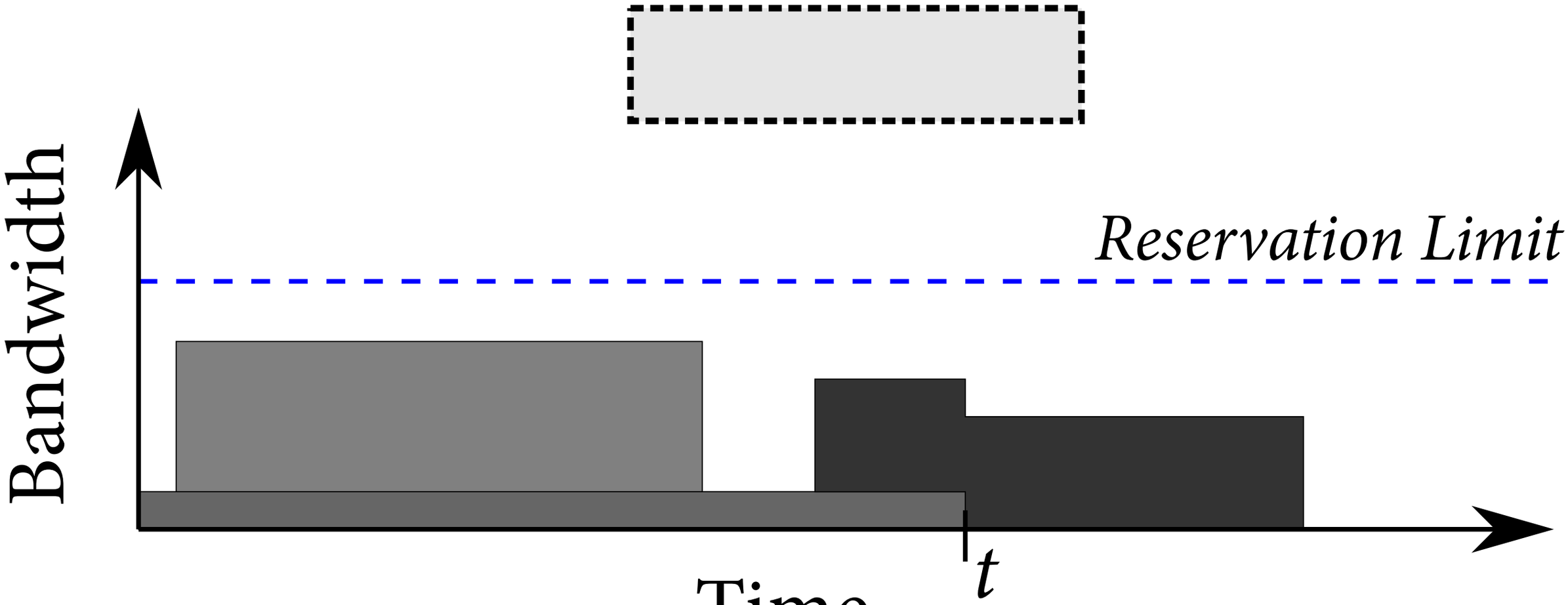
```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

PANE



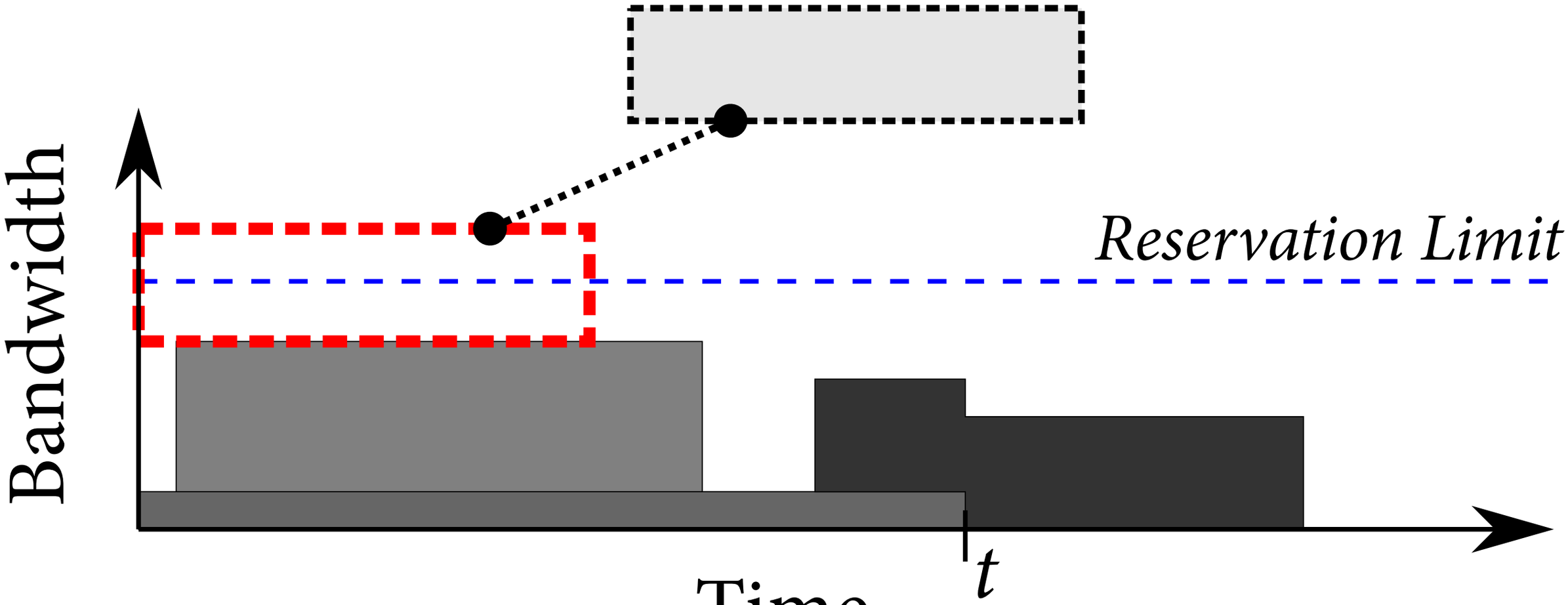

```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

PANE



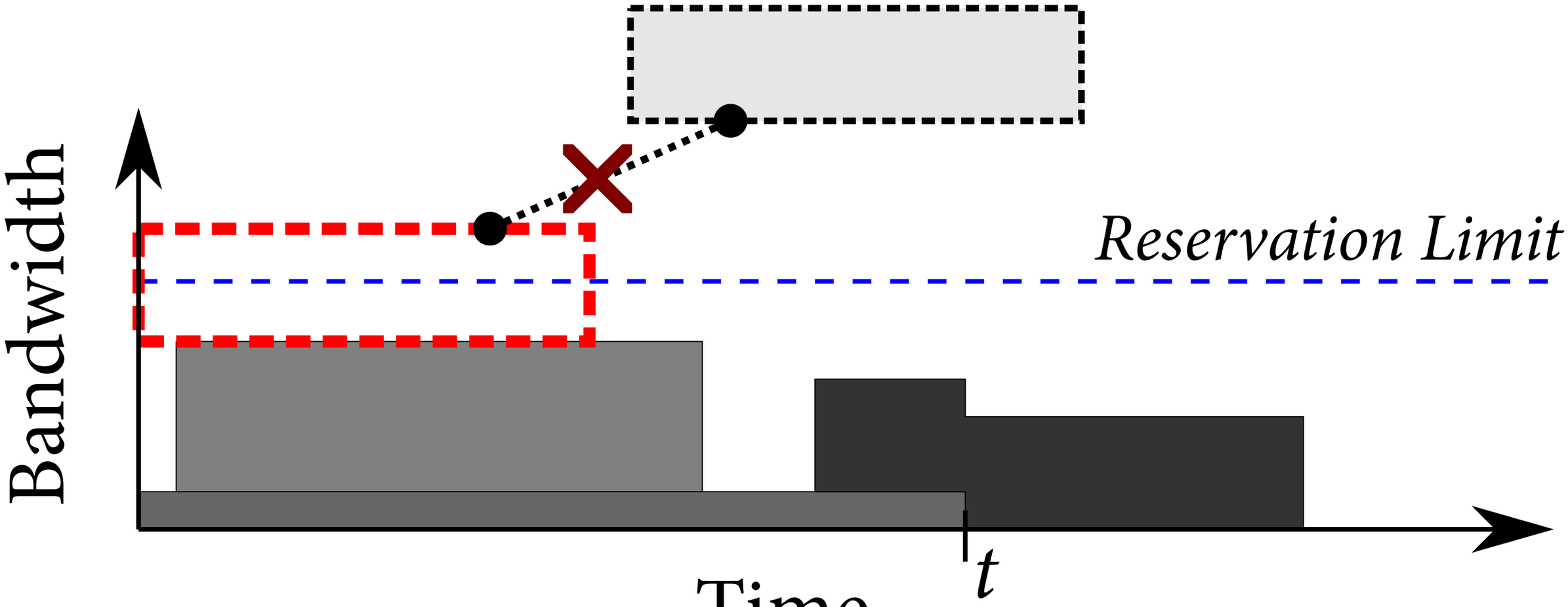

```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

PANE




```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

PANE




```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

PANE


```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

NO

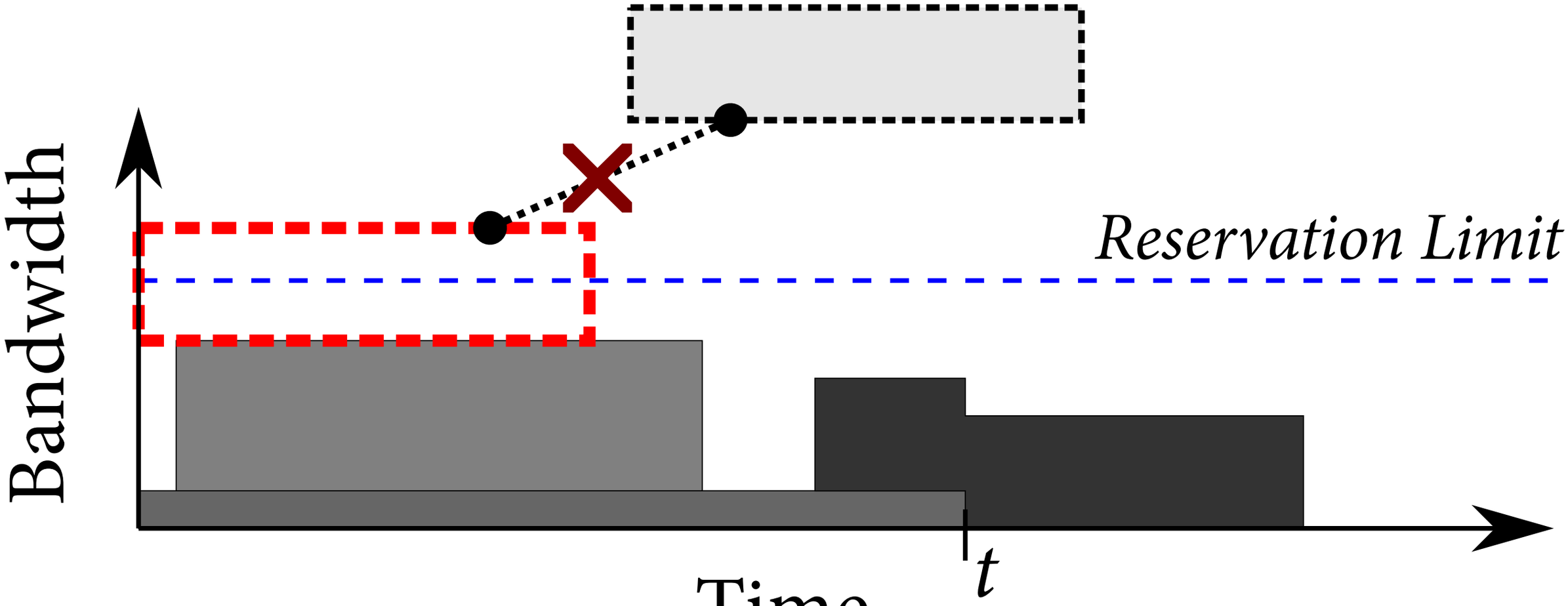
PANE


```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

NO

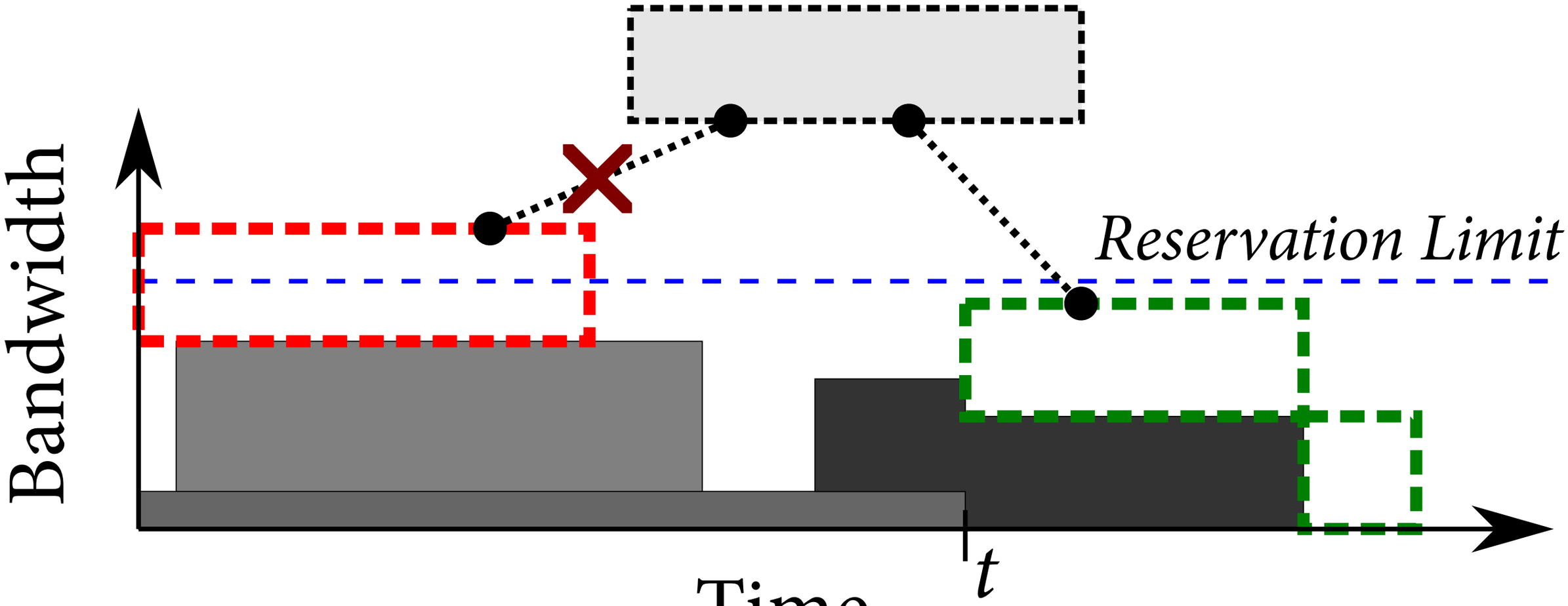
```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from +20min to +30min.
```

PANE



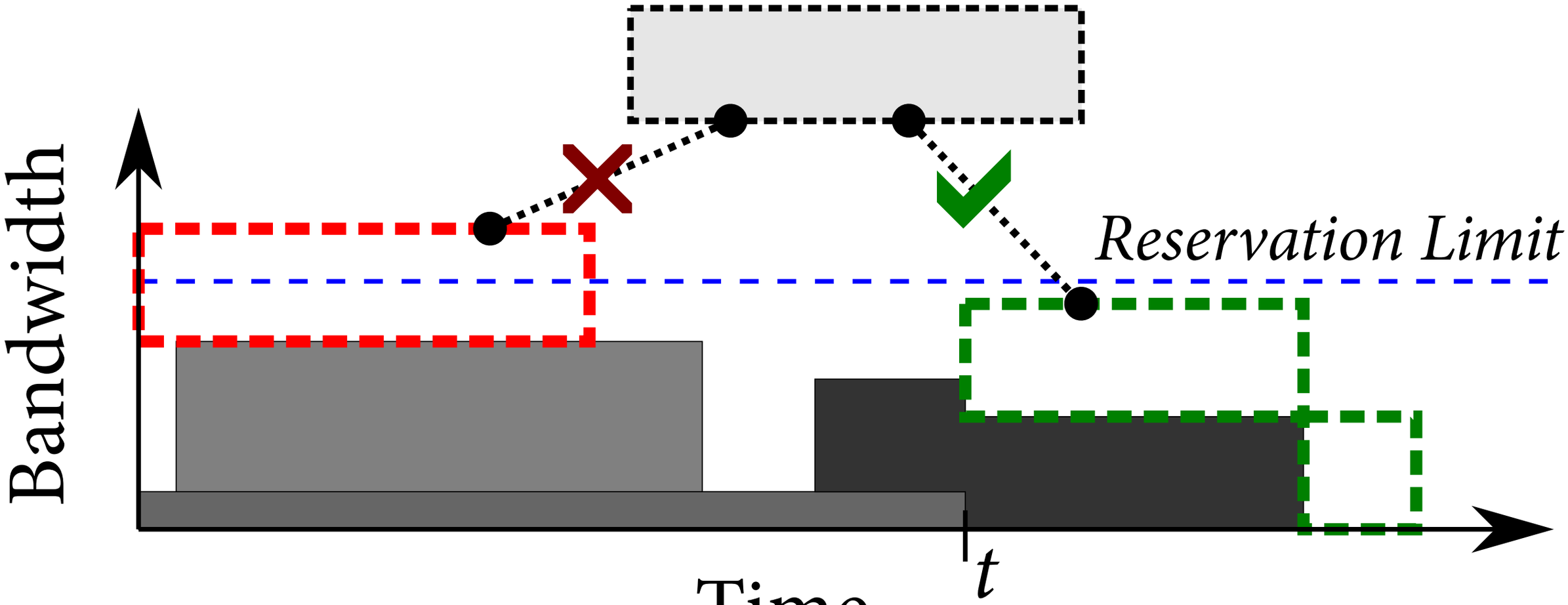

```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from +20min to +30min.
```

PANE




```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from +20min to +30min.
```

PANE




```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from +20min to +30min.
```

PANE


```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from now to +10min.
```

NO

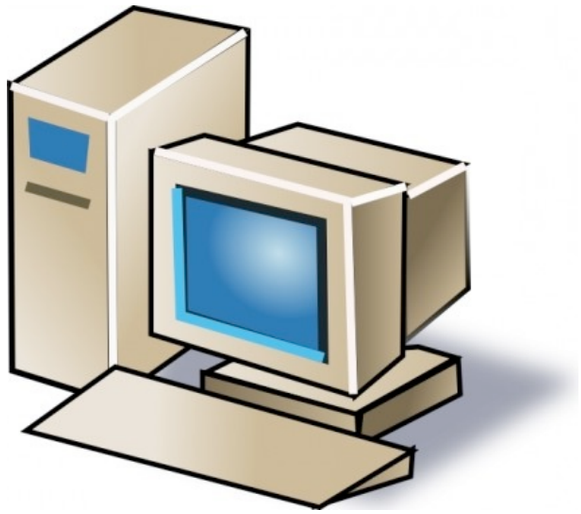
```
reserve(user=Alice,
dstPort=80) = 5Mb on aBW
from +20min to +30min.
```

OK

PANE

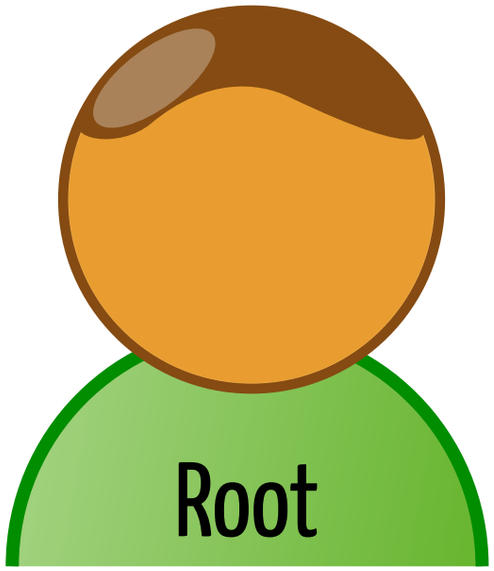
PANE

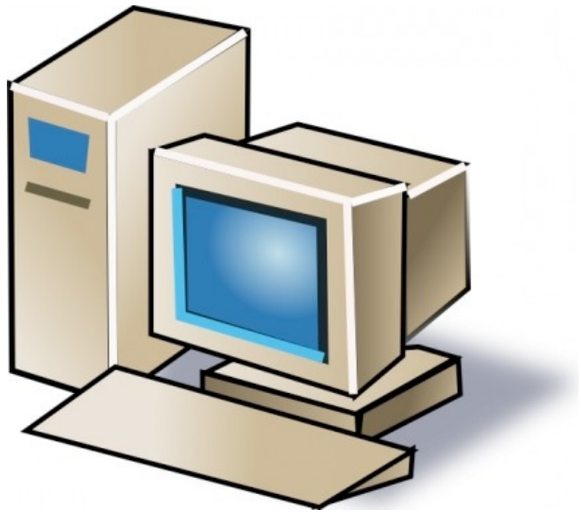
PANE



10.0.0.2

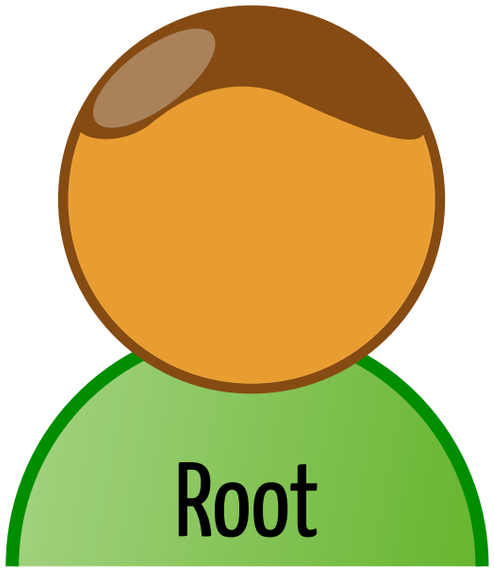
PANE



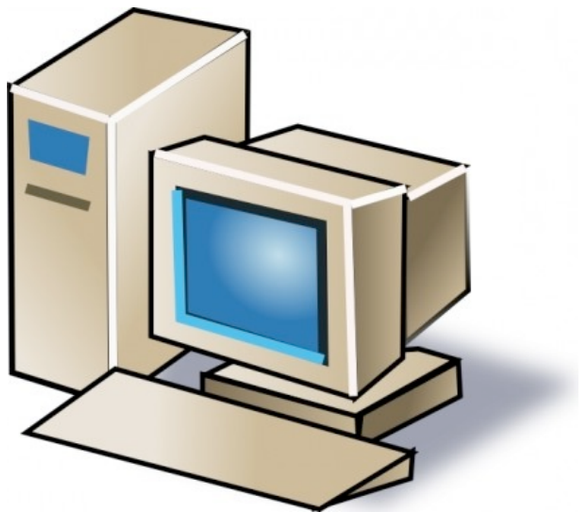


10.0.0.2

PANE

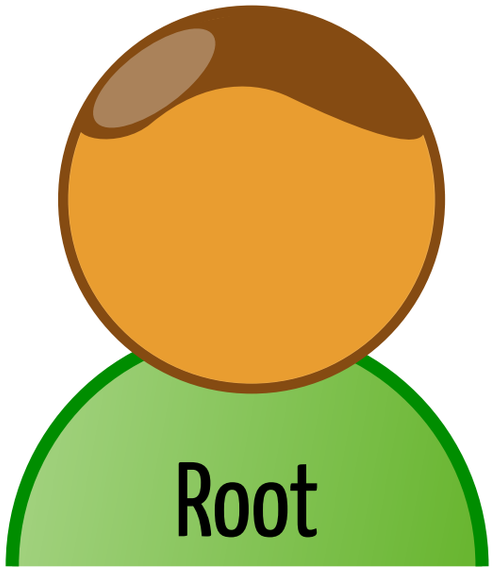


NewShare aAC for
(dstHost=10.0.0.2) [deny = True]
on rootShare.



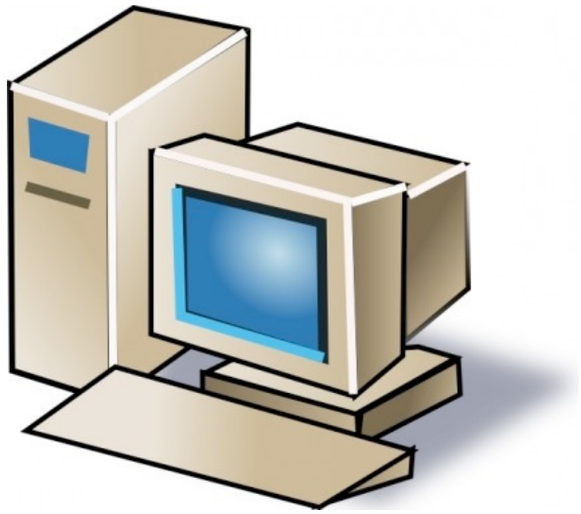
10.0.0.2

PANE



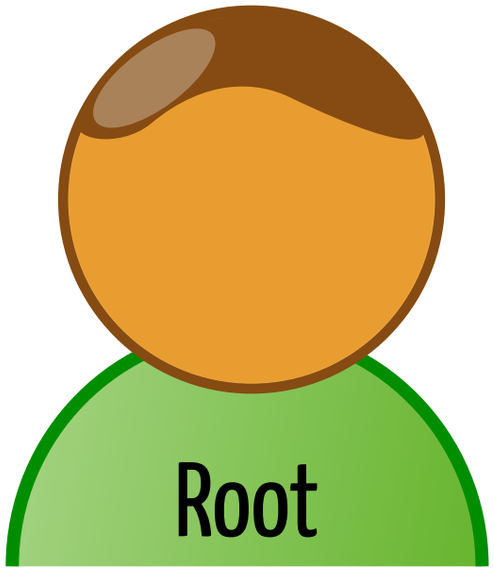
NewShare aAC for
(dstHost=10.0.0.2) [deny = True]
on rootShare.

OK



10.0.0.2

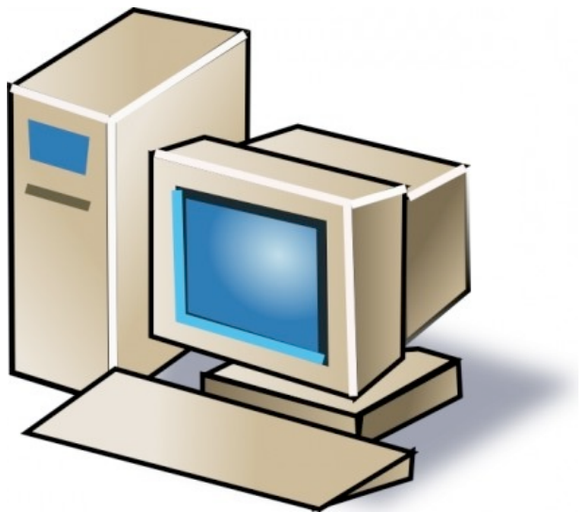
PANE



NewShare aAC for
(dstHost=10.0.0.2) [deny = True]
on rootShare.

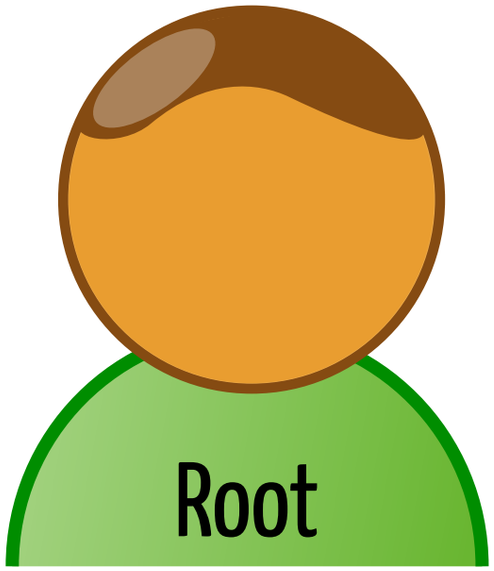
OK

Grant aAC to Alice.



10.0.0.2

PANE

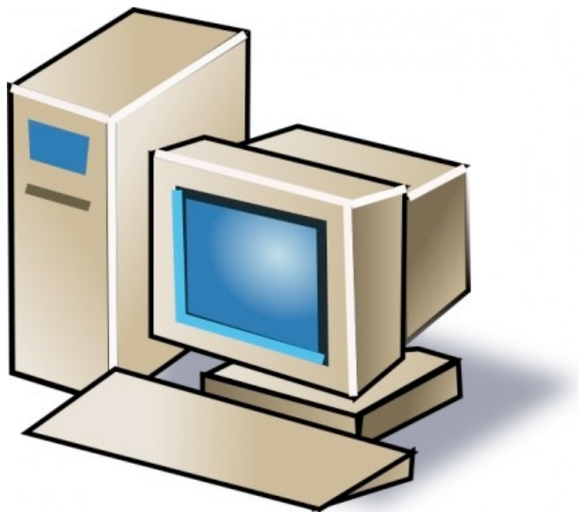


NewShare aAC for
(dstHost=10.0.0.2) [deny = True]
on rootShare.

OK

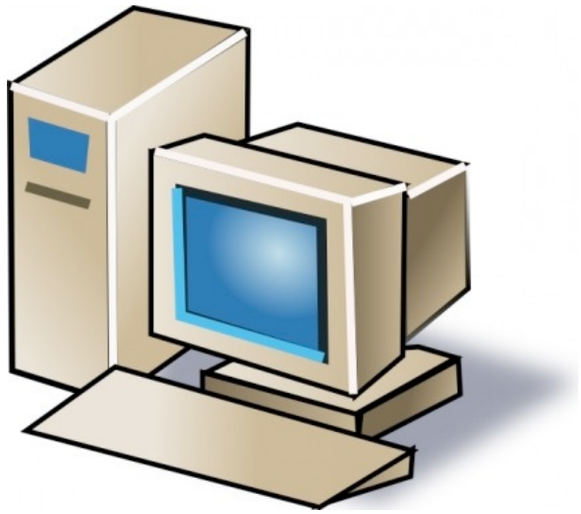
Grant aAC to Alice.

OK



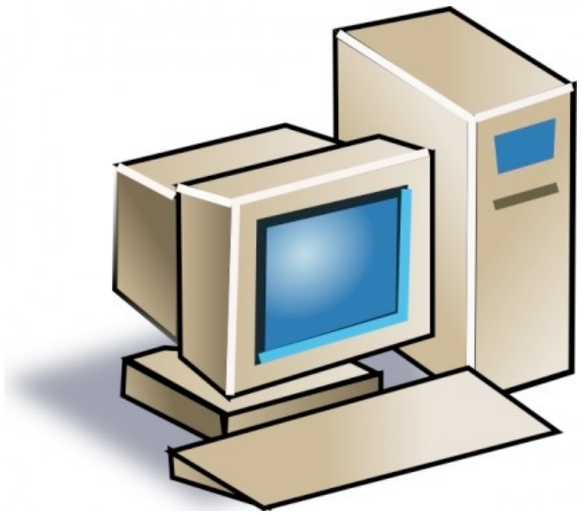
10.0.0.2

PANE

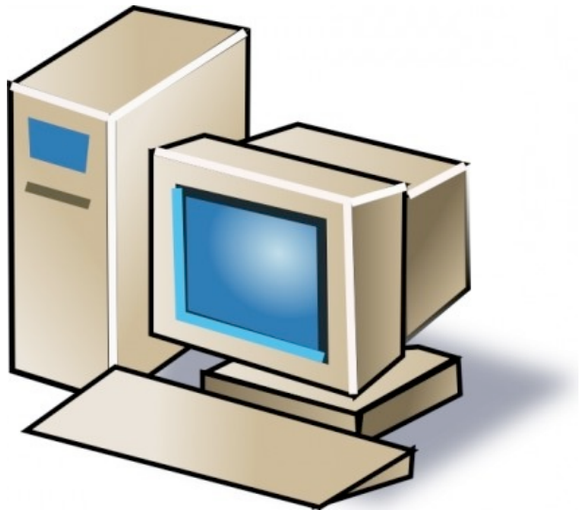


10.0.0.2

PANE



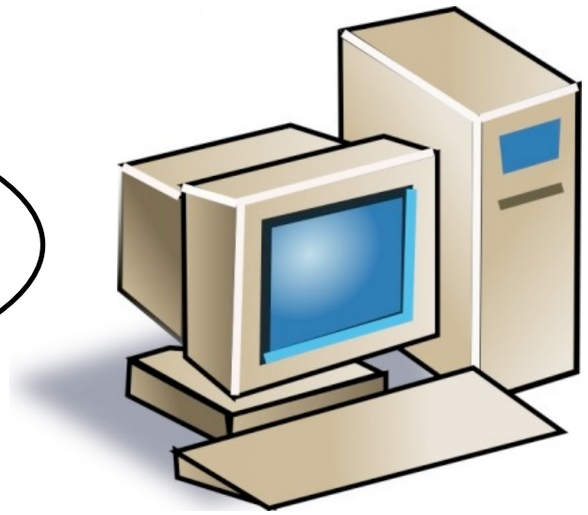
10.0.0.3



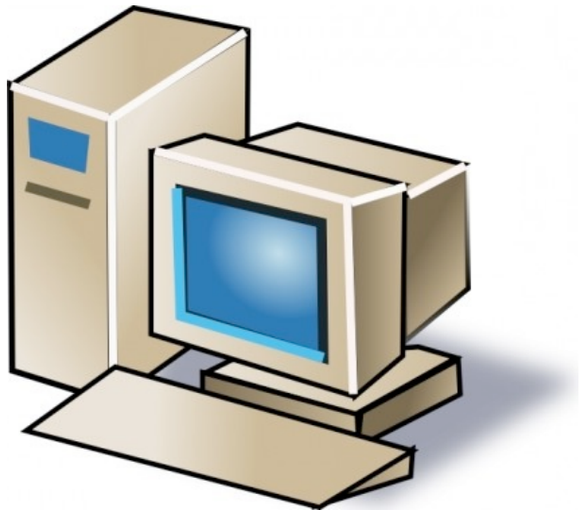
10.0.0.2

PANE

deny(dstHost=10.0.0.2,
srcHost=10.0.0.3) on aAC
from now to +5min.



10.0.0.3

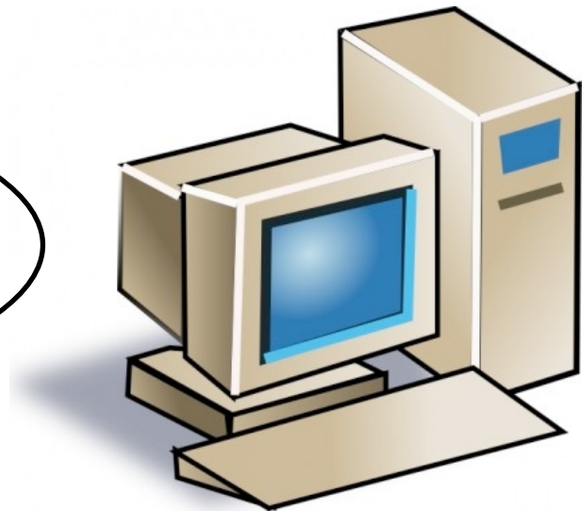


10.0.0.2

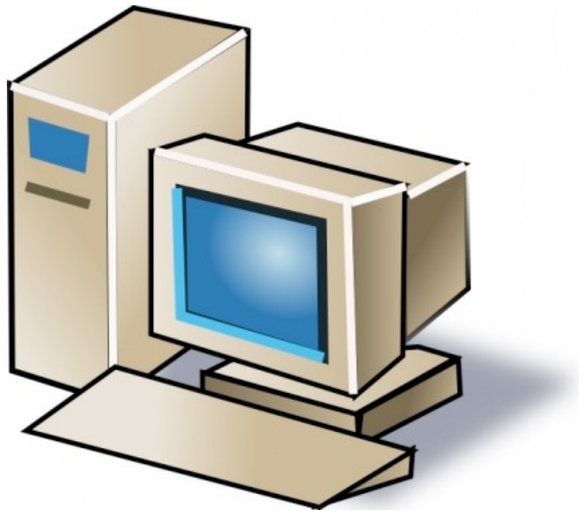
PANE

```
deny(dstHost=10.0.0.2,
srcHost=10.0.0.3) on aAC
from now to +5min.
```

OK



10.0.0.3

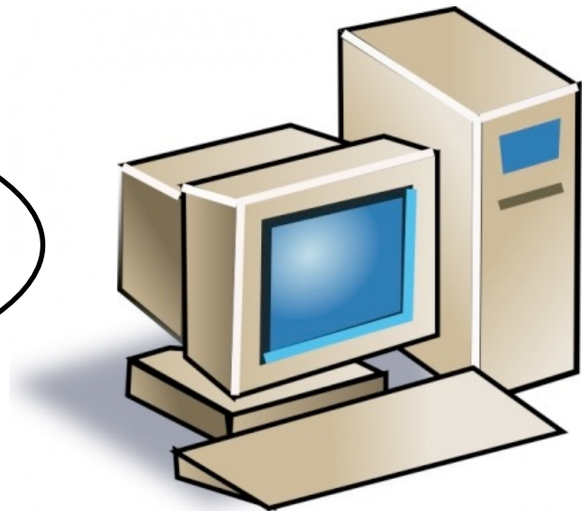


10.0.0.2

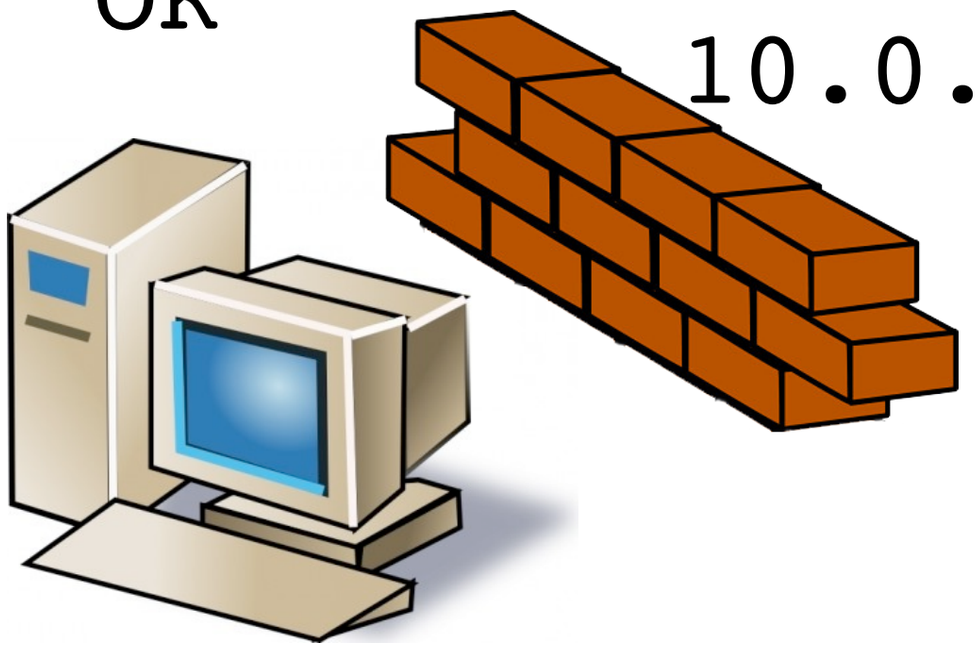
PANE

deny(dstHost=10.0.0.2,
srcHost=10.0.0.3) on aAC
from now to +5min.

OK



10.0.0.3



10.0.0.2

PANE

Netflix


YouTube
facebook
PWR SYS WLAN 4 3 2 1 WAN USB QSS
Wireless N Gigabit Router
TL-WR1043ND
QSS


NETFLIX

NETFLIX

89%

Buffering

Full Screen

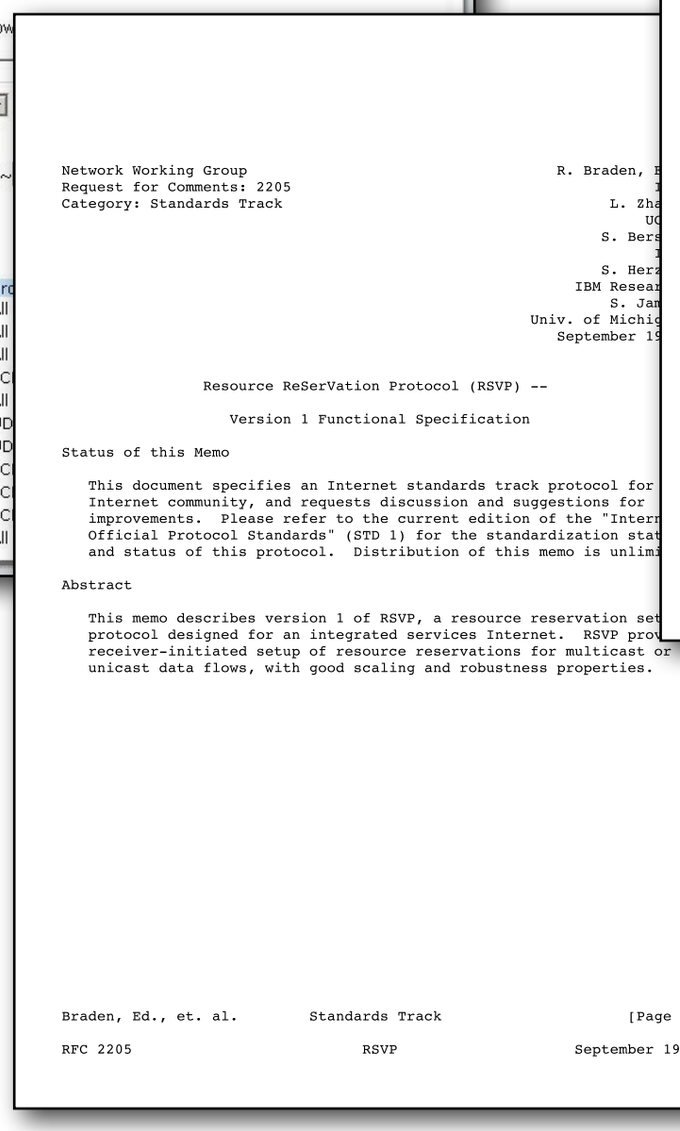
More Episodes

Back to Browsing


YouTube
facebook


PWR SYS WLAN 4 3 2 1 WAN USB QSS
Wireless N Gigabit Router
TL-WR1043ND
QSS


NETFLIX



Network Working Group
Request for Comments: 2205
Category: Standards Track

R. Braden, E
L. Zha
UC
S. Bers
I
S. Herz
IBM Resear
S. Jam
Univ. of Michig
September 19

Resource ReSerVation Protocol (RSVP) --

Version 1 Functional Specification

Status of this Memo

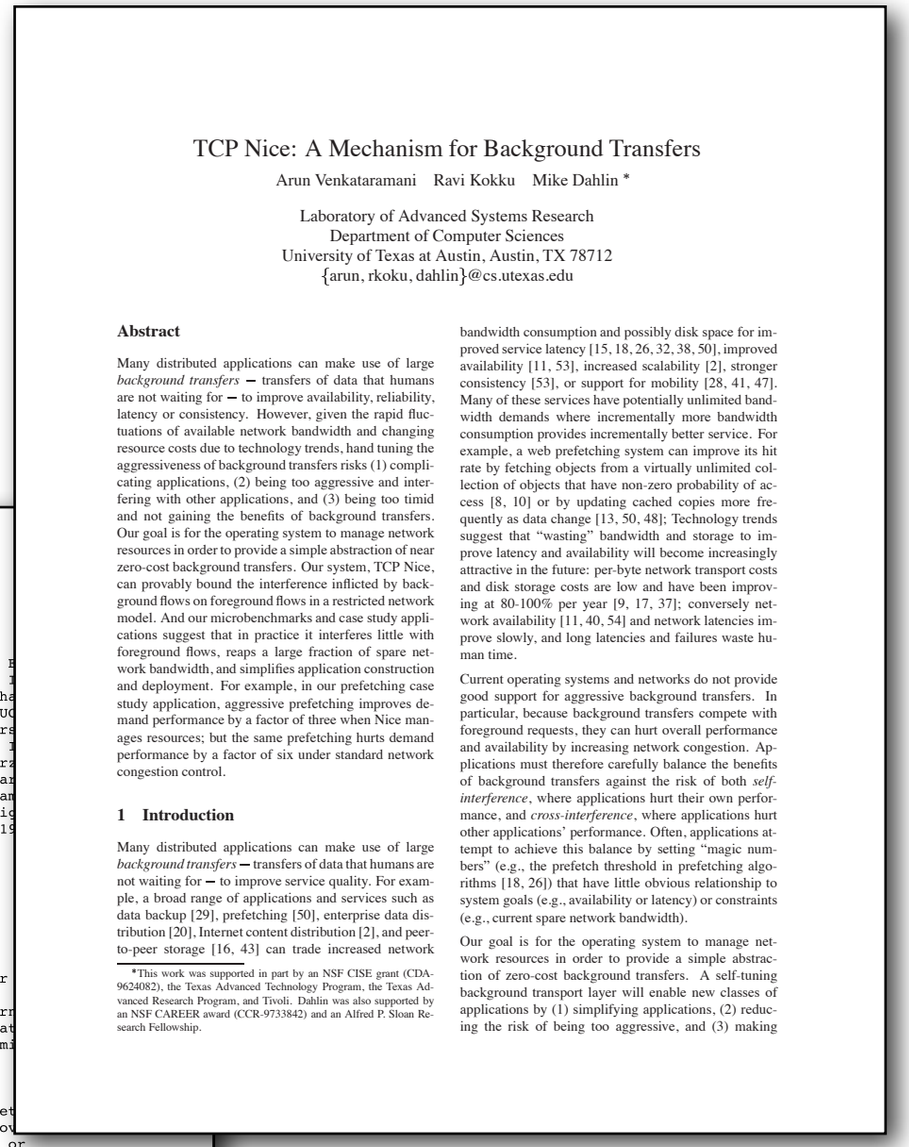
This document specifies an Internet standards track protocol for
Internet community, and requests discussion and suggestions for
improvements. Please refer to the current edition of the "Interr
Official Protocol Standards" (STD 1) for the standardization stat
and status of this protocol. Distribution of this memo is unlimi

Abstract

This memo describes version 1 of RSVP, a resource reservation set
protocol designed for an integrated services Internet. RSVP prov
receiver-initiated setup of resource reservations for multicast or
unicast data flows, with good scaling and robustness properties.

Braden, Ed., et. al. Standards Track [Page 1]

RFC 2205 RSVP September 1997



TCP Nice: A Mechanism for Background Transfers

Arun Venkataramani Ravi Kokku Mike Dahlin *

Laboratory of Advanced Systems Research
Department of Computer Sciences
University of Texas at Austin, Austin, TX 78712
{arun, rkoku, dahlin}@cs.utexas.edu

Abstract

Many distributed applications can make use of large *background transfers* — transfers of data that humans are not waiting for — to improve availability, reliability, latency or consistency. However, given the rapid fluctuations of available network bandwidth and changing resource costs due to technology trends, hand tuning the aggressiveness of background transfers risks (1) complicating applications, (2) being too aggressive and interfering with other applications, and (3) being too timid and not gaining the benefits of background transfers. Our goal is for the operating system to manage network resources in order to provide a simple abstraction of near zero-cost background transfers. Our system, TCP Nice, can provably bound the interference inflicted by background flows on foreground flows in a restricted network model. And our microbenchmarks and case study applications suggest that in practice it interferes little with foreground flows, reaps a large fraction of spare network bandwidth, and simplifies application construction and deployment. For example, in our prefetching case study application, aggressive prefetching improves demand performance by a factor of three when Nice manages resources; but the same prefetching hurts demand performance by a factor of six under standard network congestion control.

1 Introduction

Many distributed applications can make use of large *background transfers* — transfers of data that humans are not waiting for — to improve service quality. For example, a broad range of applications and services such as data backup [29], prefetching [50], enterprise data distribution [20], Internet content distribution [2], and peer-to-peer storage [16, 43] can trade increased network bandwidth consumption and possibly disk space for improved service latency [15, 18, 26, 32, 38, 50], improved availability [11, 53], increased scalability [2], stronger consistency [53], or support for mobility [28, 41, 47]. Many of these services have potentially unlimited bandwidth demands where incrementally more bandwidth consumption provides incrementally better service. For example, a web prefetching system can improve its hit rate by fetching objects from a virtually unlimited collection of objects that have non-zero probability of access [8, 10] or by updating cached copies more frequently as data change [13, 50, 48]; Technology trends suggest that "wasting" bandwidth and storage to improve latency and availability will become increasingly attractive in the future: per-byte network transport costs and disk storage costs are low and have been improving at 80-100% per year [9, 17, 37]; conversely network availability [11, 40, 54] and network latencies improve slowly, and long latencies and failures waste human time.

Current operating systems and networks do not provide good support for aggressive background transfers. In particular, because background transfers compete with foreground requests, they can hurt overall performance and availability by increasing network congestion. Applications must therefore carefully balance the benefits of background transfers against the risk of both *self-interference*, where applications hurt their own performance, and *cross-interference*, where applications hurt other applications' performance. Often, applications attempt to achieve this balance by setting "magic numbers" (e.g., the prefetch threshold in prefetching algorithms [18, 26]) that have little obvious relationship to system goals (e.g., availability or latency) or constraints (e.g., current spare network bandwidth).

Our goal is for the operating system to manage network resources in order to provide a simple abstraction of zero-cost background transfers. A self-tuning background transport layer will enable new classes of applications by (1) simplifying applications, (2) reducing the risk of being too aggressive, and (3) making

*This work was supported in part by an NSF CISE grant (CDA-9624082), the Texas Advanced Technology Program, the Texas Advanced Research Program, and Tivoli. Dahlin was also supported by an NSF CAREER award (CCR-9733842) and an Alfred P. Sloan Research Fellowship.


YouTube
facebook


PWR SYS WLAN 4 3 2 1 WAN USB QSS
Wireless N Gigabit Router
TL-WR1043ND
QSS


NETFLIX

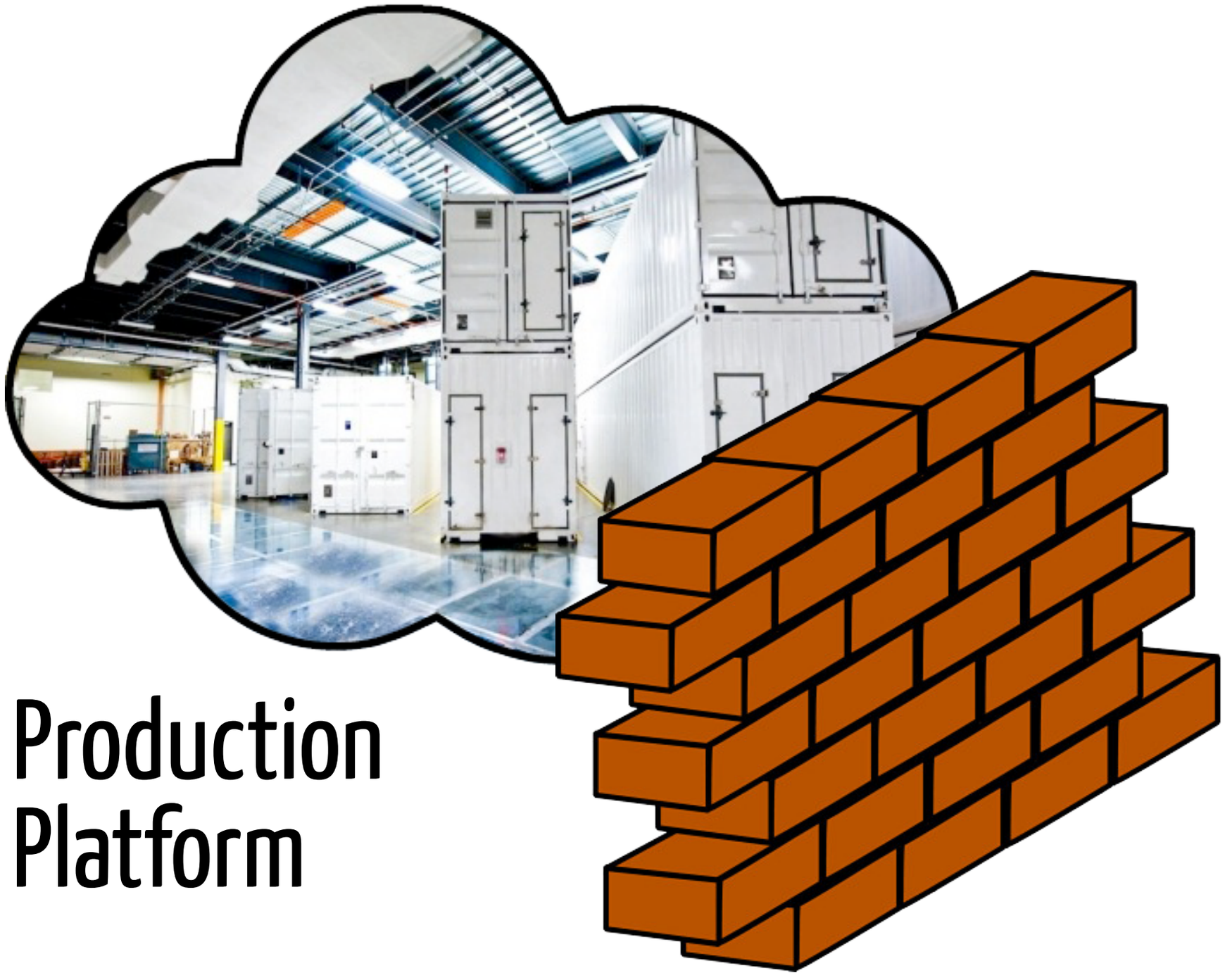

YouTube
facebook


PWR SYS WLAN 4 3 2 1 WAN USB QSS
Wireless N Gigabit Router
TL-WR1043ND
QSS

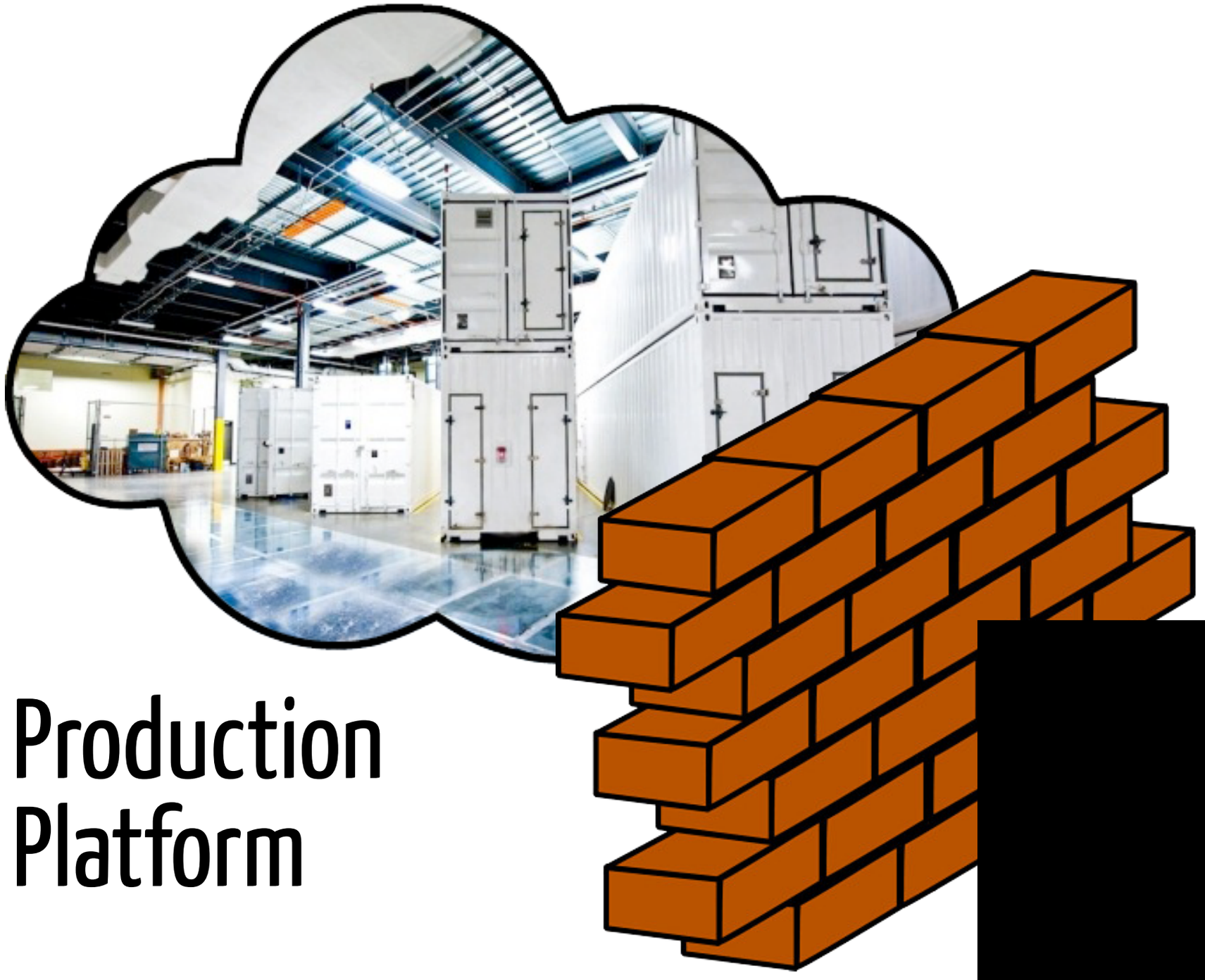

NETFLIX

Datacenter

Production Platform



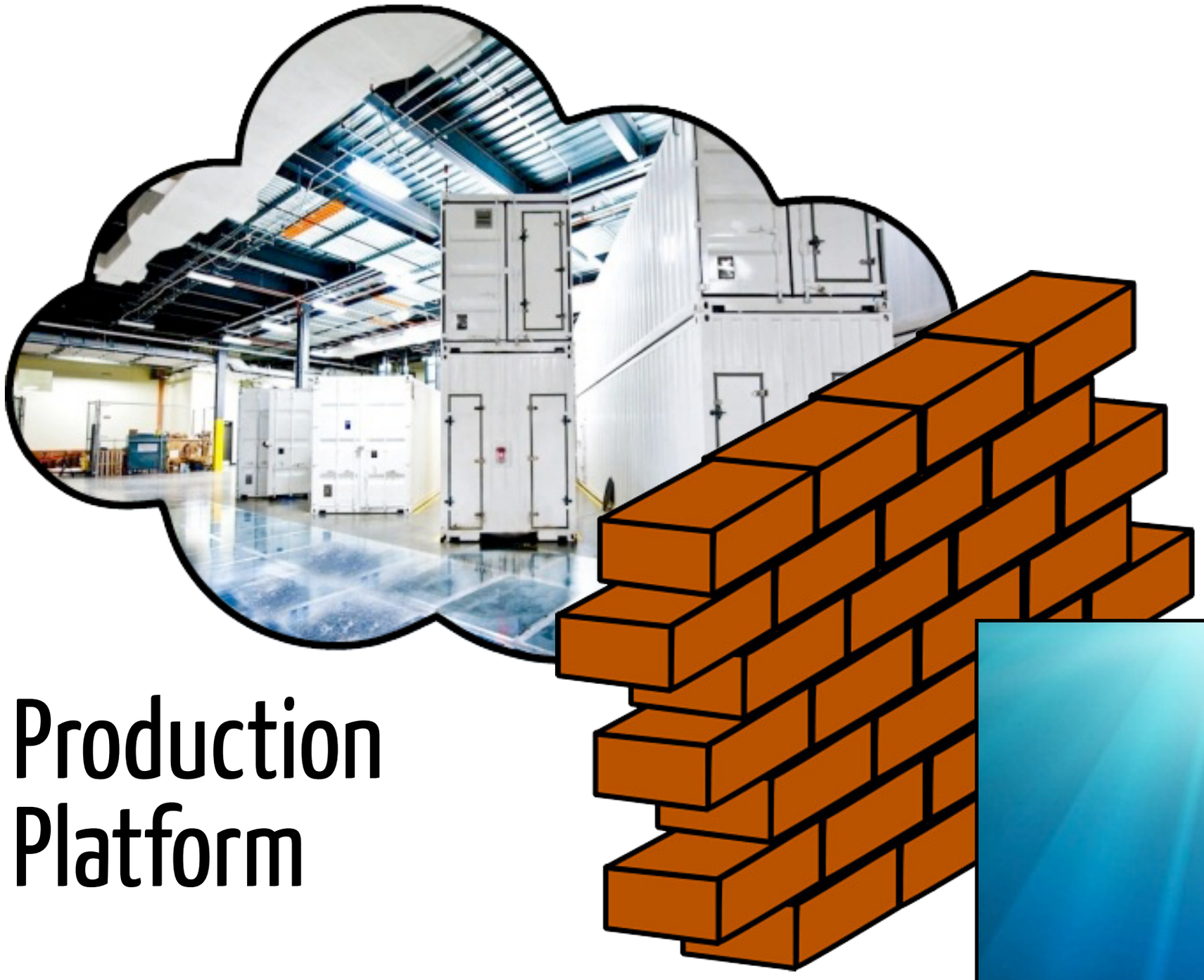
Production Platform



Production Platform

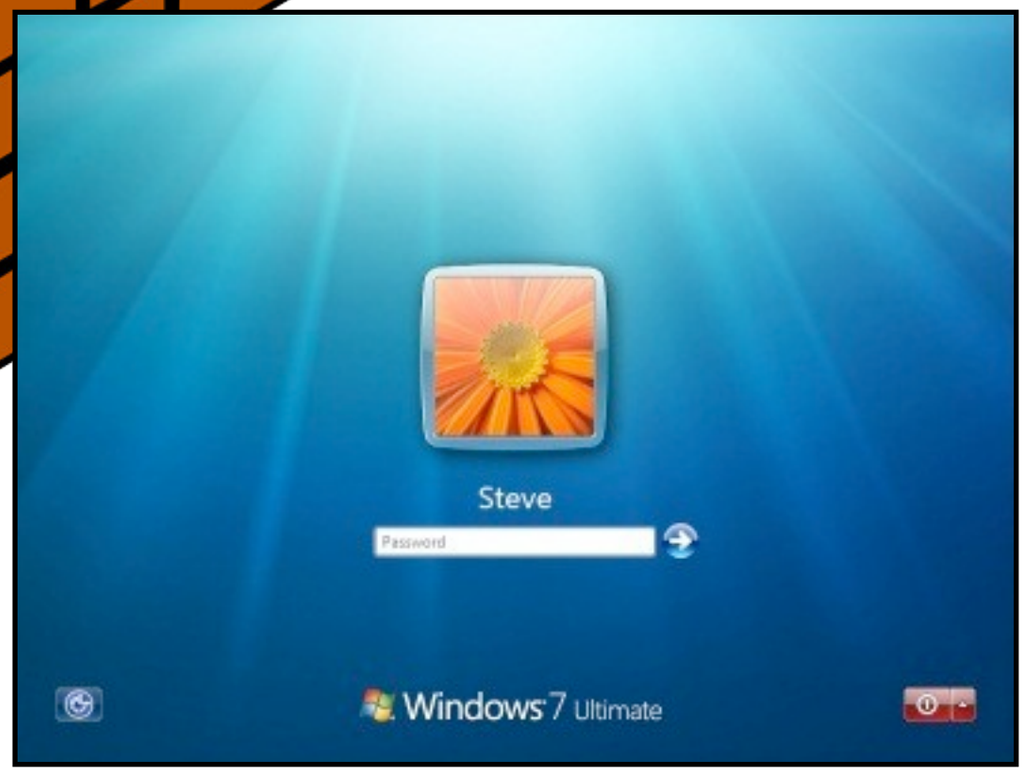
Boot Service

Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)

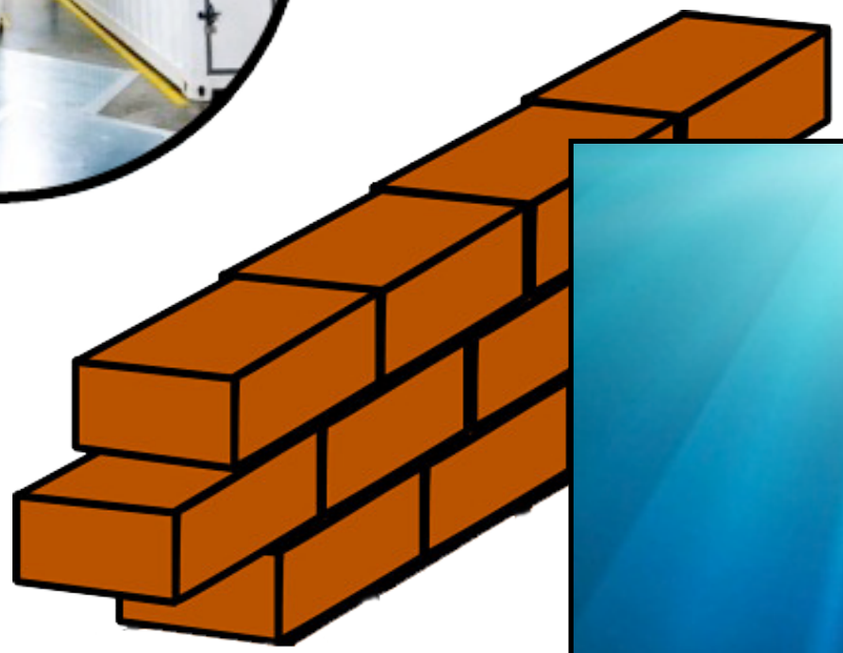


Production Platform

Boot Service



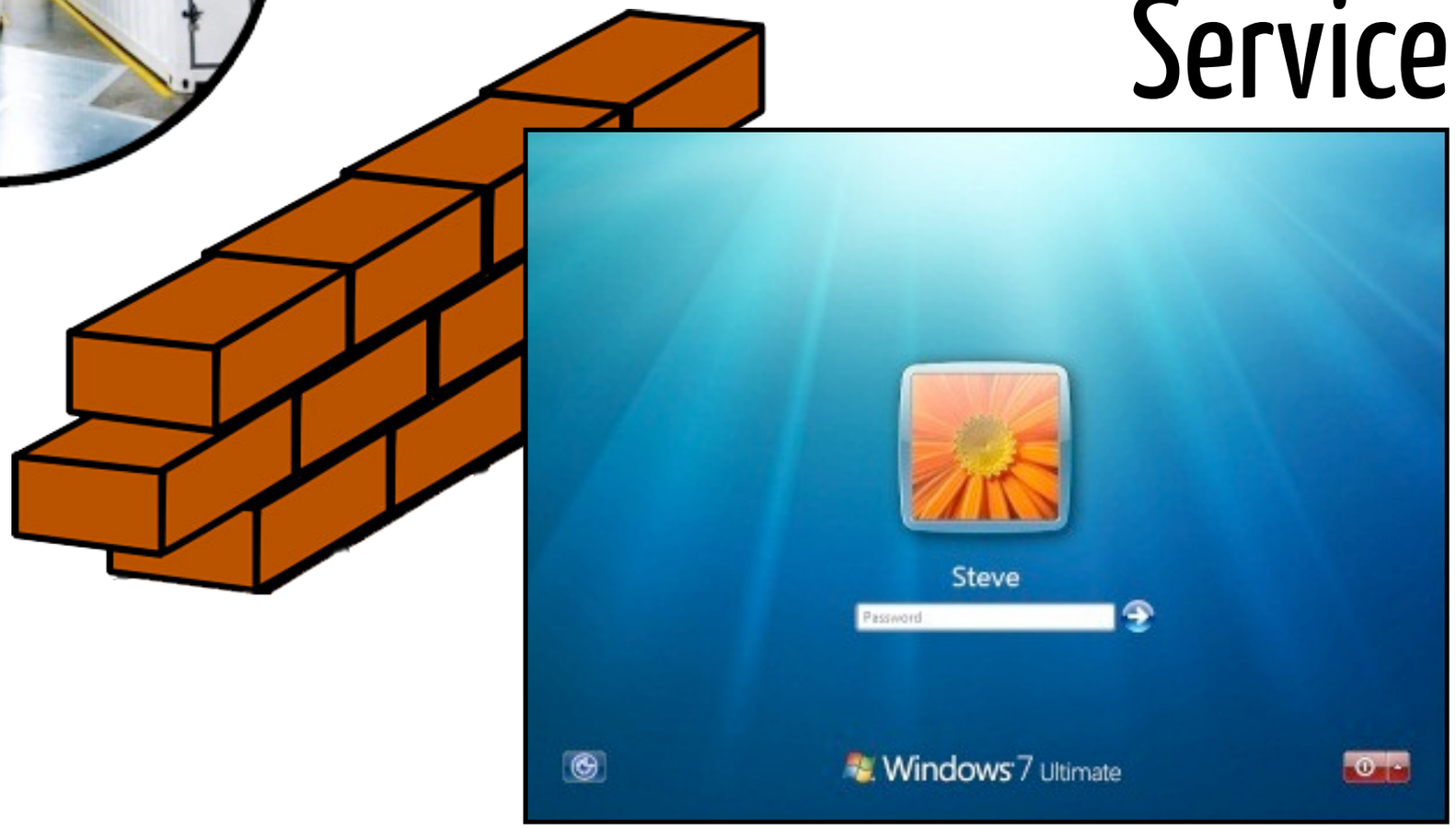
Production Platform



Boot Service

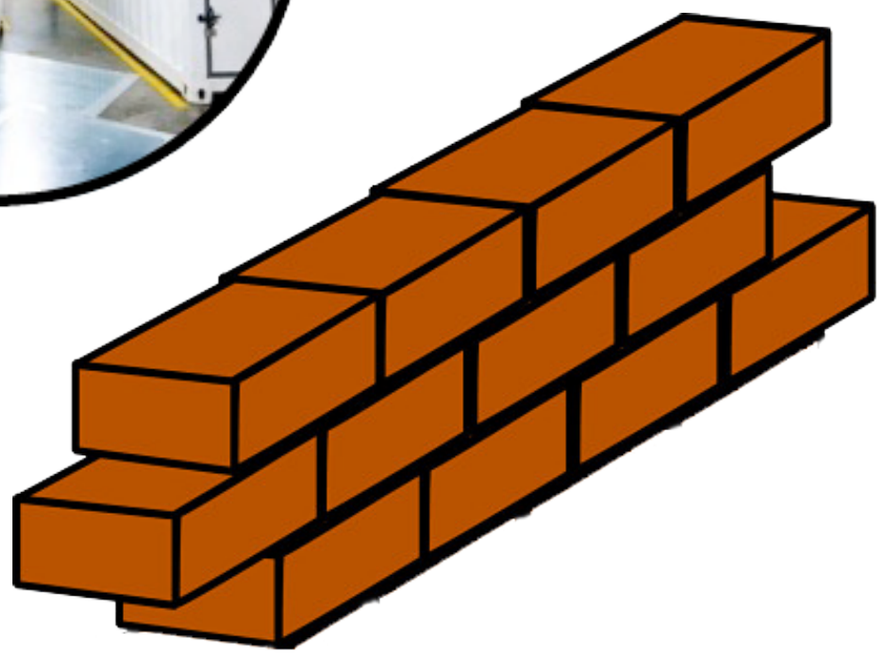
Based on “Delusional Boot: Securing Cloud Hypervisors without Massive Re-Engineering” (EuroSys 2012)

Production Platform

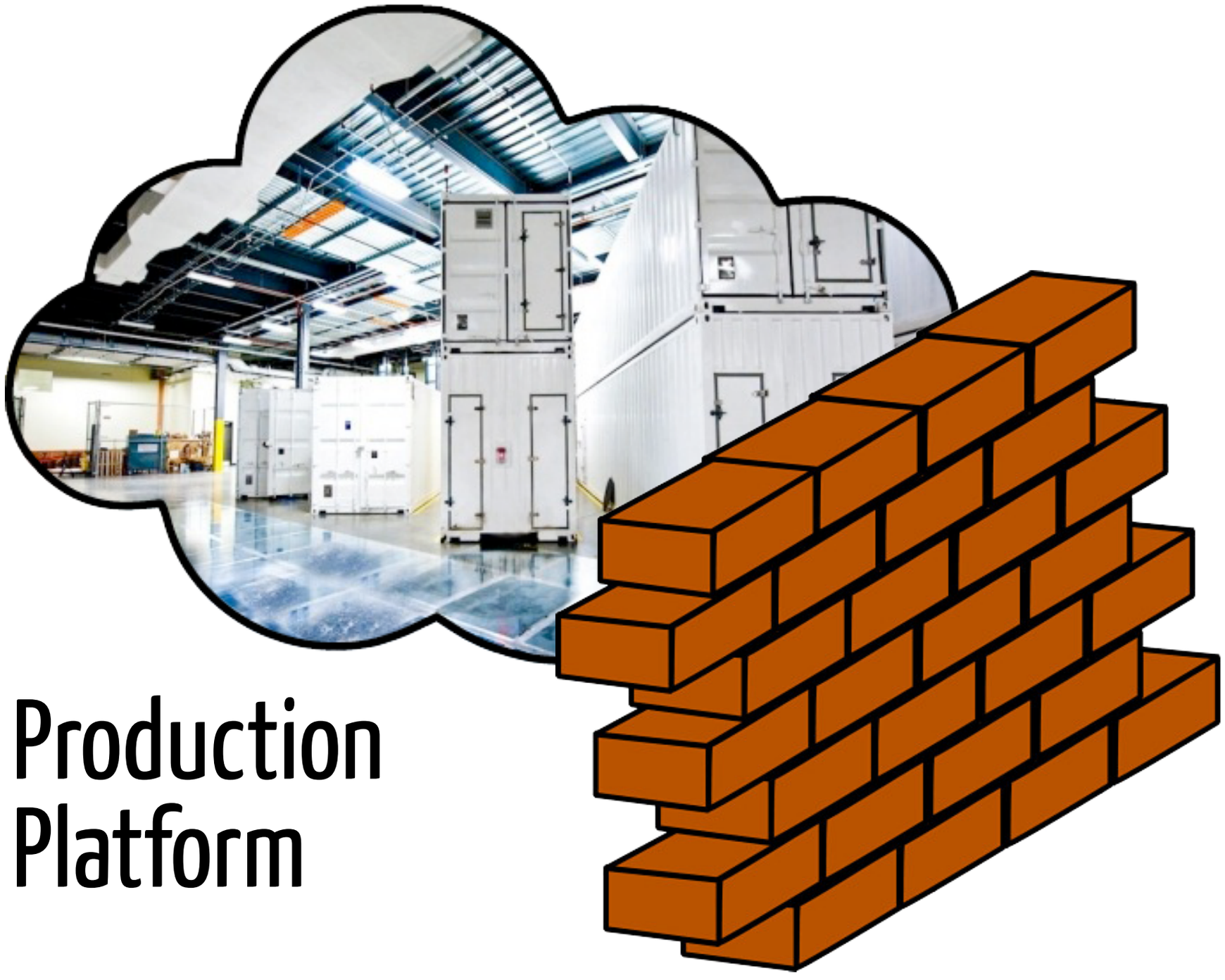


Boot Service

Production Platform



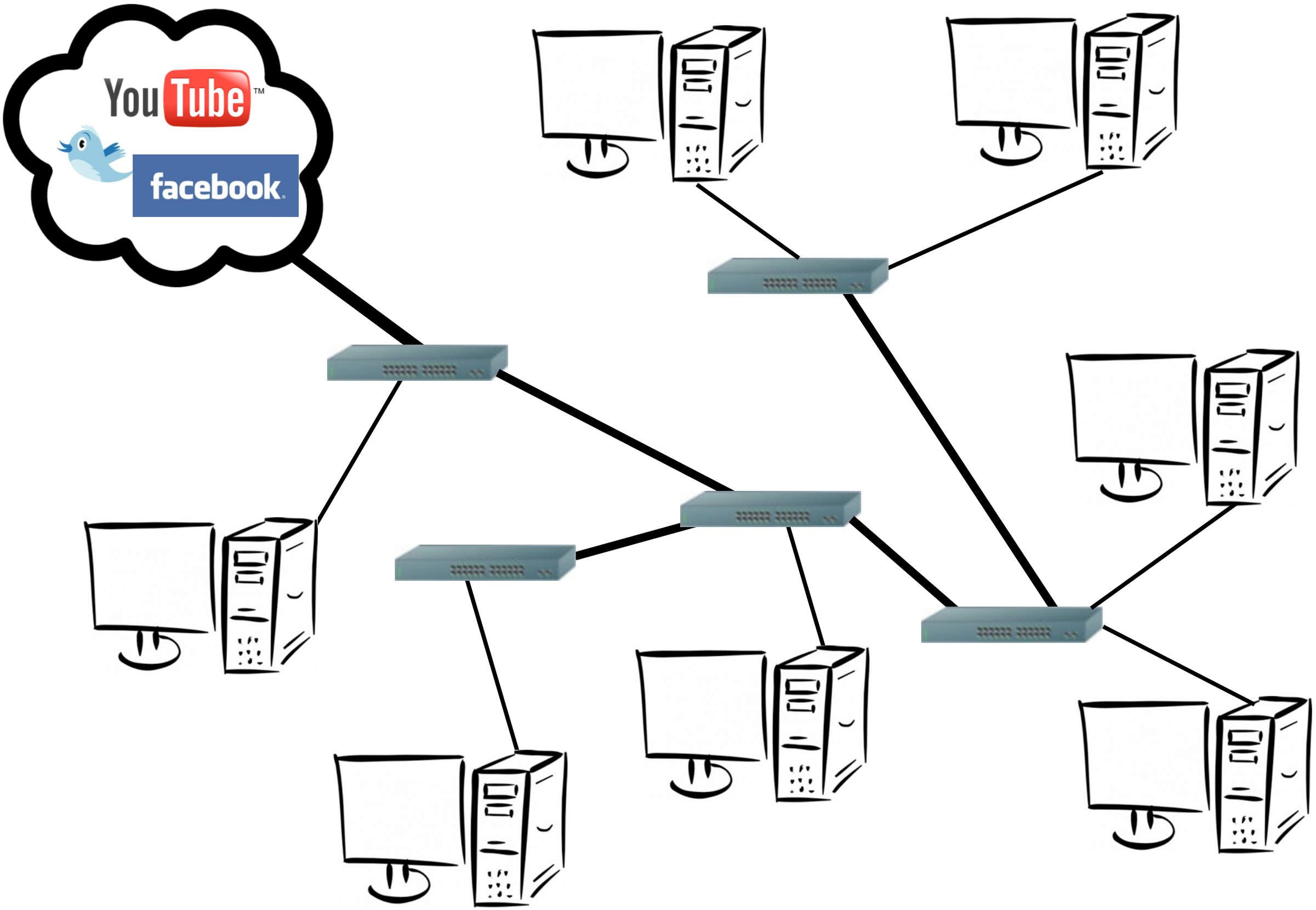
Boot Service

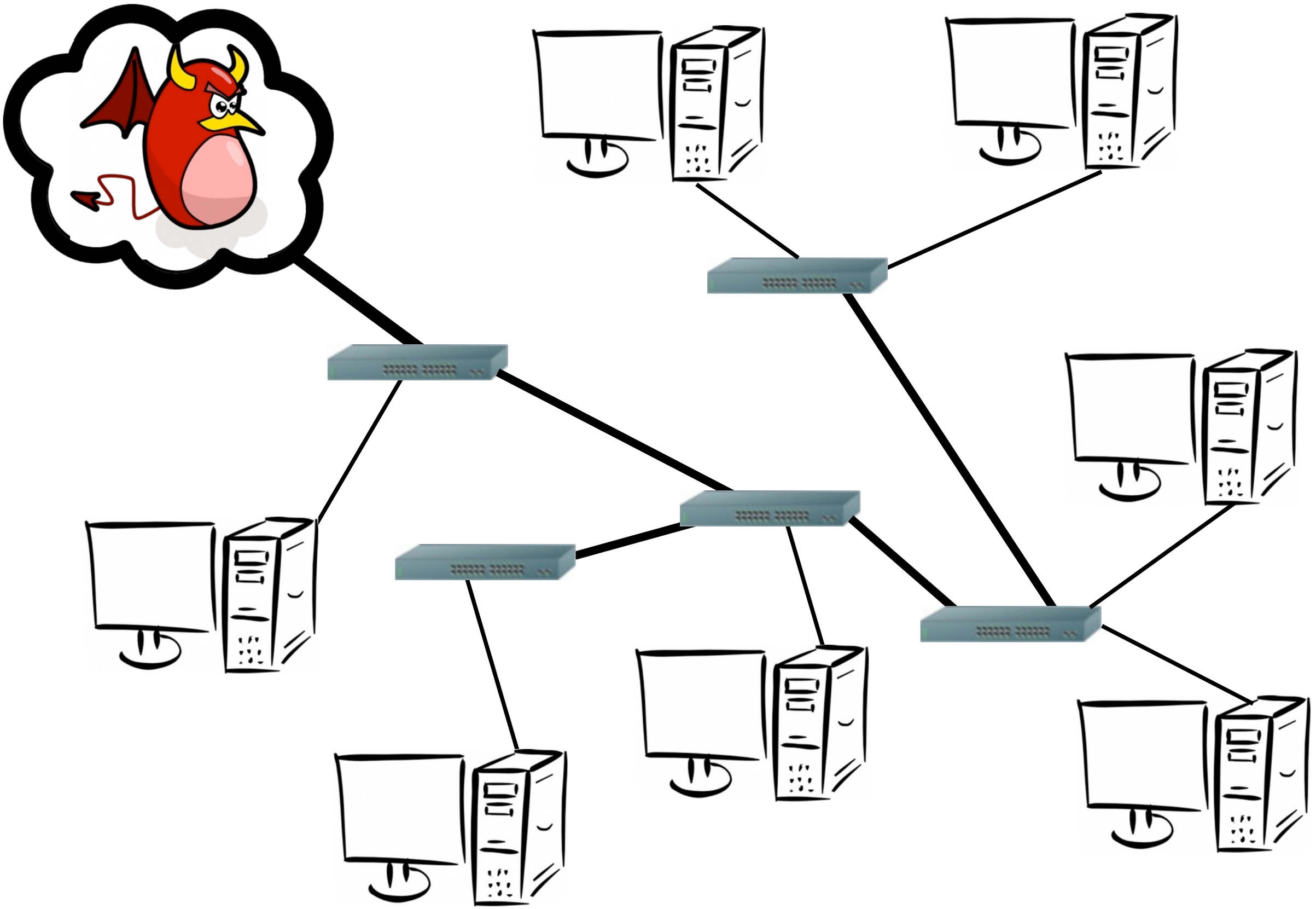


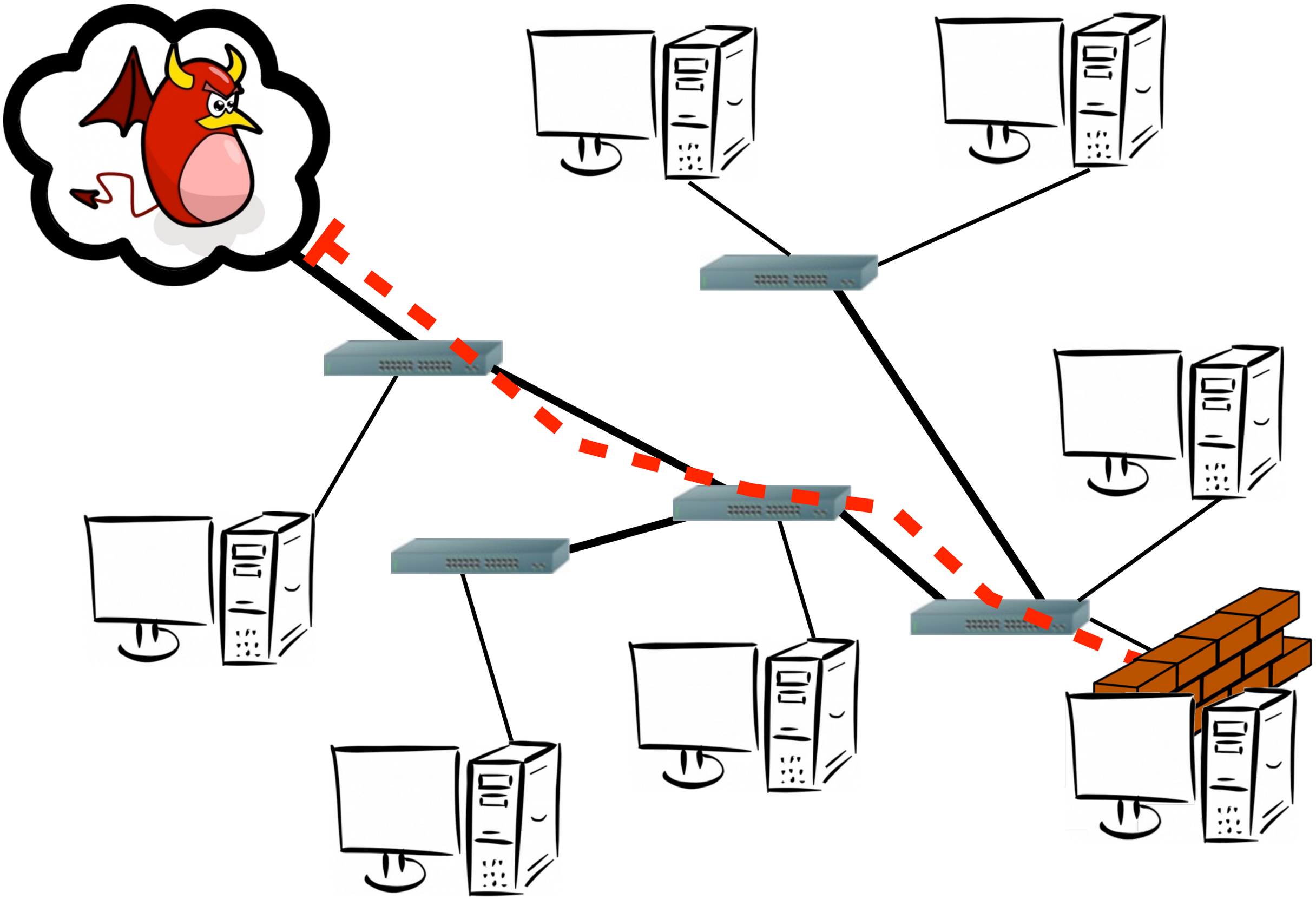
Production Platform

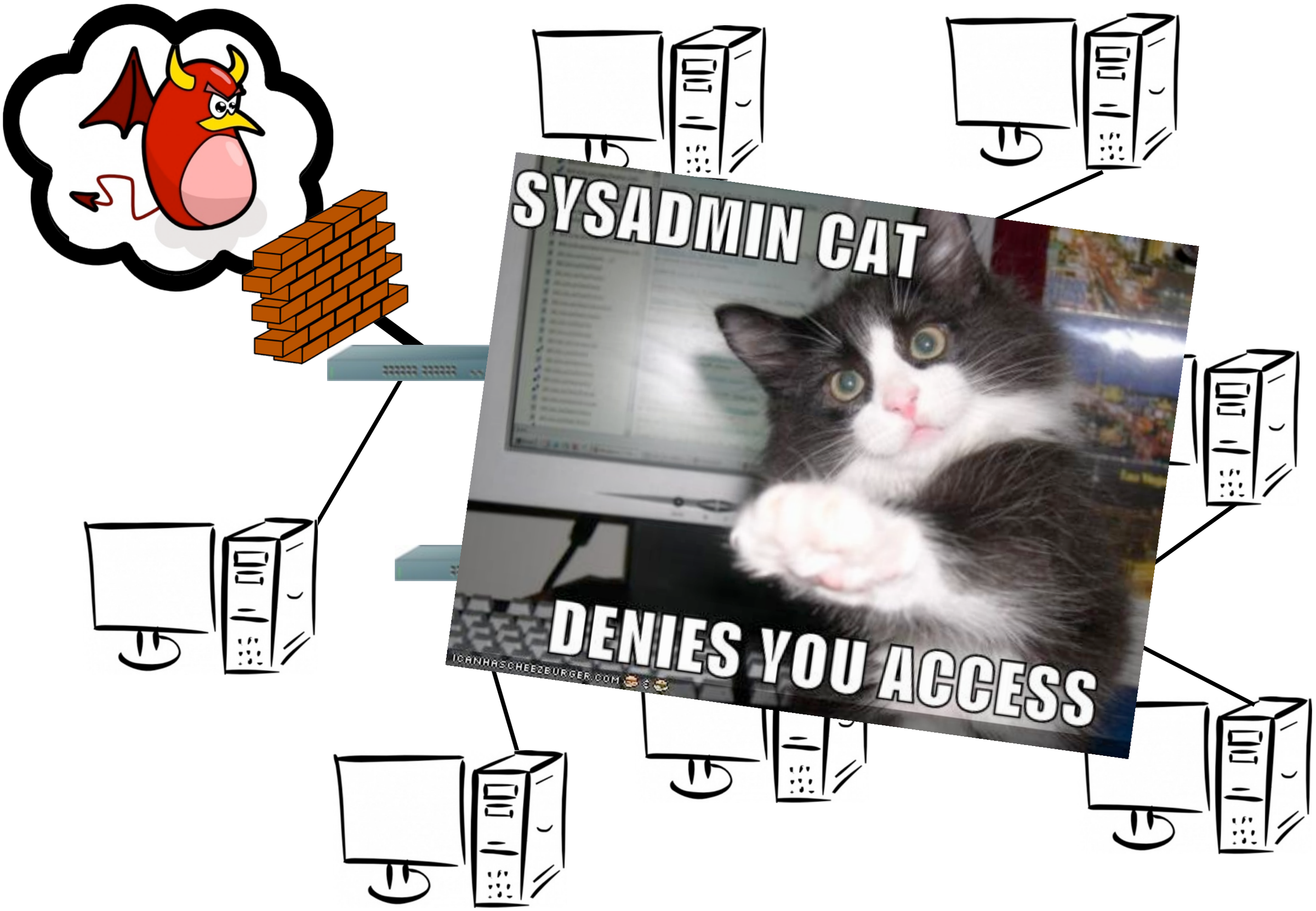
Boot Service

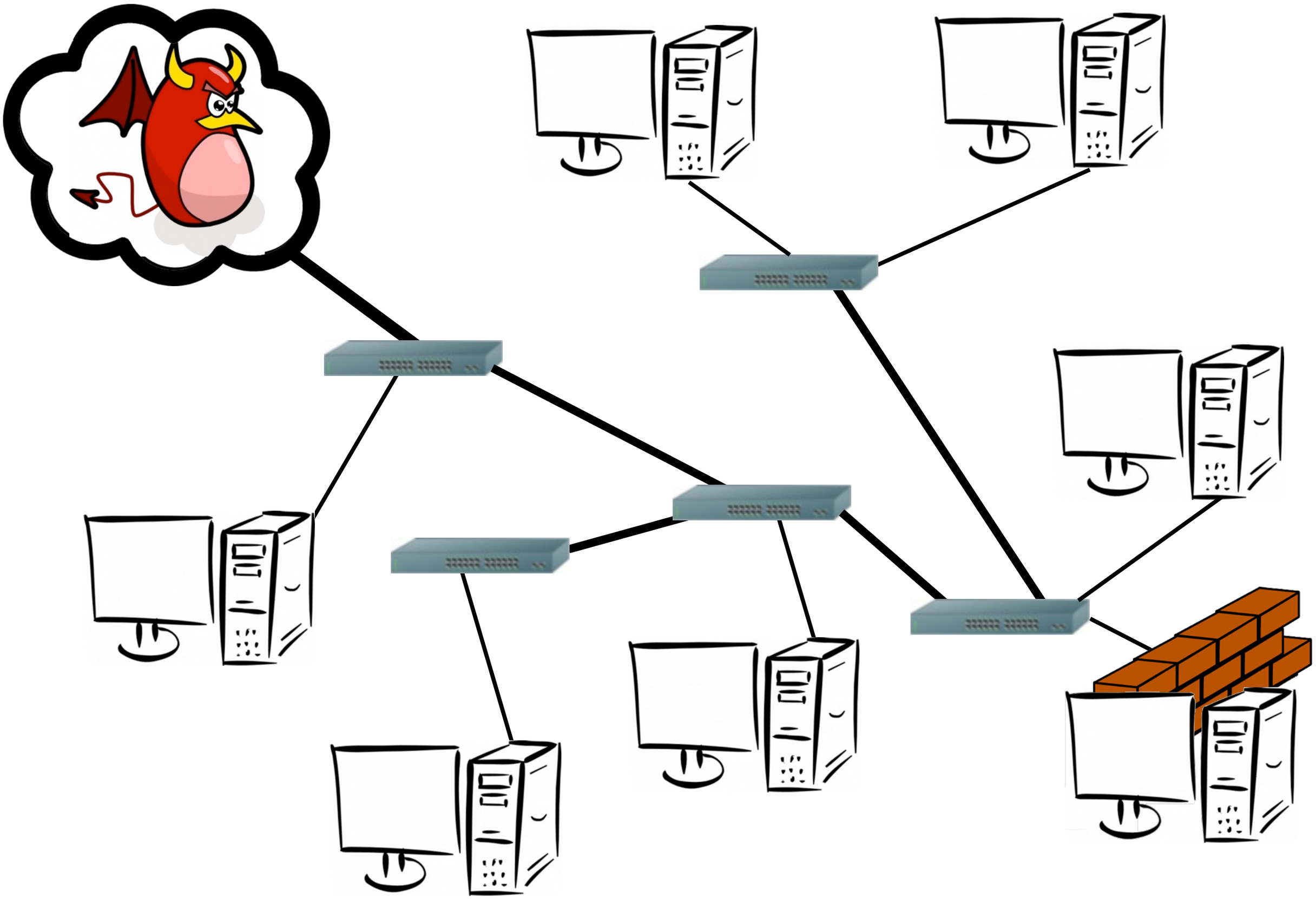
Enterprise

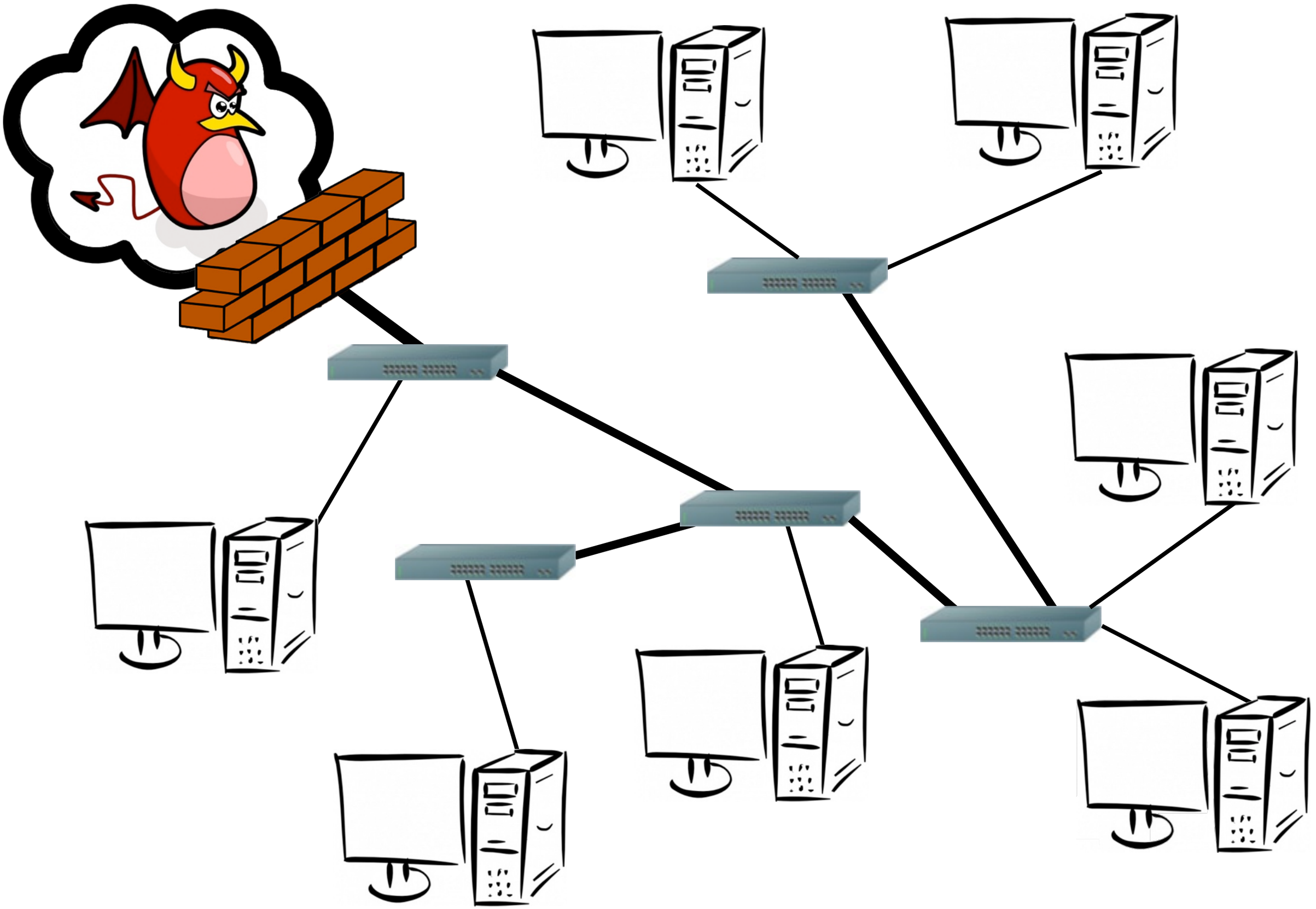

YouTube
facebook



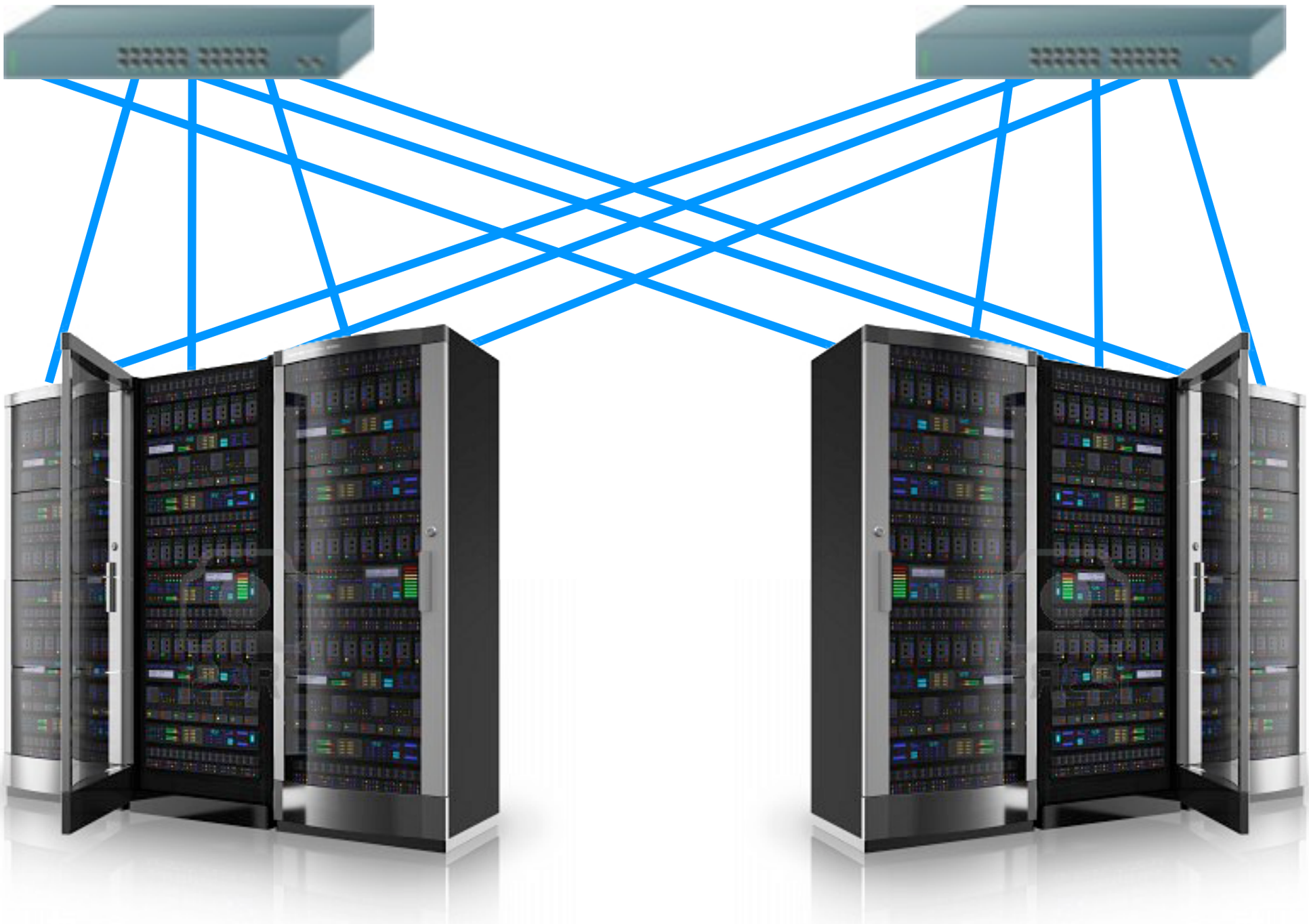


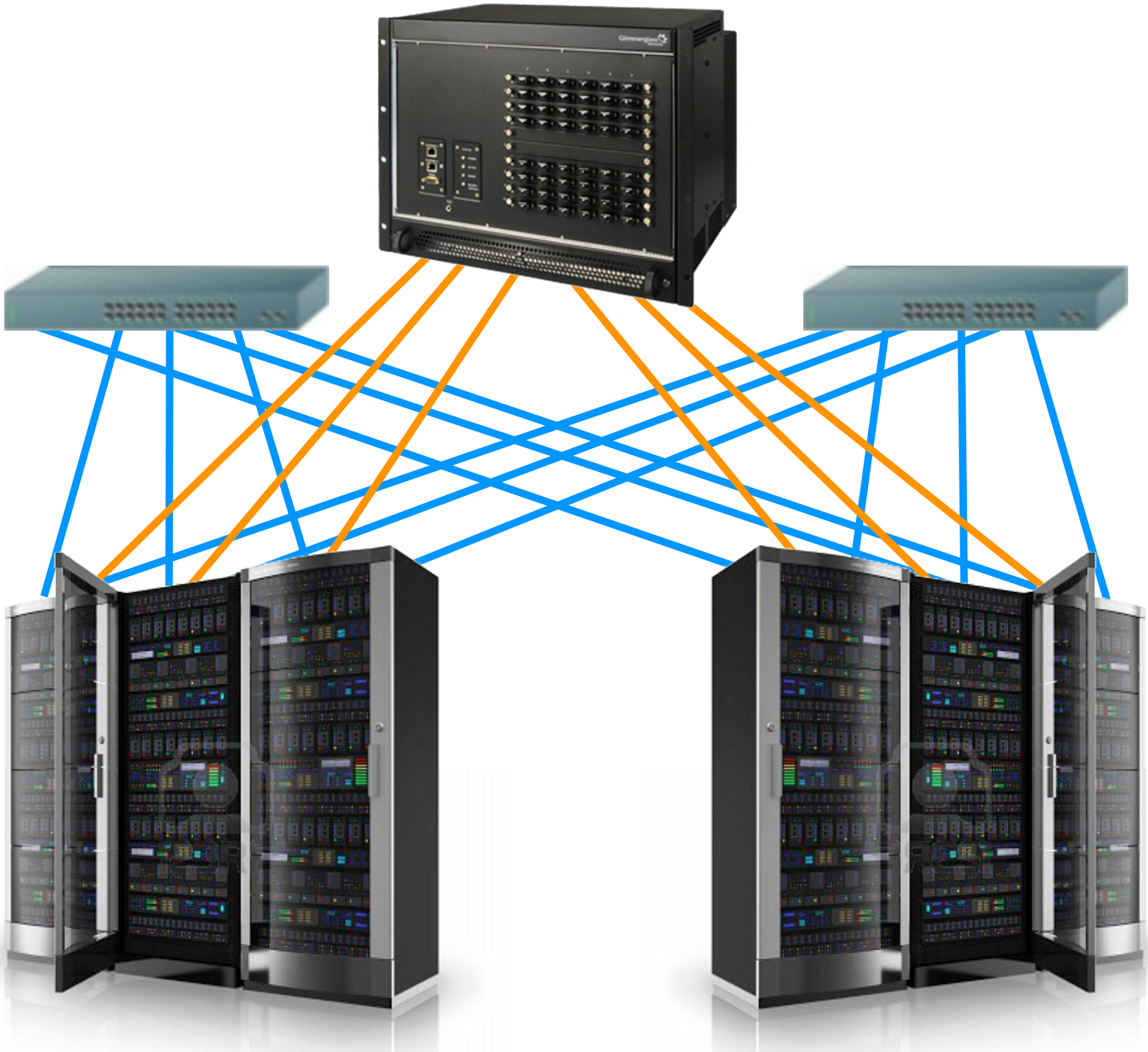

SYSADMIN CAT
DENIES YOU ACCESS
ICANHASCHEEZBURGER.COM

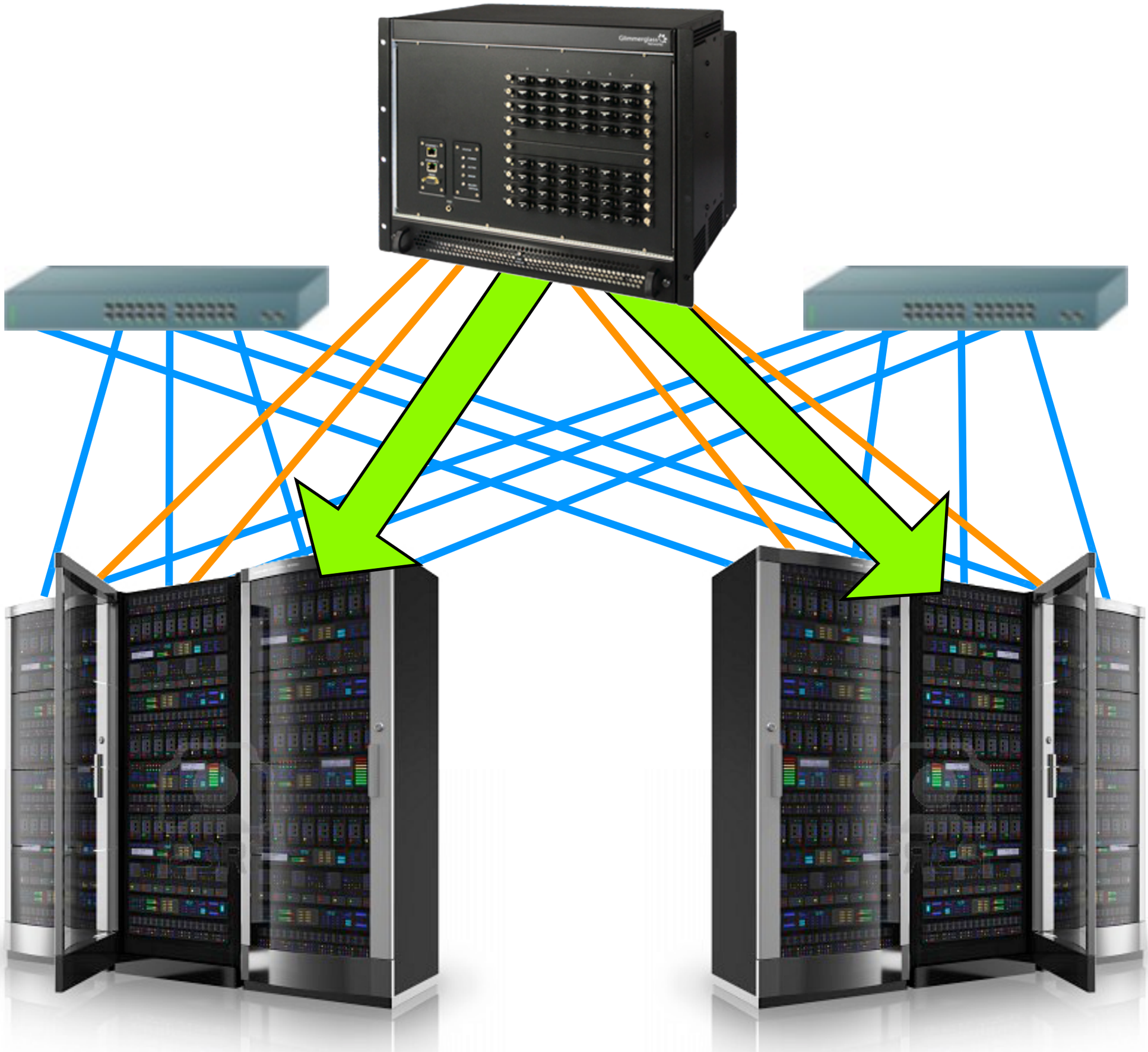


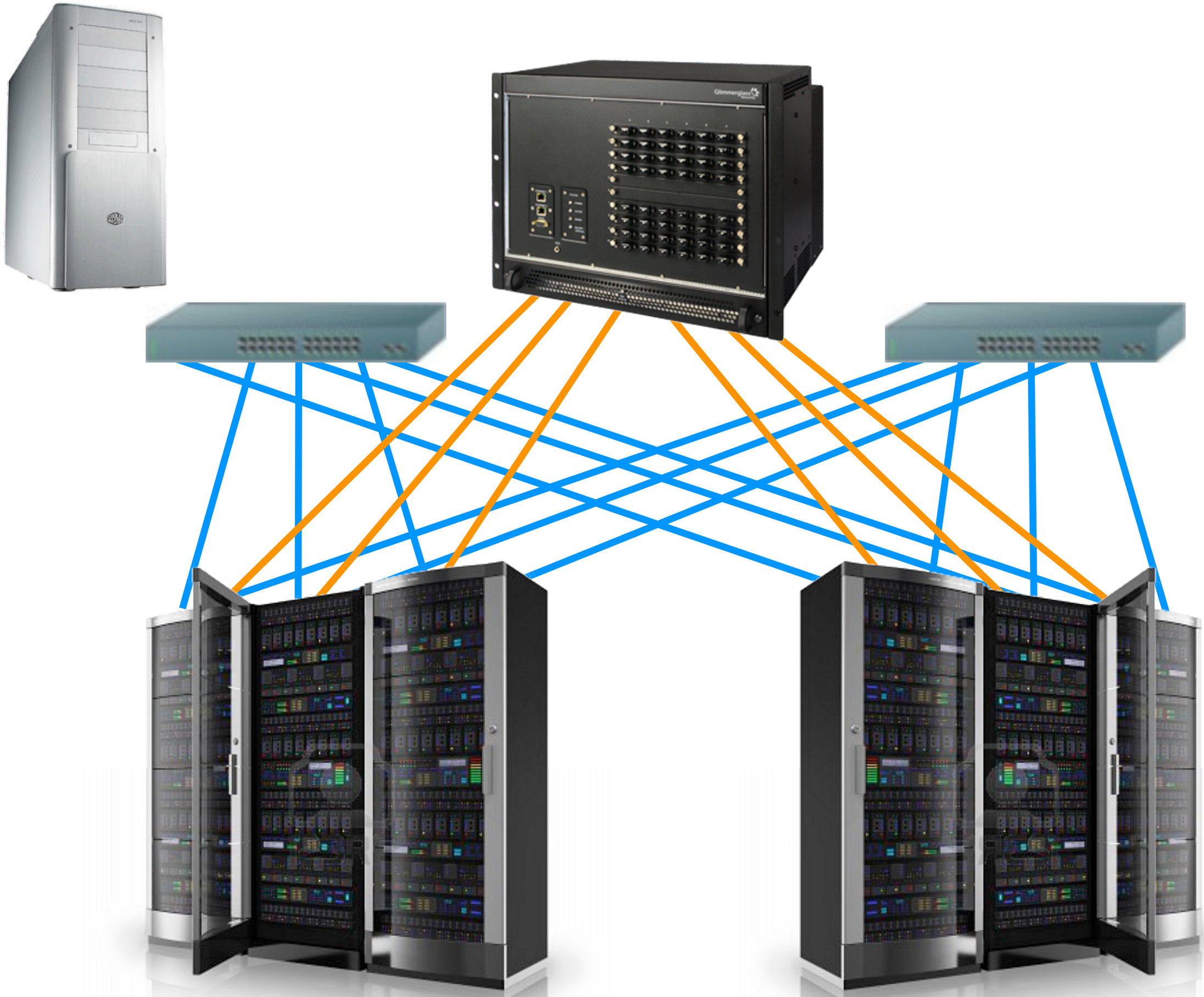


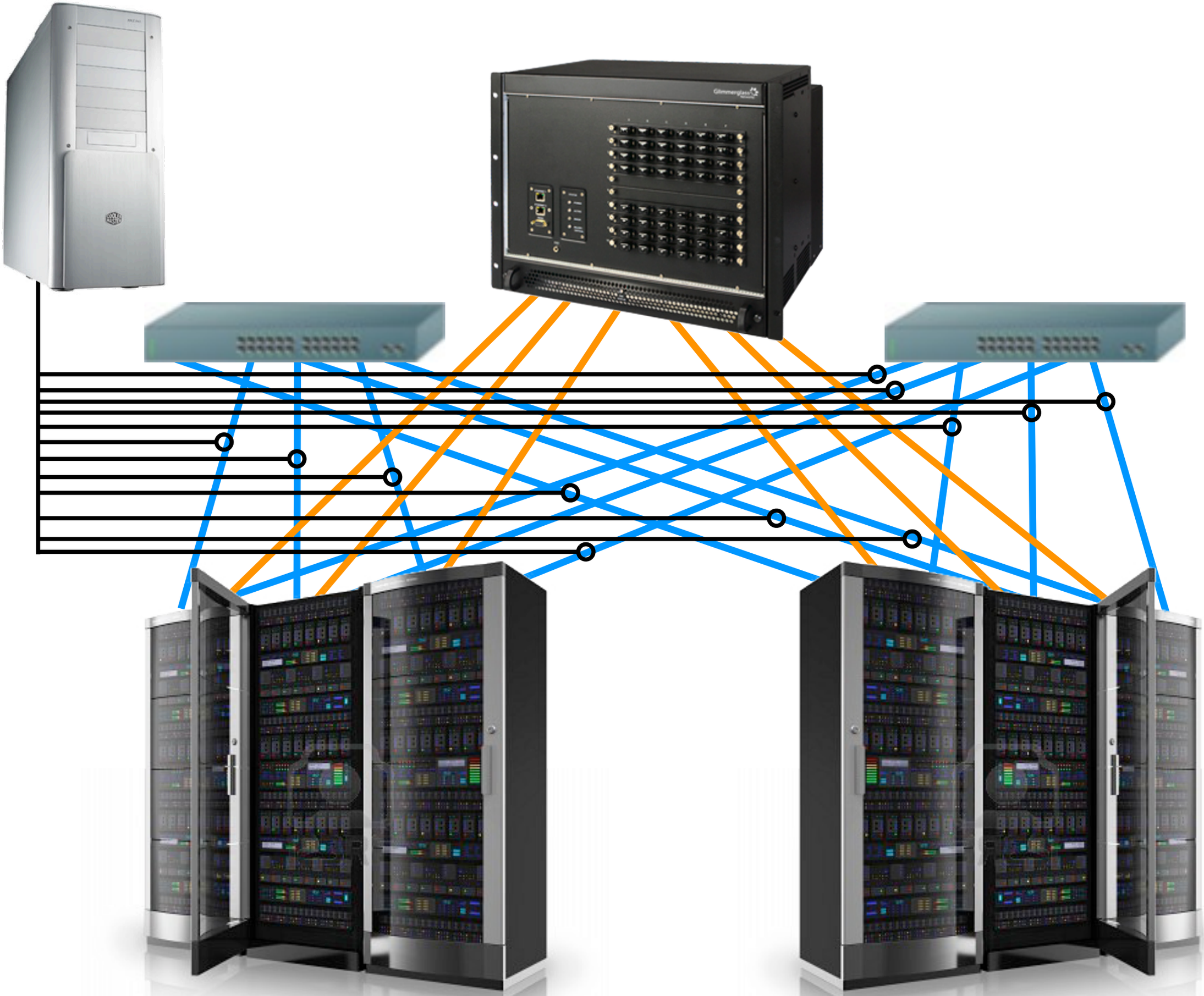
A problem in the datacenter

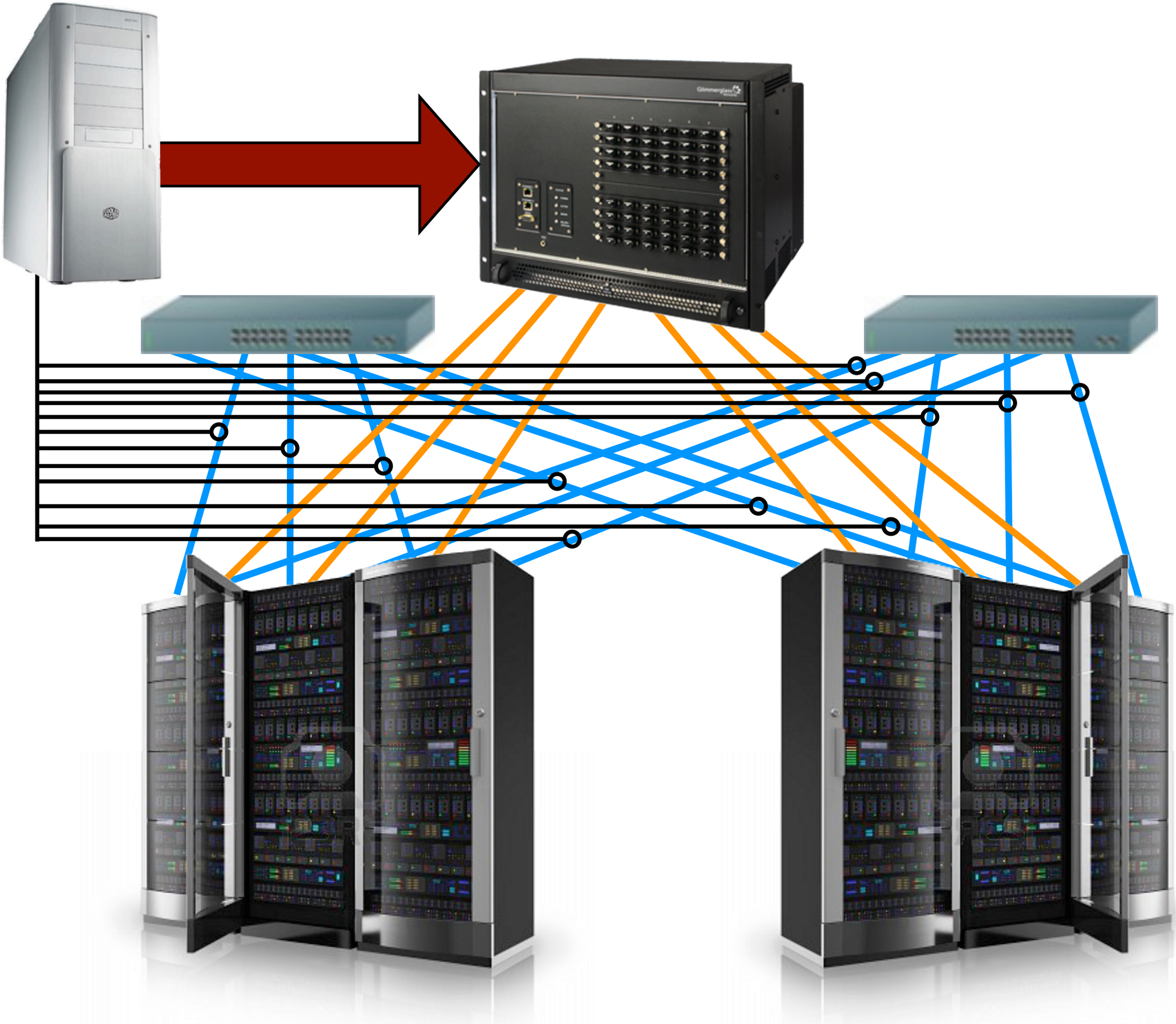






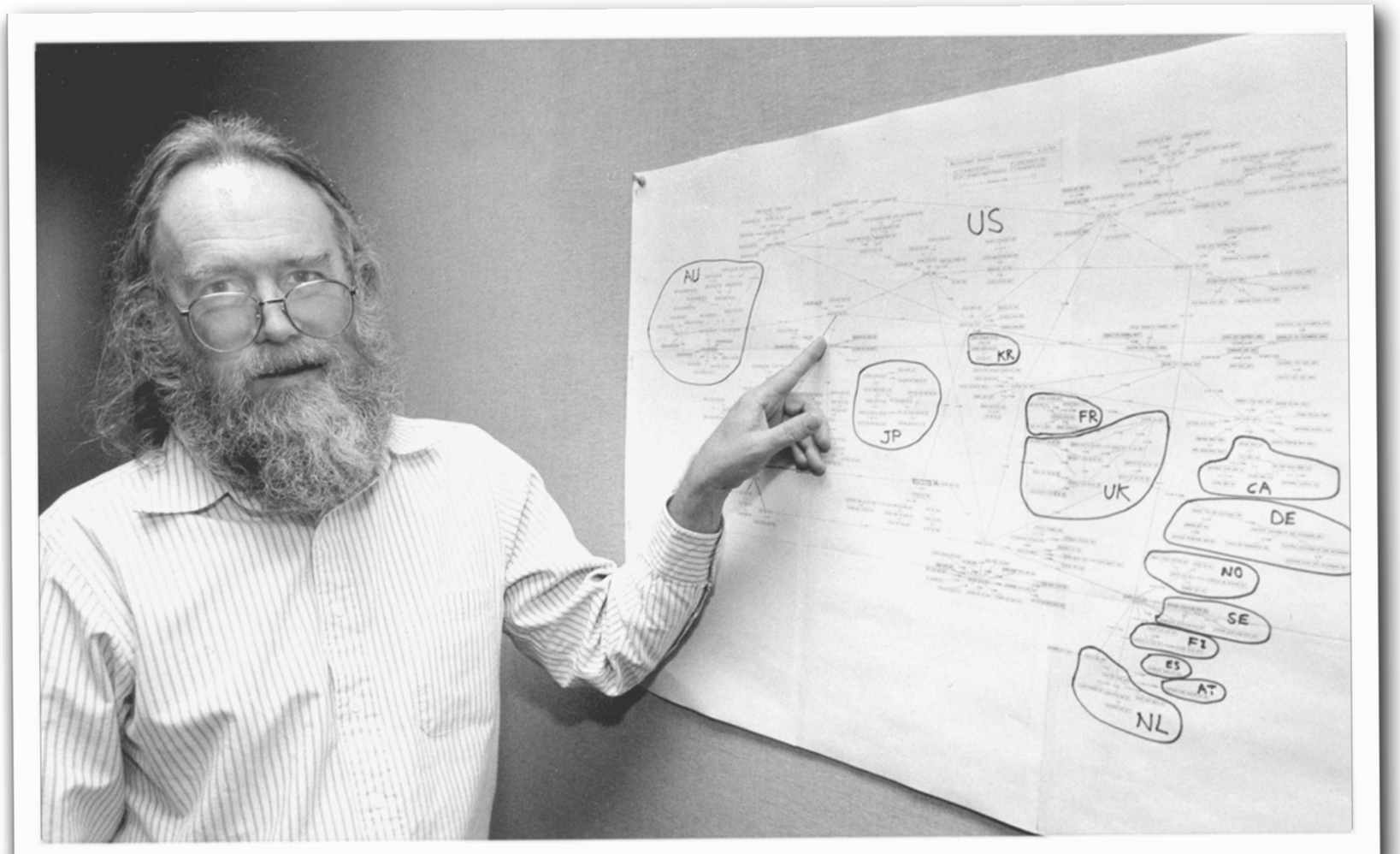






Participatory Networking

Ken Thompson & Dennis Ritchie



Jon Postel

OCCUPY EVERYTHING

#OCCUPYWALLST

WE ALREADY KNOW THAT WE OWN EVERYTHING--THE TASK IS TO EXCLUDE THE INTRUSIONS OF CAPITAL AND POWER

BANKS BAILED

Participatory Networking

Safe?

Secure?

Fair?

Loop freedom?

Black holes?