Institut für Kommunikations-Technik

Middlebox Configuration

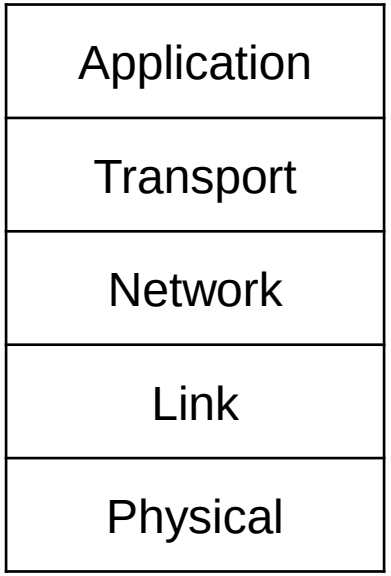
Network Management

Prof. Dr. Panagiotis Papadimitriou

Leibniz Universität Hannover

Proliferation of Network Devices

- “Middleboxes” (L3-L5)
  - Network address translation (NAT)
  - Load balancing
  - Encryption
  - Intrusion detection
  - Access control lists (ACL)
  - Firewall
  - …..
- Routers (L3)
- Bridges, switches (L2)
- Repeaters, hubs (L1)



Network Address Translation (NAT)

Network Address Translation (NAT)

- NAT allows local networks to use just one public IP address:
  - ISP assigns a single IP address for the whole network
  - Devices use private IP addresses and are not explicitly addressable (security)
    - Device IP addresses can be changed without notifying outside world
  - NAT carries out the translation between the public IP address and the private IP addresses

- NAT reduces the utilization of IPv4 address space
  - IPv4 address space was depleted in 2011
  - IPv6 deployment is slow

NAT Translation Table

WAN side	LAN side
Public IP, Port	Private IP, Port
….	….

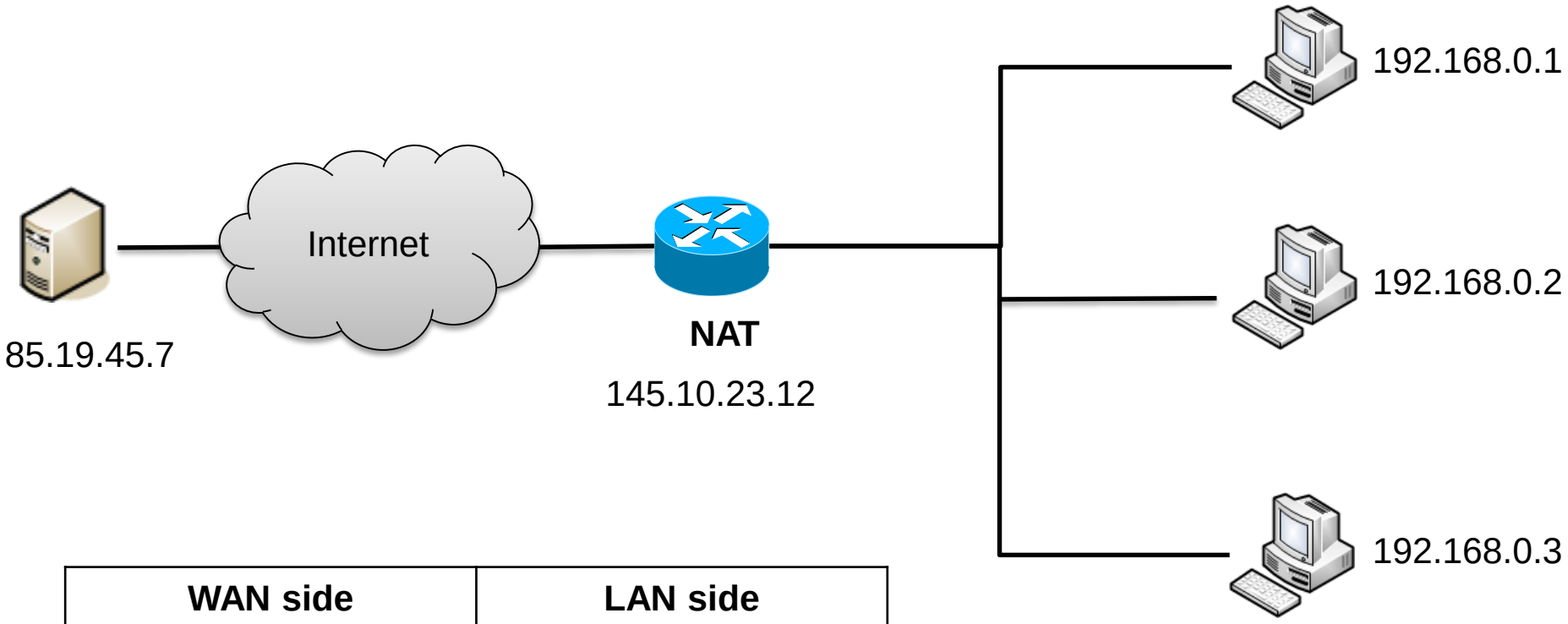
- NAT maintains a table to carry out address translation:
  - WAN side:
    - IP address is assigned by ISP
    - Port number is arbitrarily chosen by the NAT and addresses a host
  - LAN side:
    - IP address for each device is typically assigned by the LAN DHCP server
    - Port number addresses each network application running in a host

NAT Implementation

- LAN → WAN:
  - Assign a new port number for every new connection and add a new entry to the NAT table with:
    - public IP address, new port number (WAN side)
    - private IP address, port number (LAN side)
  - Replace the source IP address and source port number of every outgoing IP packet with the public IP address and the assigned port number

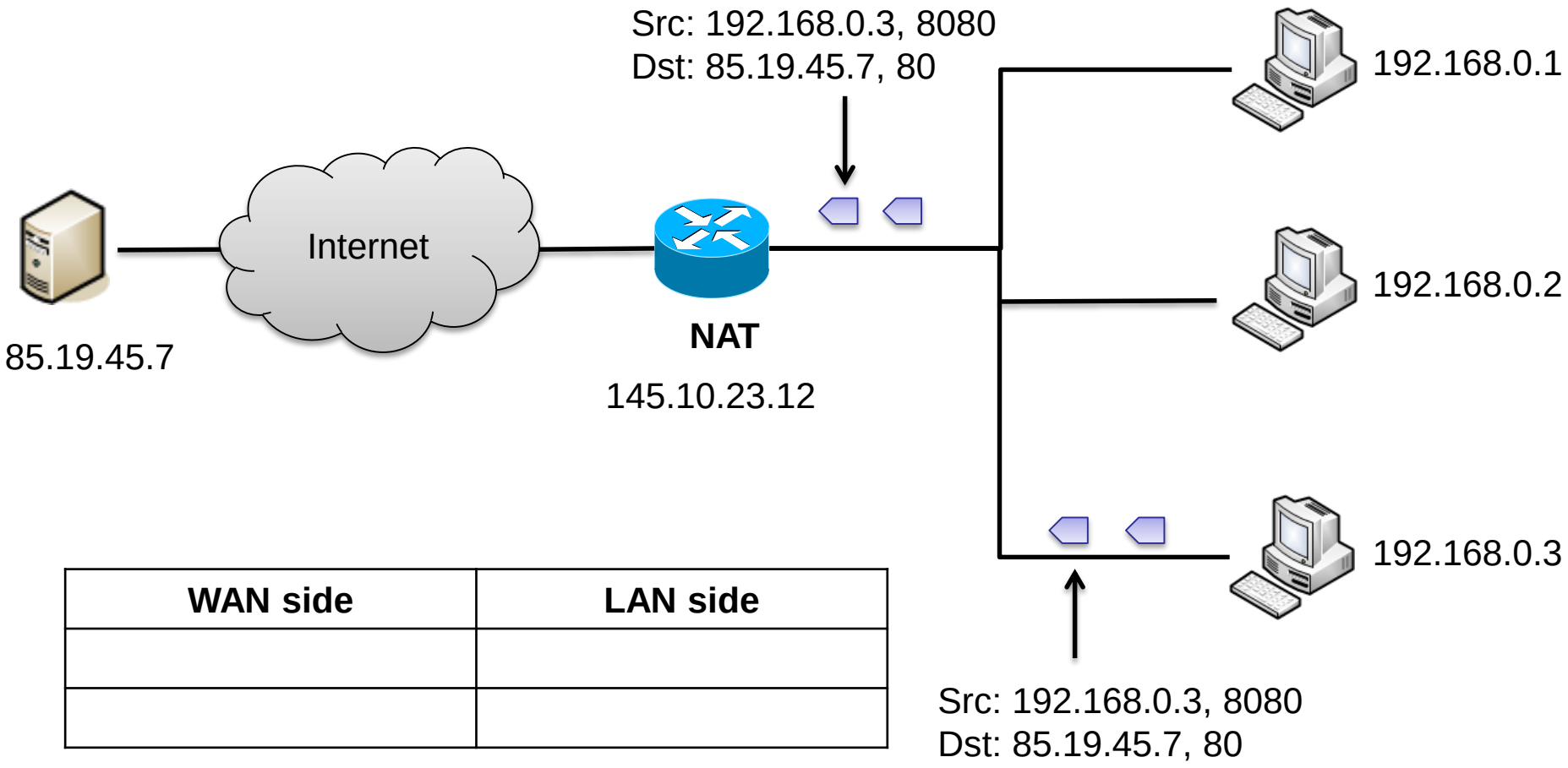
- WAN → LAN:
  - Look up the destination port number in the WAN side of the NAT table
  - Replace destination IP address and destination port number of every incoming IP packet with the corresponding private IP address and port number in the LAN side of the NAT table

NAT Example



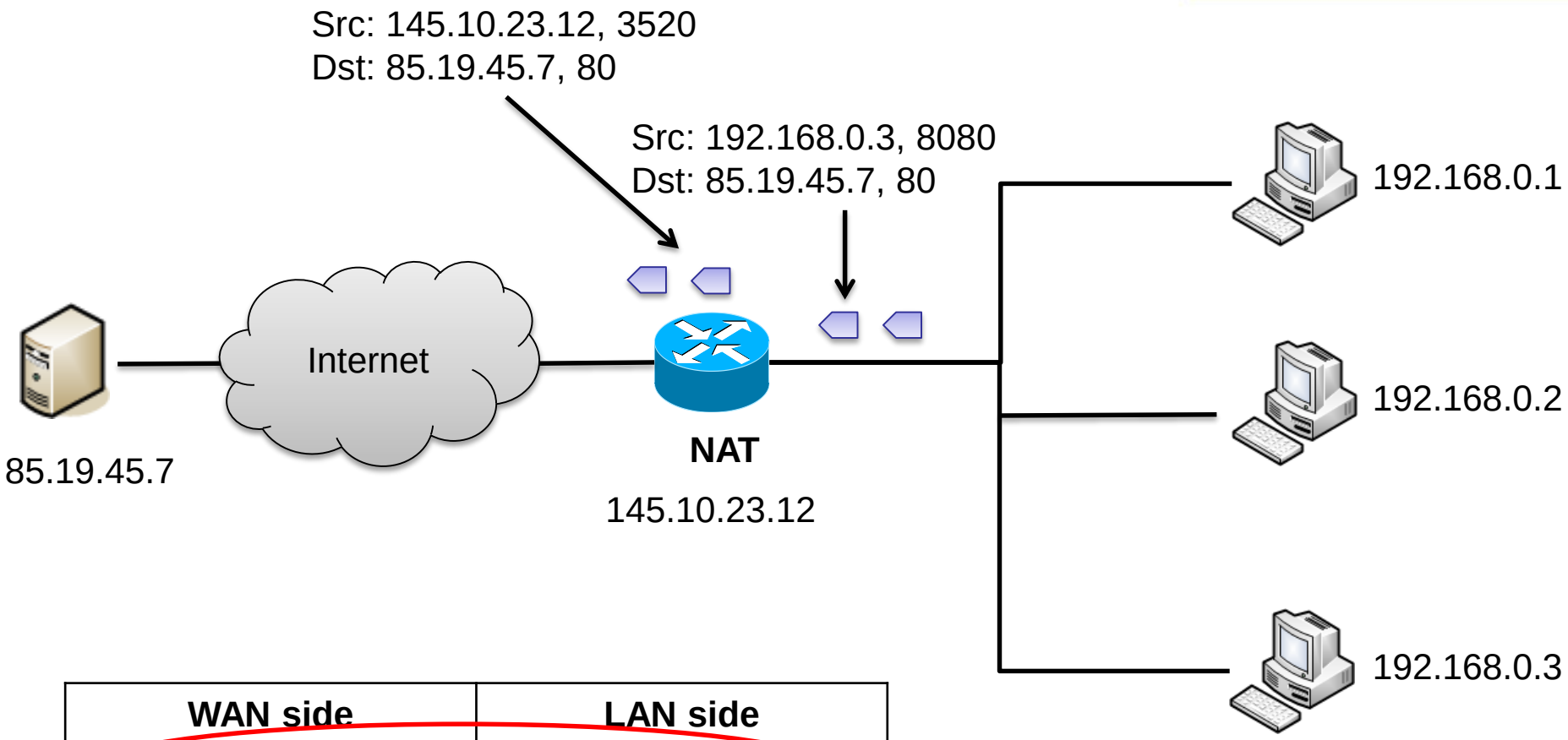
WAN side	LAN side

NAT Example



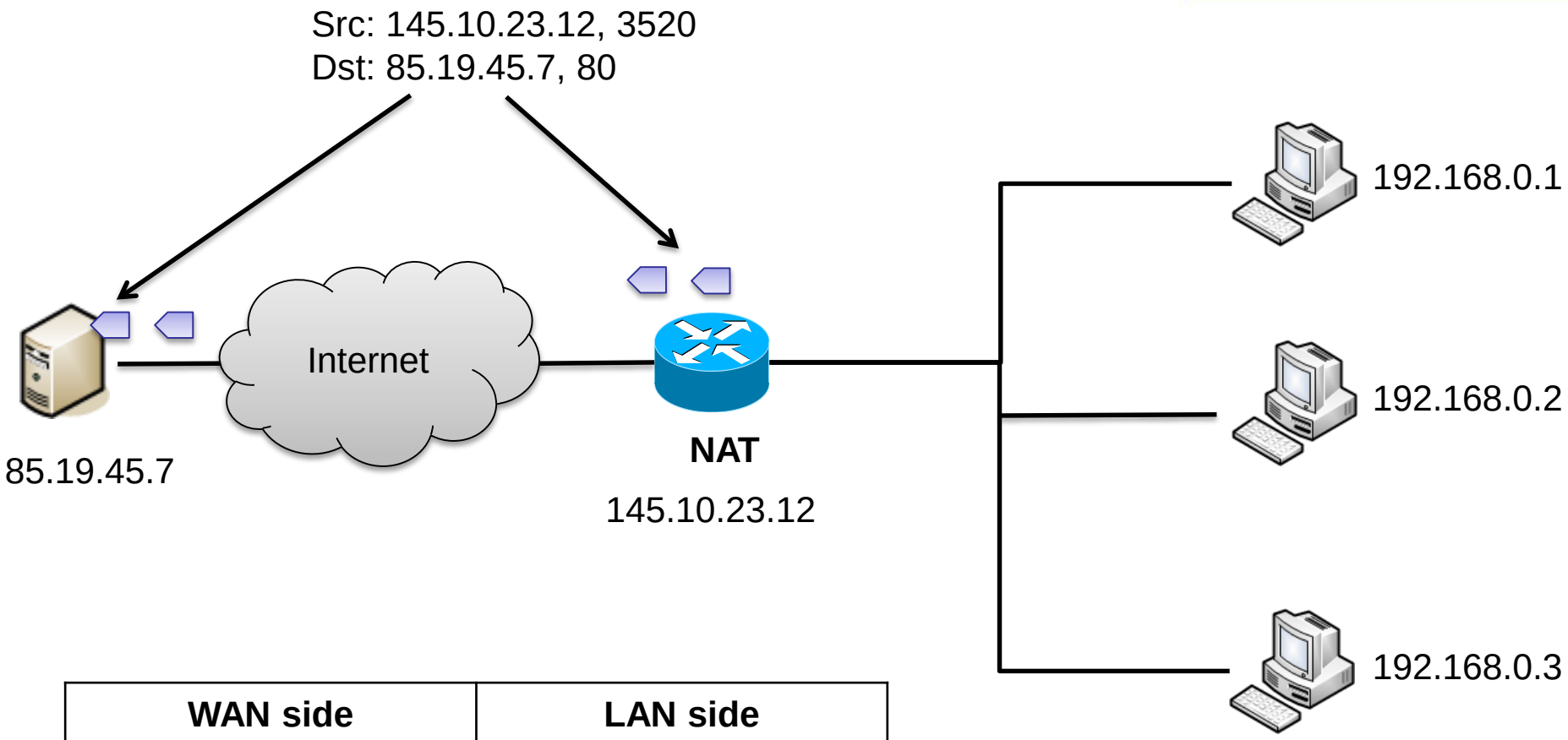
WAN side	LAN side

NAT Example



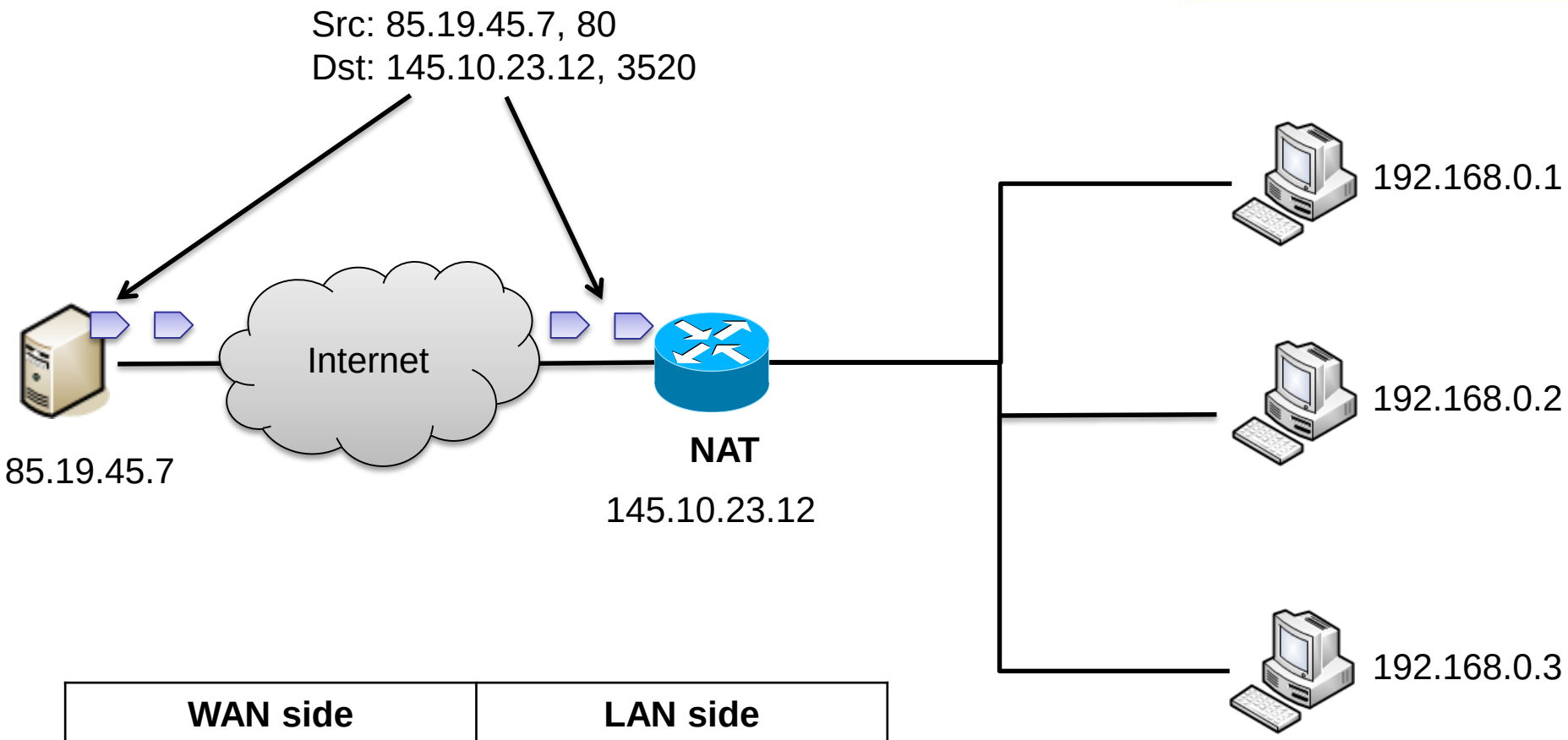
WAN side	LAN side
145.10.23.12, 3520	192.168.0.3, 8080

NAT Example



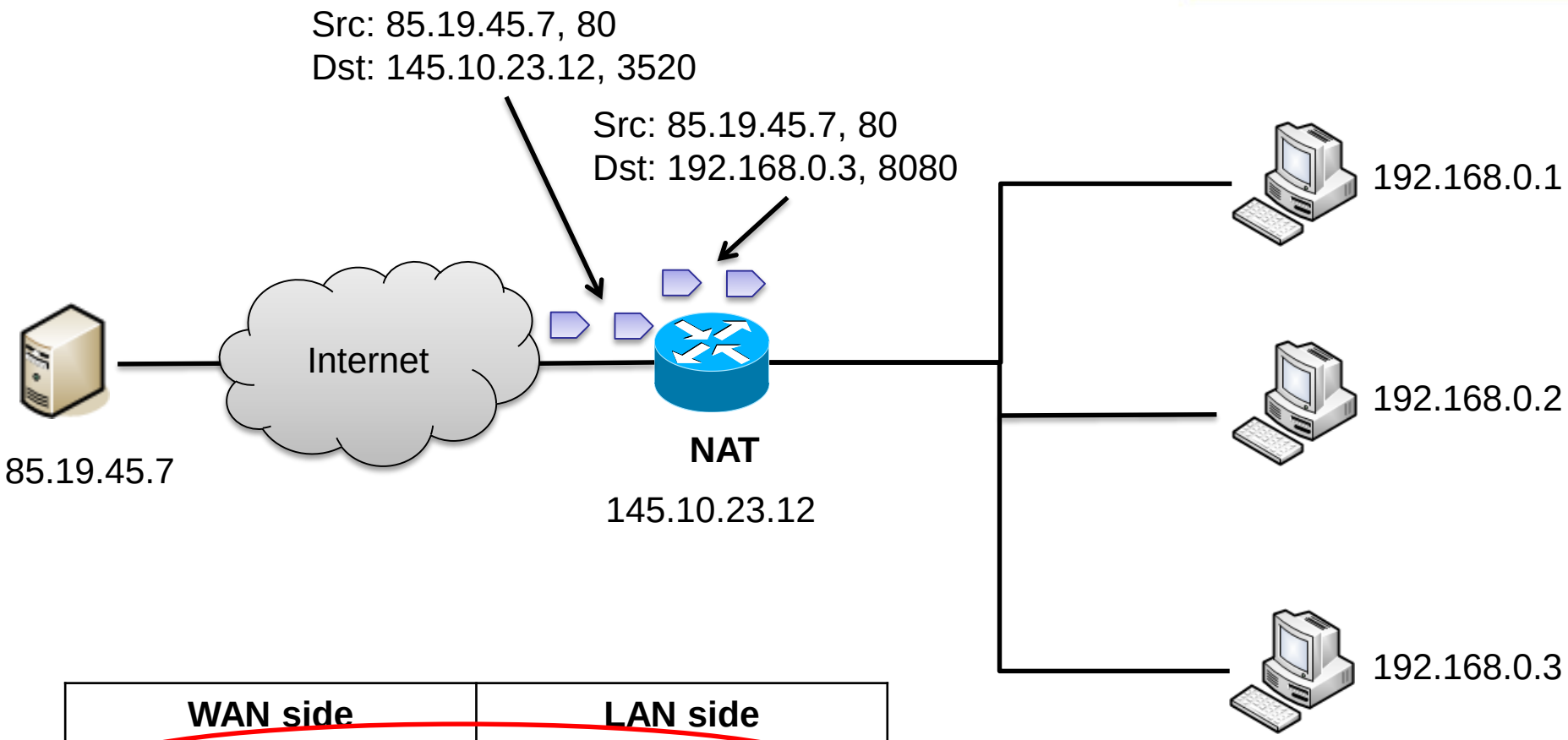
WAN side	LAN side
145.10.23.12, 3520	192.168.0.3, 8080

NAT Example



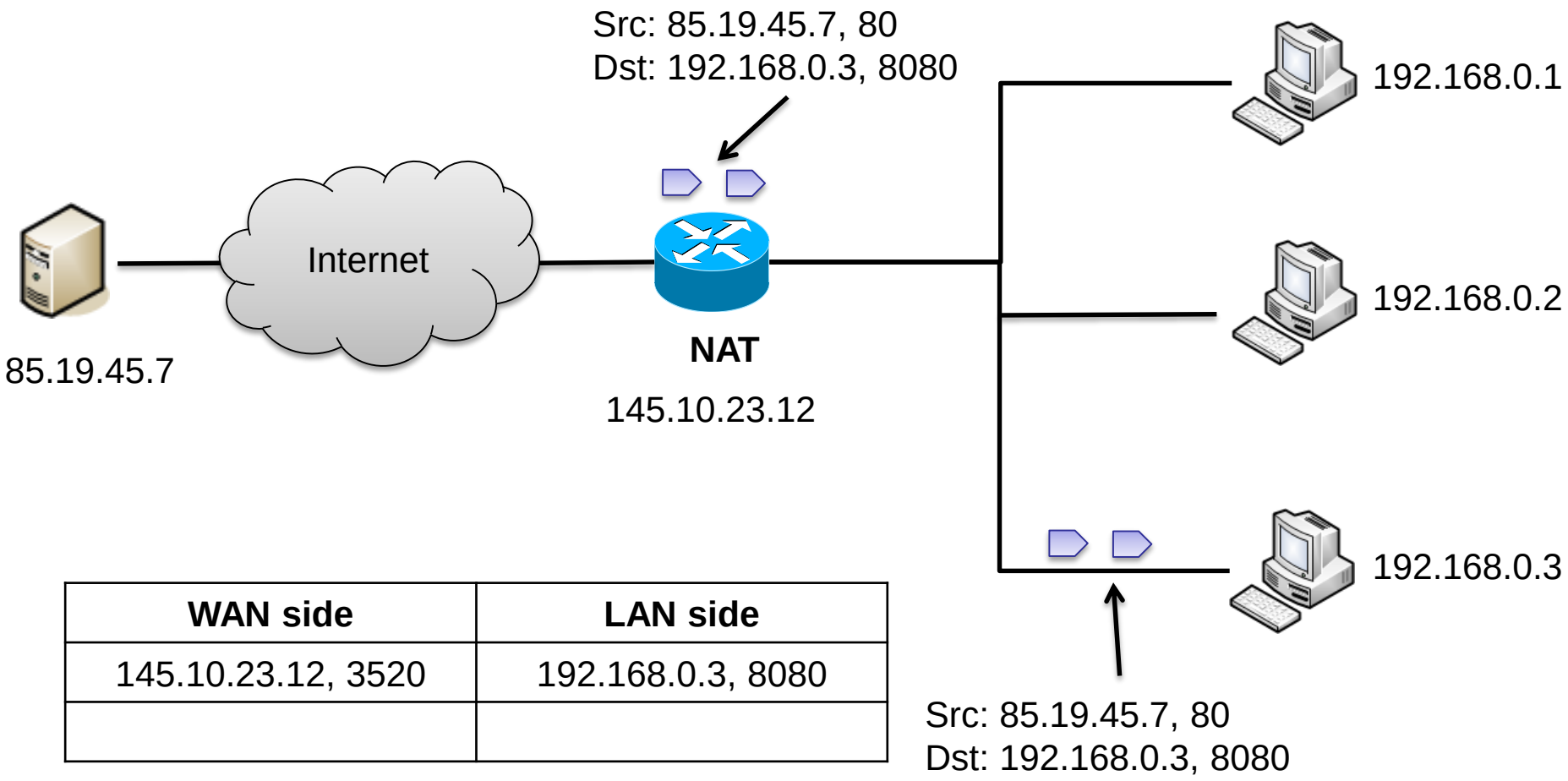
WAN side	LAN side
145.10.23.12, 3520	192.168.0.3, 8080

NAT Example



WAN side	LAN side
145.10.23.12, 3520	192.168.0.3, 8080

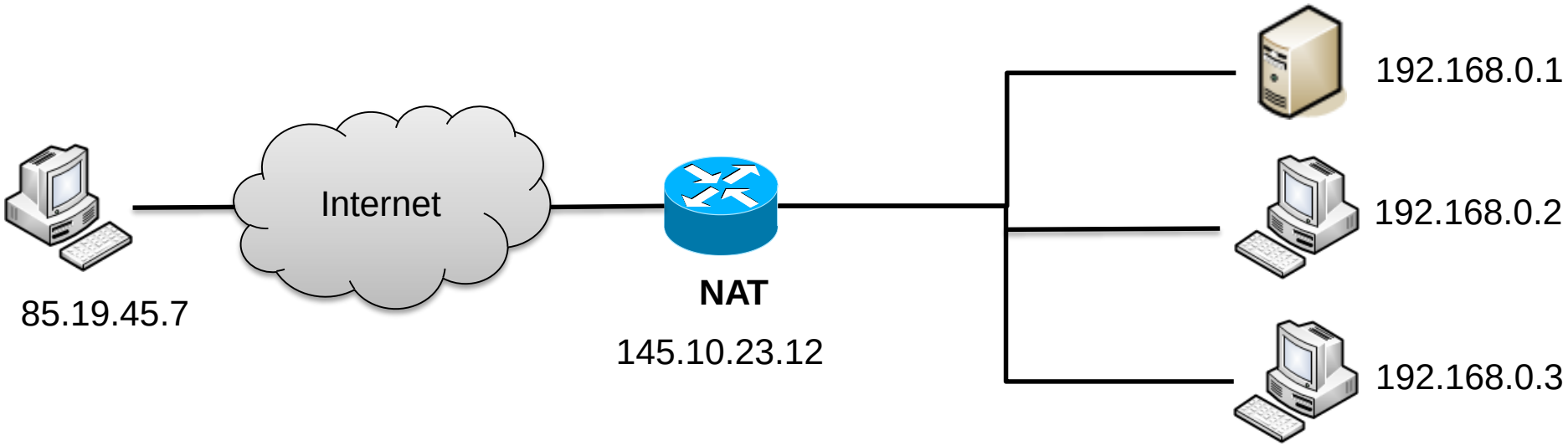
NAT Example



WAN side	LAN side
145.10.23.12, 3520	192.168.0.3, 8080

NAT Traversal Problem

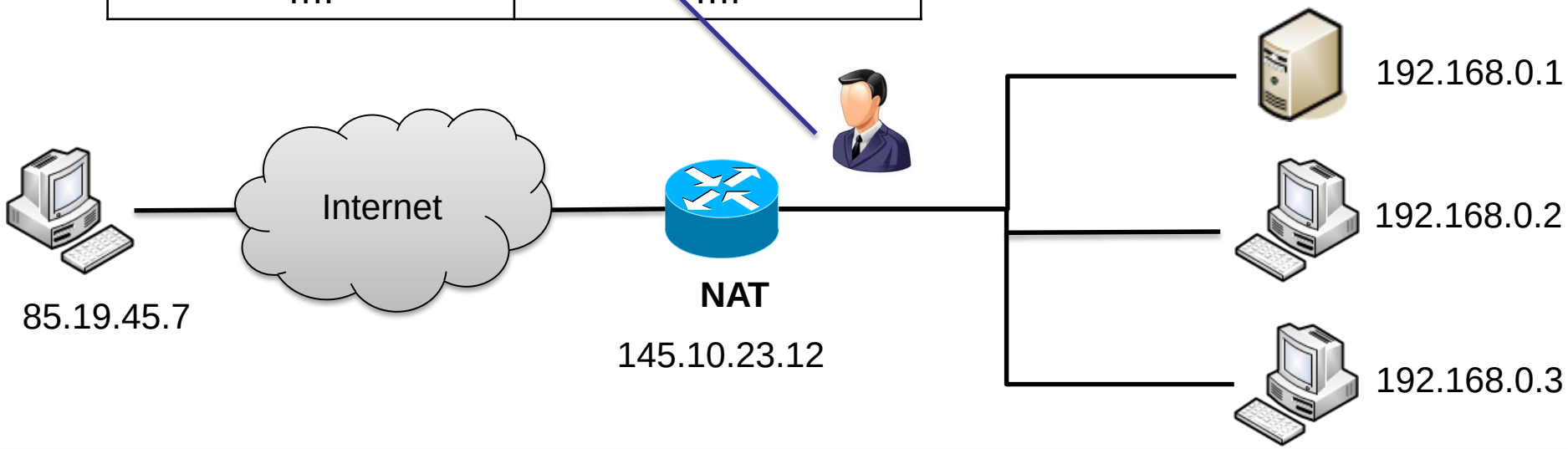
- How can a client establish a connection with a server that is behind a NAT?
  - Server has a private IP address which is not visible by the server
  - NAT's public IP address is visible to the client, but:
    - The client does not know how to address the particular server
    - The NAT does not know which host the client wants to contact



Static NAT Configuration

- The network administrator may configure the NAT statically:
  - NAT will be able to forward incoming connection requests at a given private IP address and port

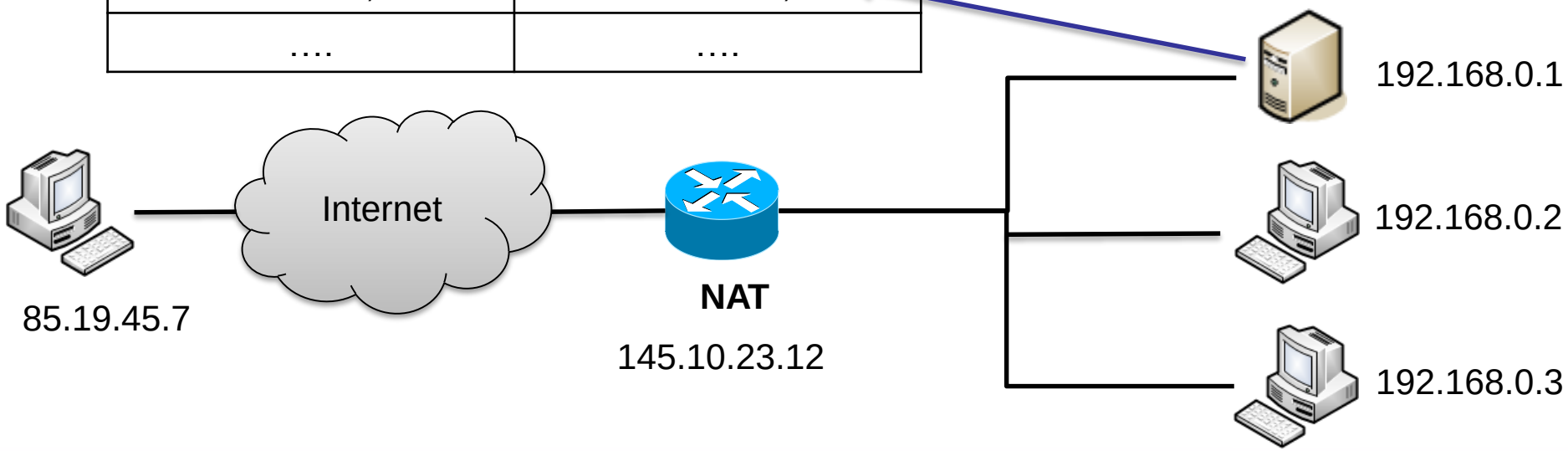
WAN side	LAN side
145.10.23.12, 8080	192.168.0.1, 80
….	….



Automated NAT Configuration

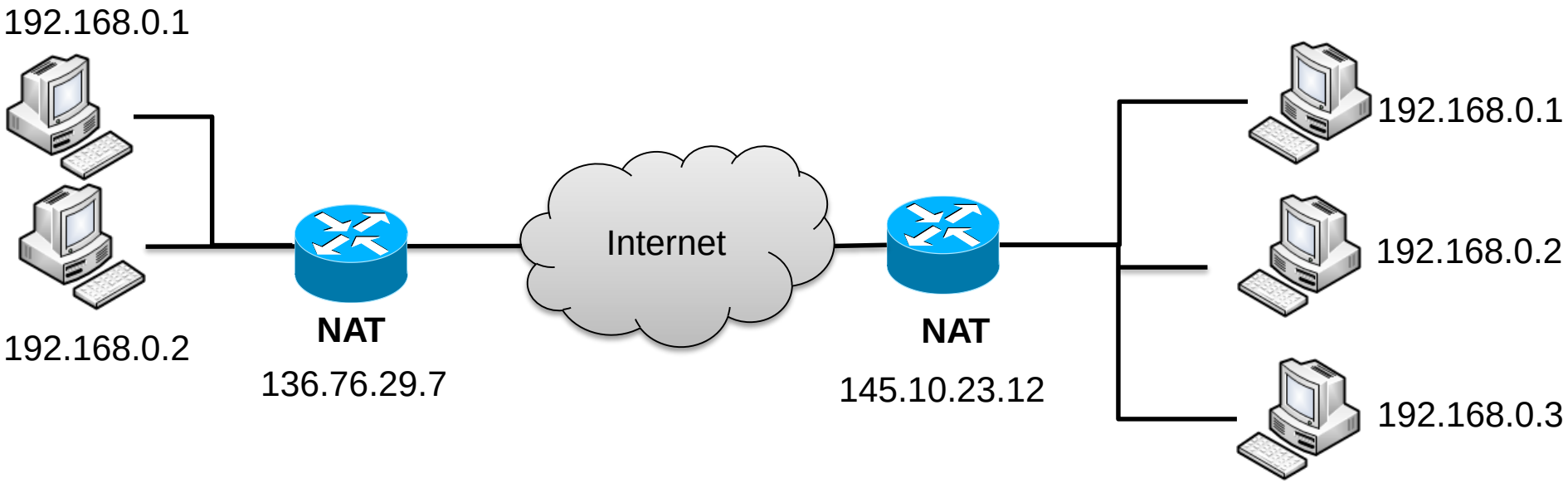
- Automated NAT configuration via Universal Plug-and-Play (UPnP) Internet Gateway Device (IGD) protocol:
  - A host is allowed to directly configure port mappings in the NAT
  - Port mappings are added on a lease basis

WAN side	LAN side
145.10.23.12, 8080	192.168.0.1, 80
….	….



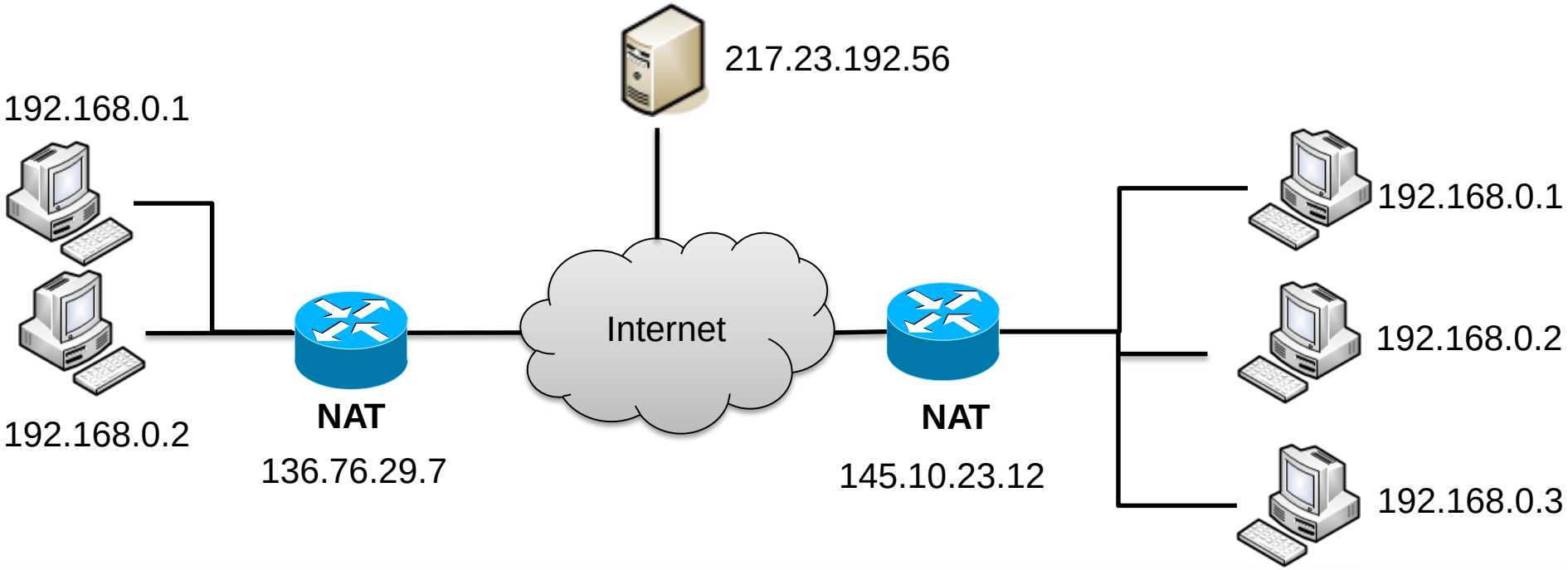
NAT Traversal Problem with 2 NATed Hosts

- How can two hosts establish a connection when both hosts are behind a NAT? (e.g., Skype)
  - Each host does not know the address of the other host
  - Static NAT configuration requires coordination between the administrators of the two networks and is not practical



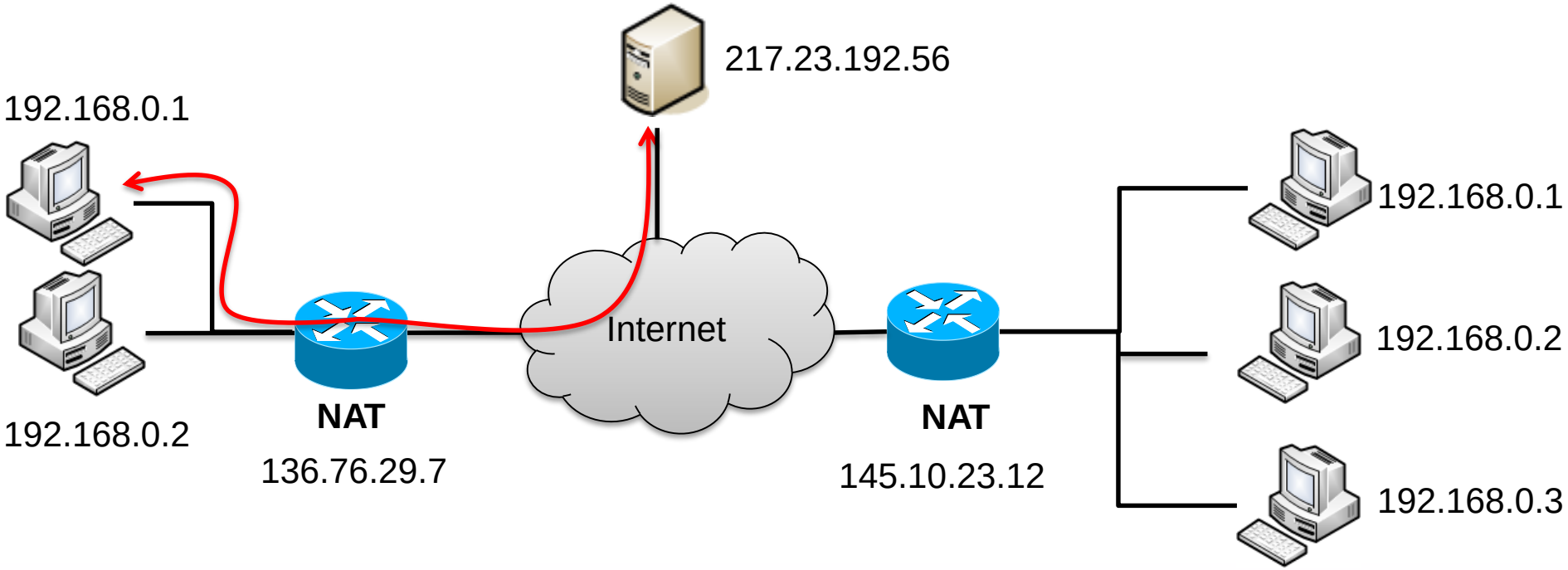
Relaying for NAT Traversal

- Connection can be established using a relay (i.e., server with a public IP address)
  - Each host establishes connection to the relay
  - The relay bridges the two connections



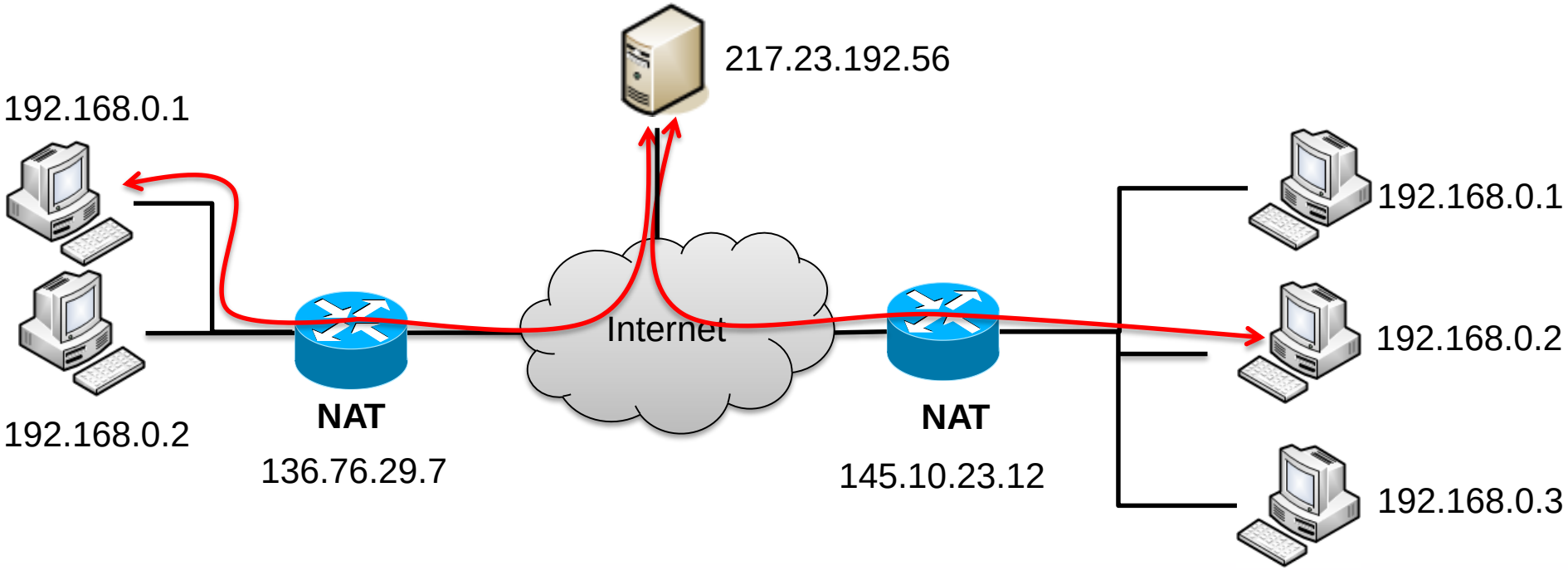
Relaying for NAT Traversal

- Connection can be established using a relay (i.e., server with a public IP address)
  - Each host establishes connection to the relay
  - The relay bridges the two connections



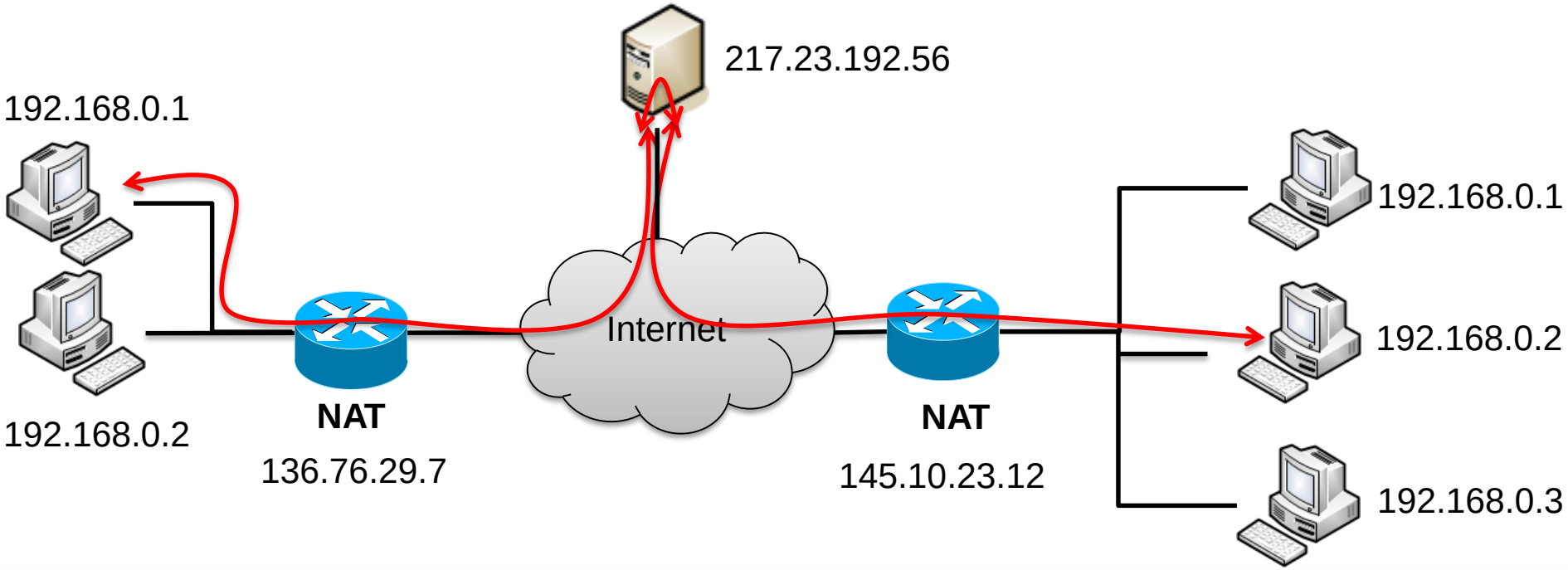
Relaying for NAT Traversal

- Connection can be established using a relay (i.e., server with a public IP address)
  - Each host establishes connection to the relay
  - The relay bridges the two connections



Relaying for NAT Traversal

- Connection can be established using a relay (i.e., server with a public IP address)
  - Each host establishes connection to the relay
  - The relay bridges the two connections



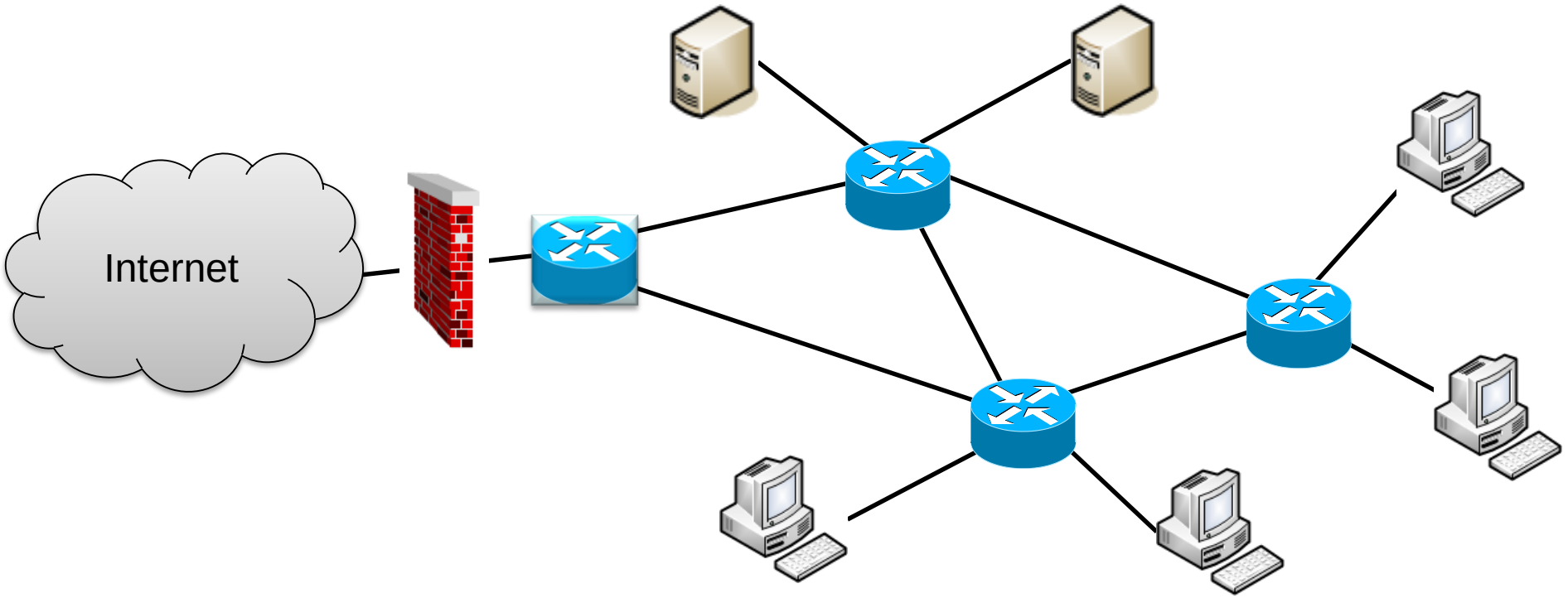
Violations with NAT

- Although NAT is used widely (e.g., ADSL modem/routers), it causes the following violations:
  - NAT uses port numbers to address hosts while port numbers should address network applications
  - NAT enables routers to process packets at layer 4 (i.e., changing port numbers) while routers are supposed to process up to layer 3
  - End-to-end argument
    - Application designers may have to take NATs into account (peer-to-peer applications, e.g., Skype)

Firewall

Firewalls

- Firewalls are used to isolate private networks from the Internet
  - All traffic entering or leaving the private network is inspected by the firewall
  - Only authorized traffic, as defined by local security policy, is allowed to pass



Firewall Taxonomy

- Firewalls can be classified into:
  - Traditional (stateless) packet filters
    - Often combined with a router, allowing the router to operate as a firewall
  - Stateful filters
  - Application gateways (a.k.a. proxy gateways)

- Major firewall vendors:
  - Checkpoint
  - Cisco

Stateless Packet Filters

- Stateless packet filters inspect each IP packet and decide which packets will be allowed to pass based on:
  - source or destination IP address
  - source or destination port number
  - TCP flag bits (e.g., SYN, ACK)
  - protocol (e.g., TCP, UDP, ICMP)
    - Firewalls are often configured to block UDP traffic
  - direction (i.e., packets entering or leaving the private network)
  - router interface
    - Separate filters may be applied per interface

Examples of Stateless Packet Filtering

Policy	Firewall Configuration
No outside Web access	Drop all outgoing packets with destination port 80
No incoming TCP connections, except those for the public Web server.	Drop all incoming TCP SYN packets to any IP address, except the Web server IP address and port 80
Prevent bandwidth-consuming UDP traffic	Drop all incoming UDP packets - except DNS and router broadcasts
Prevent the network from being used for a smurf DoS attack	Drop all ICMP packets going to a “broadcast” address (e.g., 130.207.255.255)
Prevent the network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists (ACL)

- Firewall rules are usually implemented in routers with access control lists
- ACLs are ordered lists of “allow/deny” clauses:
  - A clause includes source/destination IP address, source/destination port, protocol, flag and action
  - A clause may have wild cards
  - Clauses can overlap (the match with the highest priority is selected)

action	source address	destination address	Protocol	source port	destination port	flag bit

ACL Example

- Rules are applied from top to bottom:

action	source address	destination address	protocol	source port	destination port	flag bit
allow	222.22.0.0/16	outside of 222.22.0.0/16	TCP	> 1023	80	any
allow	outside of 222.22.0.0/16	222.22.0.0/16	TCP	80	> 1023	ACK
allow	222.22.0.0/16	outside of 222.22.0.0/16	UDP	> 1023	53	---
allow	outside of 222.22.0.0/16	222.22.0.0/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateless Packet Filters

- Advantages:
  - One firewall (e.g., router with ACL) can protect an entire private network
  - Efficient when security policy and the corresponding filtering rules are simple
  - Easily available (almost every router, Linux PC with iptables)

- Disadvantages:
  - Penetration is possible
  - Enforcing (and testing) complex security policies may be difficult
  - Some policies cannot be applied (e.g., permit certain users)

Penetrating Stateless Packet Filtering

- Firewall allows packets with ACK = 1 and source port 80 to get in:
  - No protection against DoS attacks
  - Blocking incoming TCP packets with ACK = 1 would prevent internal network users from accessing the Web

action	source address	destination address	protocol	source port	destination port	flag bit
allow	222.22.0.0/16	outside of 222.22.0.0/16	TCP	> 1023	80	any
allow	**outside of 222.22.0.0/16**	**222.22.0.0/16**	**TCP**	**80**	**> 1023**	**ACK**
allow	222.22.0.0/16	outside of 222.22.0.0/16	UDP	> 1023	53	---
allow	outside of 222.22.0.0/16	222.22.0.0/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Stateful Packet Filters

- Stateful packet filtering introduces more intelligence to firewalls:
  - Packet filters can keep track of ongoing connections
  - Ongoing connections are kept in a separate table

- Connection tracking:
  - New connections are added to the table by observing a 3-way handshake (SYN, SYN/ACK, ACK)
  - Connections are removed from the table, when:
    - firewall observes a FIN packet
    - there is no activity for a given period (e.g., 60 sec)

Connection Table

- Connection table maintains all active (TCP or UDP) connections:

source address	destination address	source port	destination port
222.22.10.5	125.10.23.89	1520	80
222.22.24.1	65.23.47.1	18912	80
222.22.76.12	212.34.72.42	46390	80

Stateful Packet Filtering

- For every new incoming or outgoing packet, stateful packet filter:
  - Checks filter table
  - If there is a match with a rule in the ACL, the rule indicates whether the connection table needs to be checked
  - If the connection is listed in the connection table, packet is allowed to get in or out

action	source address	destination address	protocol	source port	destination port	flag bit	check connection
allow	222.22.0.0/16	outside of 222.22.0.0/16	TCP	> 1023	80	any	
allow	outside of 222.22.0.0/16	222.22.0.0/16	TCP	80	> 1023	ACK	X
allow	222.22.0.0/16	outside of 222.22.0.0/16	UDP	> 1023	53	---	
allow	outside of 222.22.0.0/16	222.22.0.0/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Stateful Packet Filter Example

- Arrival of packet with:
  - SA: 125.10.23.89, SP: 80
  - DA: 222.22.10.5, DP: 1520
  - SYN = 0, ACK = 1

action	source address	destination address	protocol	source port	destination port	flag bit	check connection
allow	222.22.0.0/16	outside of 222.22.0.0/16	TCP	> 1023	80	any	
allow	outside of 222.22.0.0/16	222.22.0.0/16	TCP	80	> 1023	ACK	X
allow	222.22.0.0/16	outside of 222.22.0.0/16	UDP	> 1023	53	---	
allow	outside of 222.22.0.0/16	222.22.0.0/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Stateful Packet Filter Example

- Arrival of packet with:
  - SA: 125.10.23.89, SP: 80
  - DA: 222.22.10.5, DP: 1520
  - SYN = 0, ACK = 1

action	source address	destination address	protocol	source port	destination port	flag bit	check connection
allow	222.22.0.0/16	outside of 222.22.0.0/16	TCP	> 1023	80	any	
allow	**outside of 222.22.0.0/16**	**222.22.0.0/16**	**TCP**	**80**	**> 1023**	**ACK**	**X**
allow	222.22.0.0/16	outside of 222.22.0.0/16	UDP	> 1023	53	---	
allow	outside of 222.22.0.0/16	222.22.0.0/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Stateful Packet Filter Example

- Arrival of packet with:
  - SA: 125.10.23.89, SP: 80
  - DA: 222.22.10.5, DP: 1520
  - SYN = 0, ACK = 1

source address	destination address	source port	destination port
125.10.23.89	222.22.10.5	80	1520
65.23.47.1	222.22.24.1	80	18912
212.34.72.42	222.22.76.12	80	46390

Stateful Packet Filter Example

- Arrival of packet with:
  - SA: 125.10.23.89, SP: 80
  - DA: 222.22.10.5, DP: 1520
  - SYN = 0, ACK = 1

source address	destination address	source port	destination port
125.10.23.89	**222.22.10.5**	**80**	**1520**
65.23.47.1	222.22.24.1	80	18912
212.34.72.42	222.22.76.12	80	46390

Packet is allowed to get in

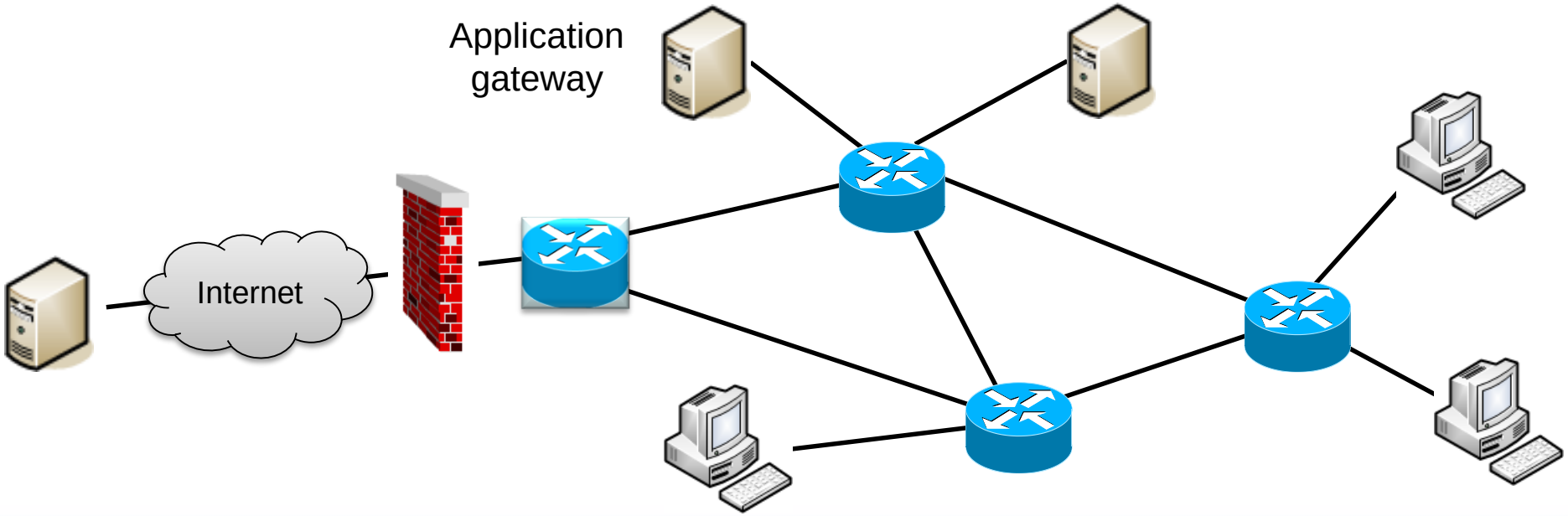
Application Gateways

- Security policy may require decisions based on application data (i.e., only privileged users are allowed to run certain applications)
  - Packet filters are unable to provide this level of control since they make decisions based only on IP/TCP/UDP headers

- Application gateways allow more fine-grained and sophisticated control than packet filtering:
  - An application gateway resides in the private network between the client and the remote server
  - Sessions of applications that fall into local security policy (e.g., SSH) have to be relayed via the gateway
  - The private network operator typically sets up different application gateways for different applications (multiple gateways may be hosted on the same local server)

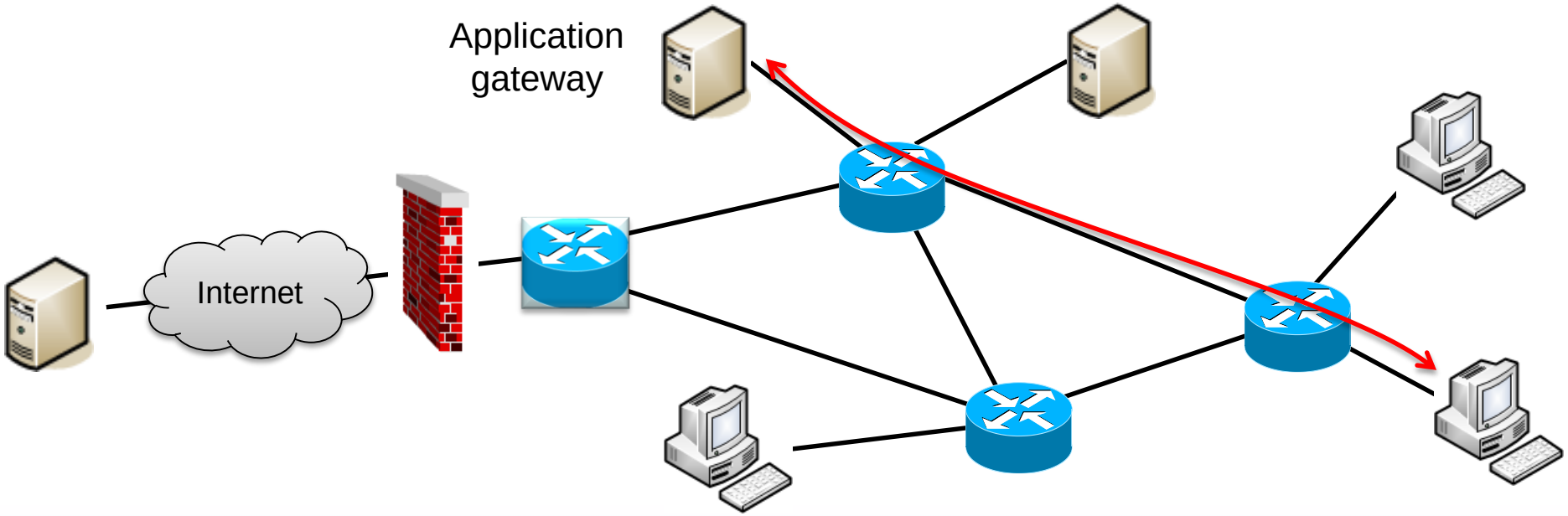
Application Gateway Operation

- A privileged user wants to establish a SSH connection with a remote server:



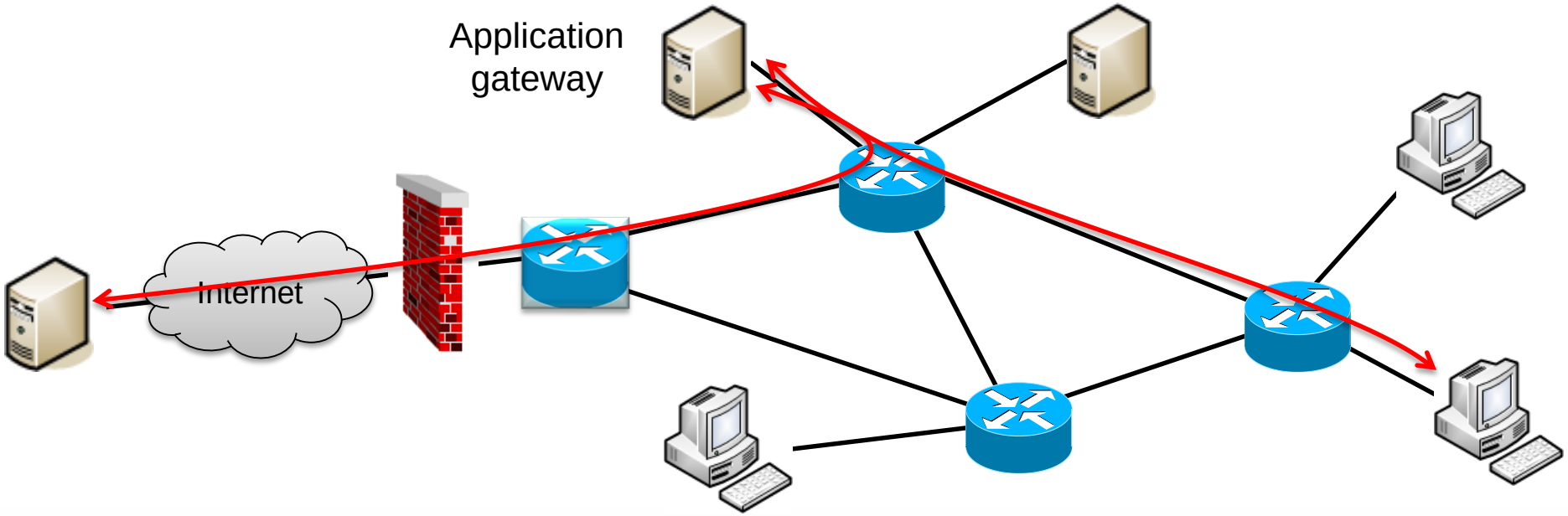
Application Gateway Operation

- A privileged user wants to establish a SSH connection with a remote server:
  - The user connects to the gateway



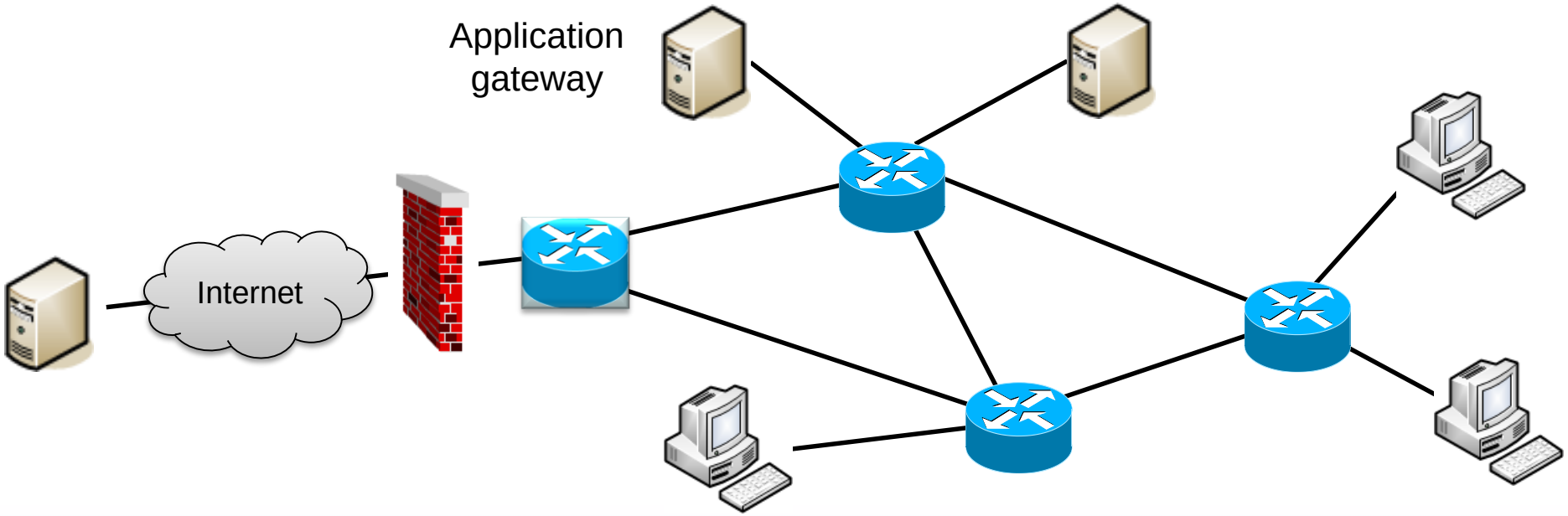
Application Gateway Operation

- A privileged user wants to establish a SSH connection with a remote server:
  - The user connects to the gateway
  - The gateway authenticates the user and sets up the SSH connection with the server
  - The firewall permits the SSH connection, since it originates from the corresponding gateway



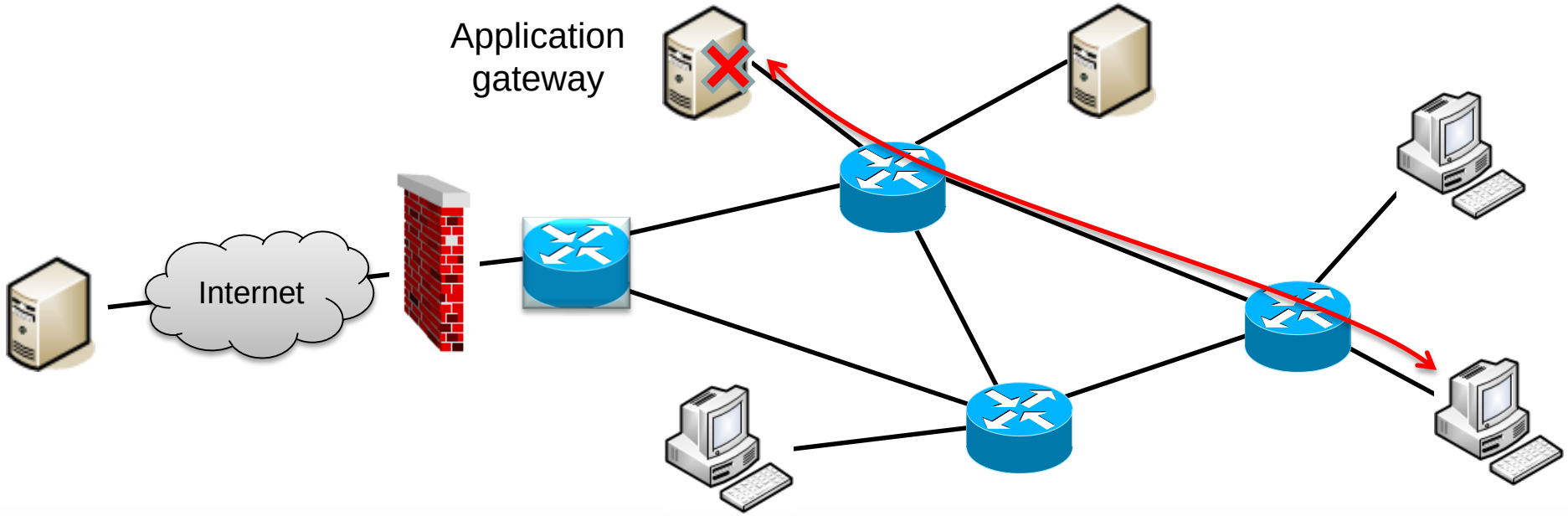
Application Gateway Operation

- A non-privileged user wants to establish SSH with a remote server:



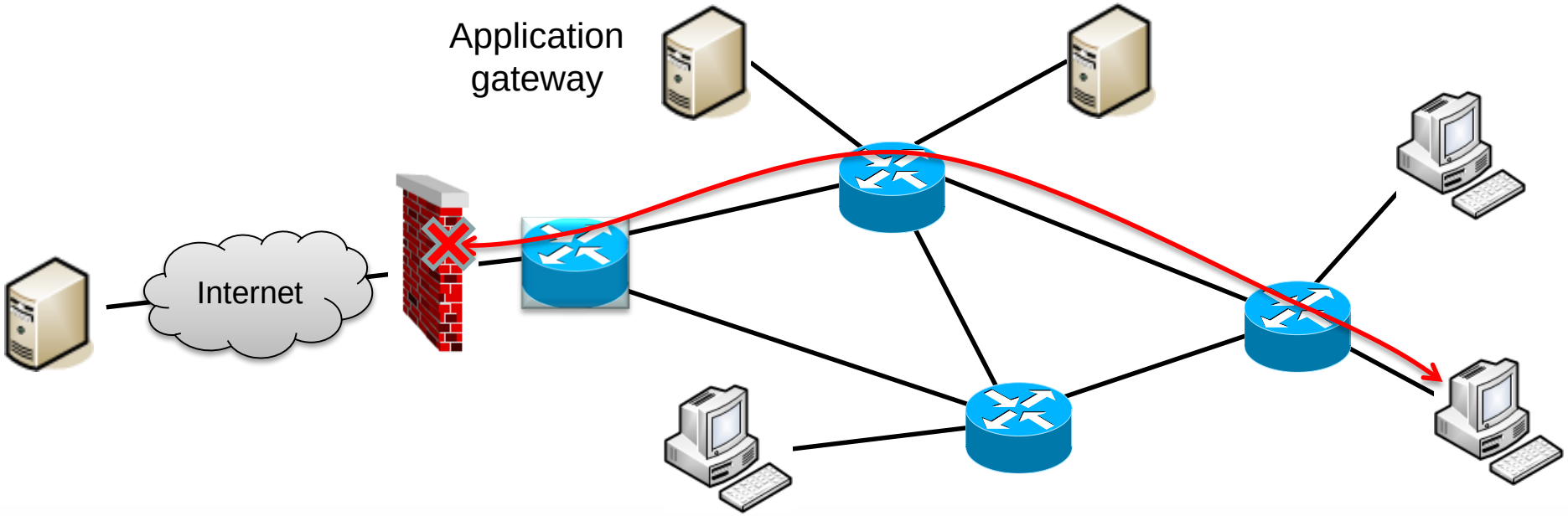
Application Gateway Operation

- A non-privileged user wants to establish SSH with a remote server:
  - If the user connects to the gateway, his SSH request will not be granted



Application Gateway Operation

- A non-privileged user wants to establish SSH with a remote server:
  - If the user connects to the gateway, his SSH request will not be granted
  - If the user attempts to establish a direct connection with the server, the firewall will block it, since it does not originate from the corresponding gateway



Advantages of Application Gateways

- An application gateway can:
  - log all connections, including activity in connections
  - provide caching
    - widely used for web traffic
  - carry out intelligent filtering based on content
  - perform user-level authentication, e.g.:
    - allow certain users to run applications, such as SSH
    - allow only certain users to download large files (e.g., > 10 MB) using FTP

Limitations of Application Gateways

- Performance is degraded for the connections that have to be relayed via the application gateway
- Applications should support proxifying
  - Not all applications have proxied versions
- Different proxy server for each application / service
  - Even worse with the proliferation of Internet services
- Need for end-host configuration
  - A user may have to set the proxy IP address in his application (e.g., web browser)

Intrusion Detection and Prevention

Intrusion Detection Systems

- Packet filtering as used by firewalls cannot provide high protection:
  - Operates only on packet headers (TCP/UDP/IP)
  - Lack of intelligence
    - No correlation check among sessions

- Intrusion detection (IDS) and intrusion prevention systems (IPS) carry out deep packet inspection (DPI) to detect a wider range of attacks, e.g.,
  - port scanning
  - DoS
  - worms
  - viruses
  - OS vulnerabilities
  - application vulnerabilities

IDS vs. IPS

- Intrusion detection systems:
  - perform DPI
  - detect suspicious traffic
  - send alerts (e.g., email, logging) to network administrator so that he can take the appropriate actions on the suspicious traffic (e.g., further inspection and, if needed, filtering)

- Intrusion prevention systems:
  - perform DPI
  - detect suspicious traffic
  - filter suspicious traffic

IDS Classification (1)

- Signature-based systems:
  - use an extensive database of attack signatures to detect suspicious traffic
  - if a packet matches a signature in the database, the IDS generates an alert
  - may generate false alarms for traffic that is not the result of an attack
  - depend on previous knowledge of attacks (i.e., existing signature in the database) and are unable to detect attacks that have not been previously recorded
  - have very high processing requirements

IDS Classification (2)

- Anomaly-based systems:
  - observe traffic and look for packet streams that are statistically unusual, e.g.:
    - inordinate percentage of ICMP packets
    - exponential growth in port scans and ping sweeps
  - do not rely on previous knowledge about existing attacks
  - may be able to detect attacks that have not been previously recorded

Snort

- Snort is a multi-mode packet analysis tool
  - Widely used for intrusion detection

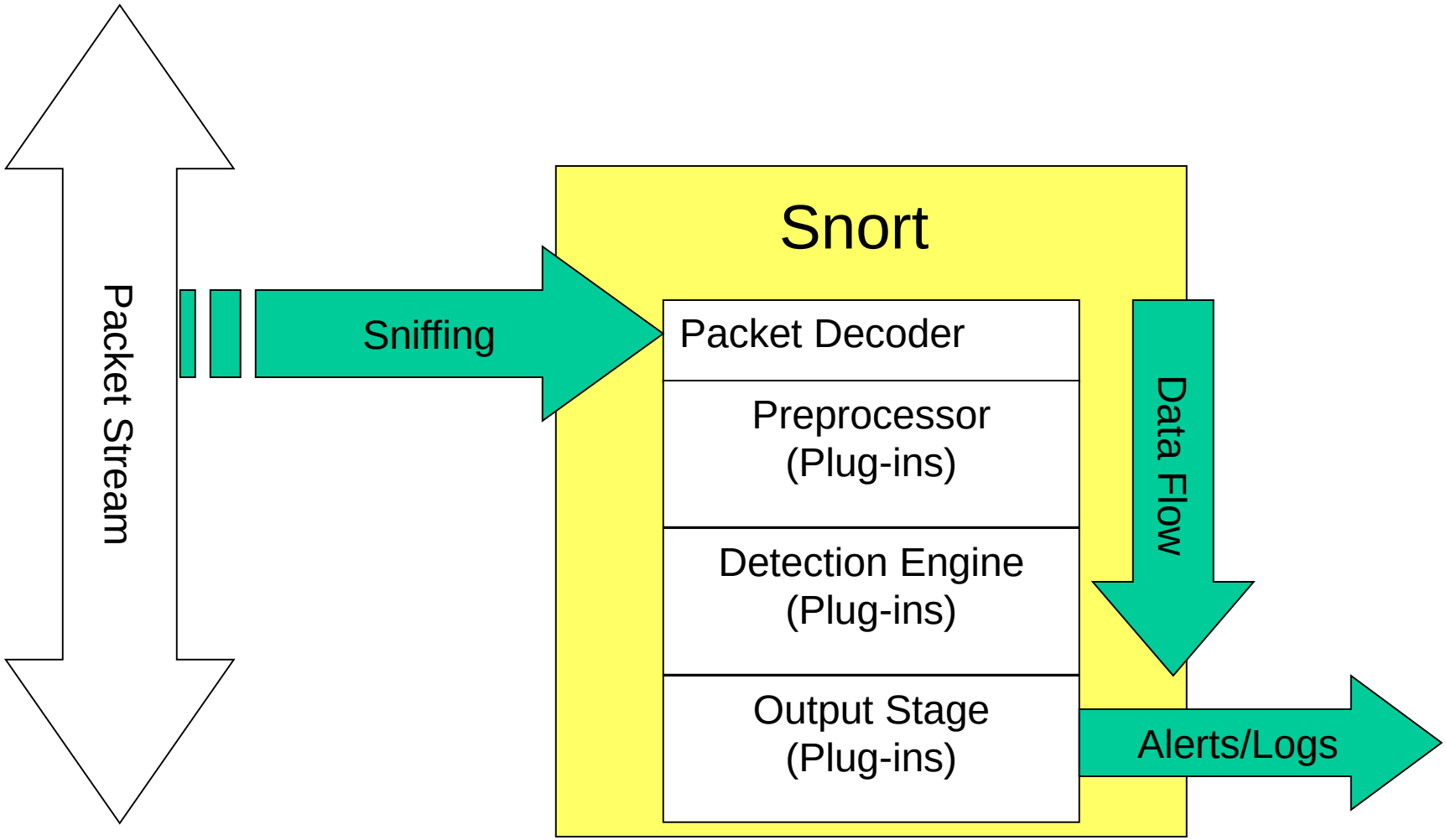
- Three main operational modes:
  - Sniffer mode
  - Packet logger mode
  - IDS mode

- Operational modes are configured via command line parameters
  - Snort automatically switches to IDS mode if no command line parameters are given

Snort Advantages

- Small
  - ~800k source download
- Portable
  - Linux, Windows, MacOS X, Solaris, BSD, etc.
- Fast
  - High probability of detection for a given attack on 100Mbps networks
- Configurable
  - Easy rules language
  - Plenty of reporting/logging options
- Free
  - Open-source (GPL)

Snort Architecture



Snort Components

- Packet Decoder
  - Sniffs packets using the *libcap* library and decodes packets into a data structure for subsequent processing
- Preprocessor
  - Packets are examined/manipulated before being handed to the detection engine
- Detection
  - Performs single, simple tests on a single aspect/field of the packet
- Output
  - Reports results from the other plug-ins

Detection Engine

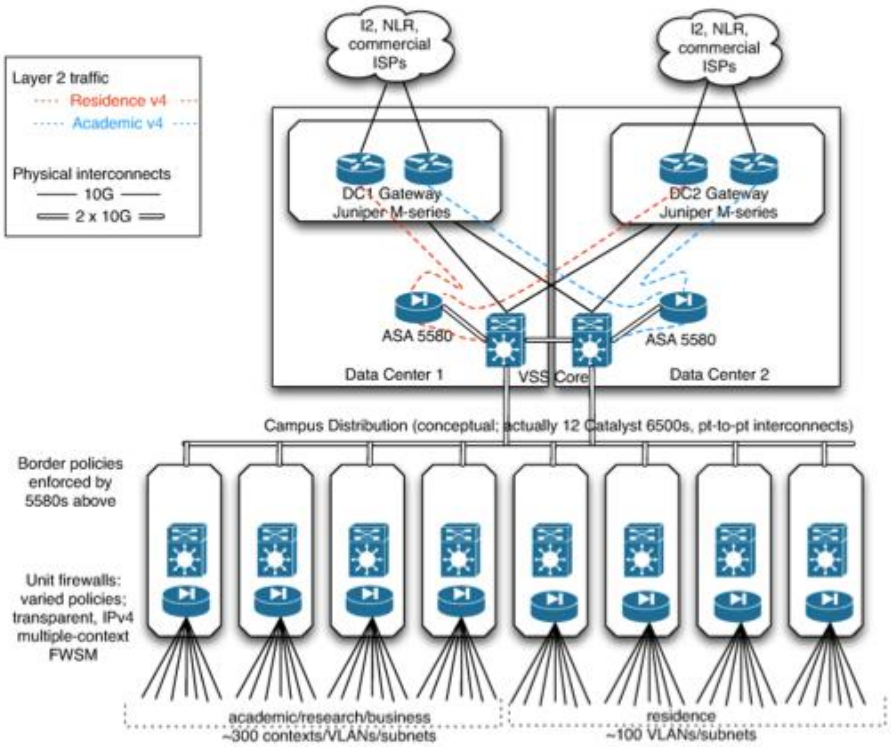
- Rules form “signatures”
- Modular detection elements are combined to form these signatures
- Wide range of detection capabilities:
  - Stealth scans, OS fingerprinting, buffer overflows, back doors, CGI exploits, etc.
- Rules system is very flexible, and creation of new rules is relatively simple

Rule Header	Rule Options
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: SF; msg: “SYN-FIN Scan”;)
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: S12; msg: “Queso Scan”;)
Alert tcp 1.1.1.1 any -> 2.2.2.2 any	(flags: F; msg: “FIN Scan”;)

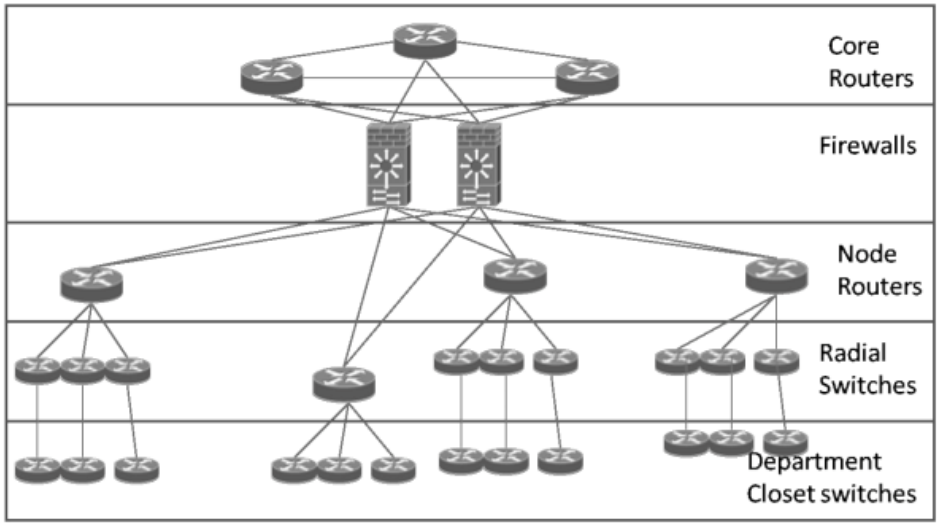
Network Device Configuration Statistics

Statistics from H. Kim, et al., “The Evolution of Nework Configuration: A Tale of Two Campuses”

Statistics from two University Campuses



(a) Georgia Tech



(b) University of Wisconsin

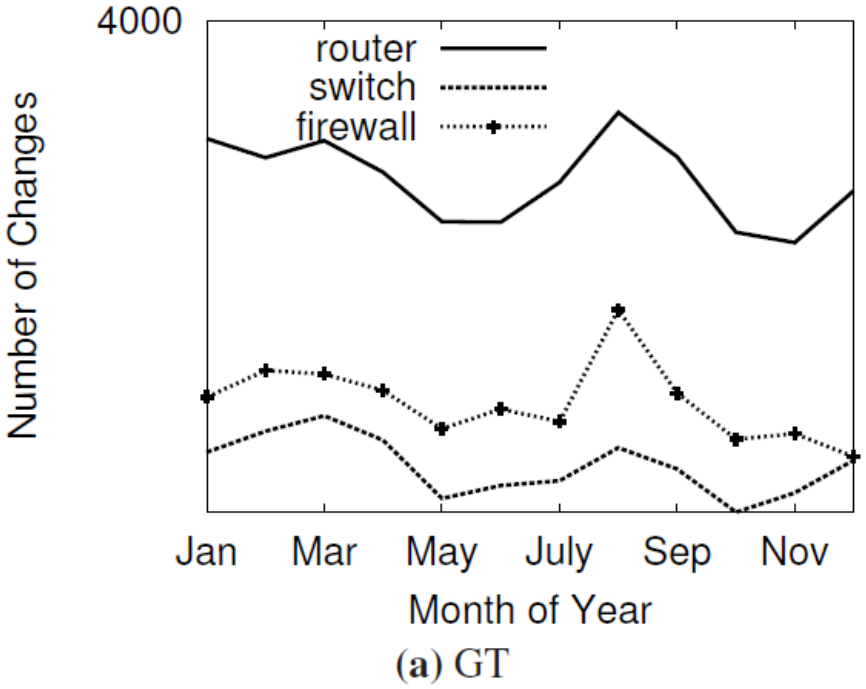
	Routers	*Firewalls*	*Switches*	*Total*
Georgia Tech	16	365	716	1097
Wisconsin	53	325	1246	1624

Network Device Configuration Changes

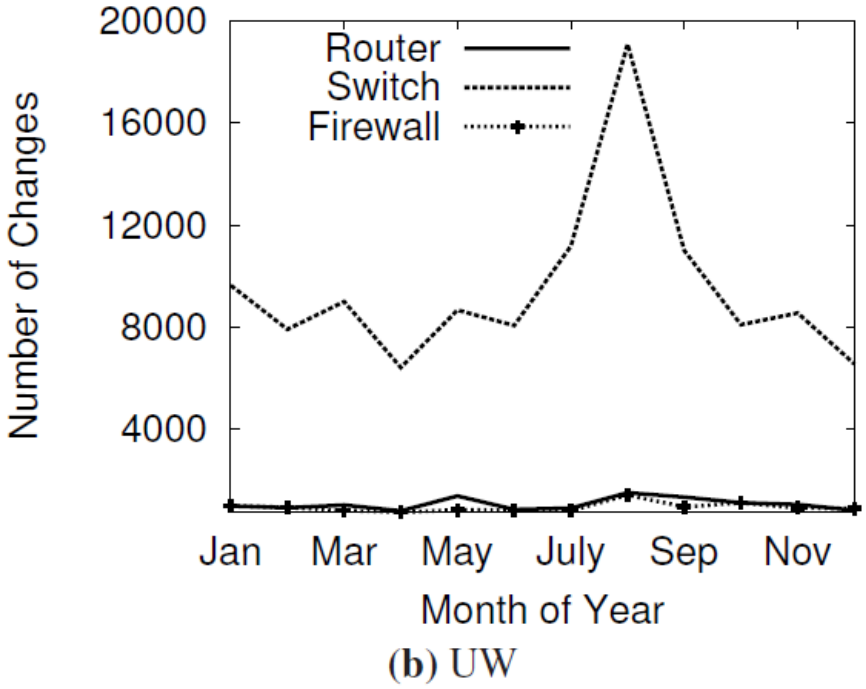
Georgia Tech	*add*	*del*	*mod*	Total
Routers (16)	31,178	27,064	262,216	326,458
Firewalls (365)	249,595	118,571	171,005	539,171
Switches (716)	216,958	20,185	116,277	353,420
Rtr avg. per device	2,324	1,692	16,389	20,404
FW avg. per device	684	325	469	1,477
Swt avg. per device	303	28	162	494

UW-Madison	*add*	*del*	*mod*	Total
Routers (53)	79,202	38,288	154,407	271,897
Firewalls (325)	193,499	73,827	161,863	429,189
Switches (1246)	1608,512	213,910	1,768,384	3,590,806
Rtr avg. per device	1,494	722	2,913	5,130
FW avg. per device	595	227	498	1,321
Swt avg. per device	1,291	172	1,419	2,882

Network Device Configuration Changes over Time



(a) GT

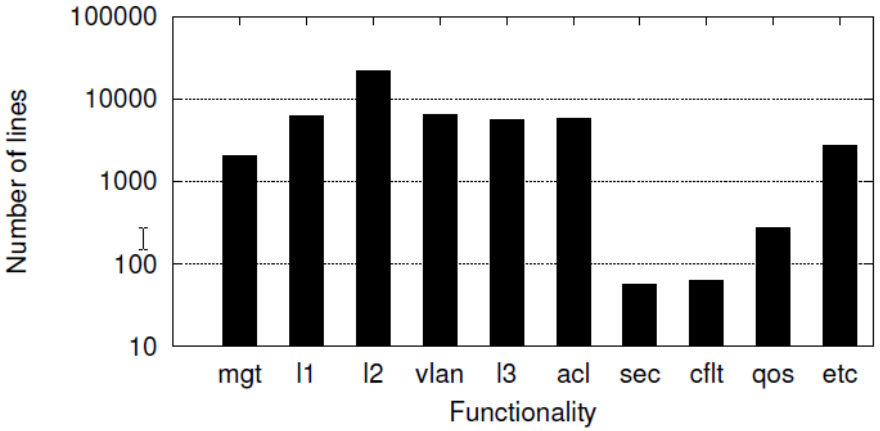


(b) UW

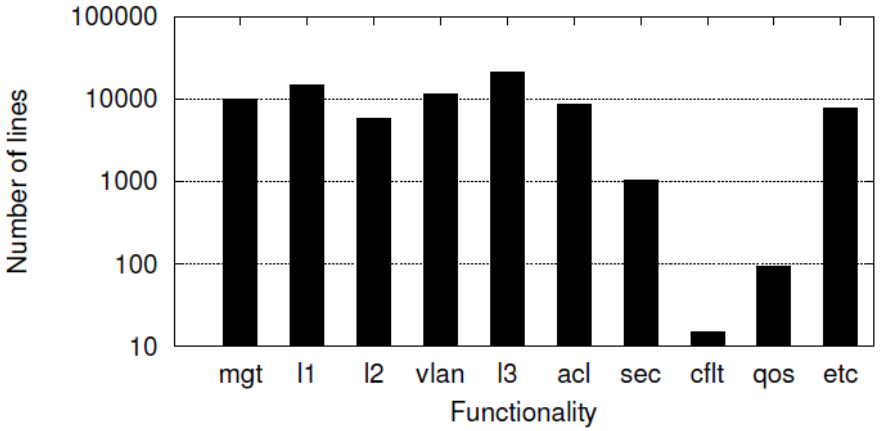
Network Device Functionality Map

	Meaning	Examples
mgt	device management settings	username, password, telnet, ssh, logging, aaa, clock, console, radius-server
l1	interface/port settings	interface definition, bandwidth, switchport, description, duplex
l2	layer 2 settings	arp x.x.x.x, mac-address, ip proxy-arp, arp timeout, spanning-tree
vlan	VLAN settings	vlan definition, switchport mode trunk, switchport mode access vlan, set vtp, set vmps
l3	layer 3 related	ip address x.x.x.x , ip gateway x.x.x.x, ip route x.x.x.x, nat, router bgp/ospf/rip, router-id, neighbor
acl	access control	object define, access-list, permit, deny
sec	security related	vpn, ipsec, crypto, webvpn, any-connect, ssl, tunnel-group, flood-guard
cflt	control filtering	prefix-list
qos	QoS	policy-map, class-map, service-policy, port-channel load-balance, set qos

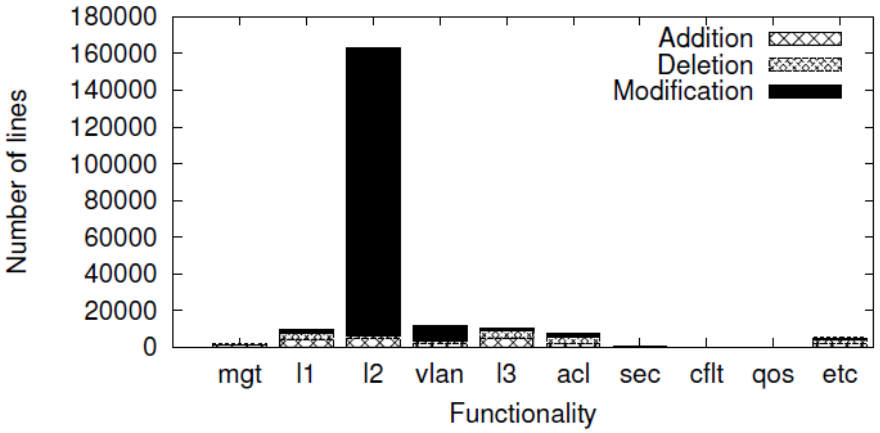
Router Configuration Changes



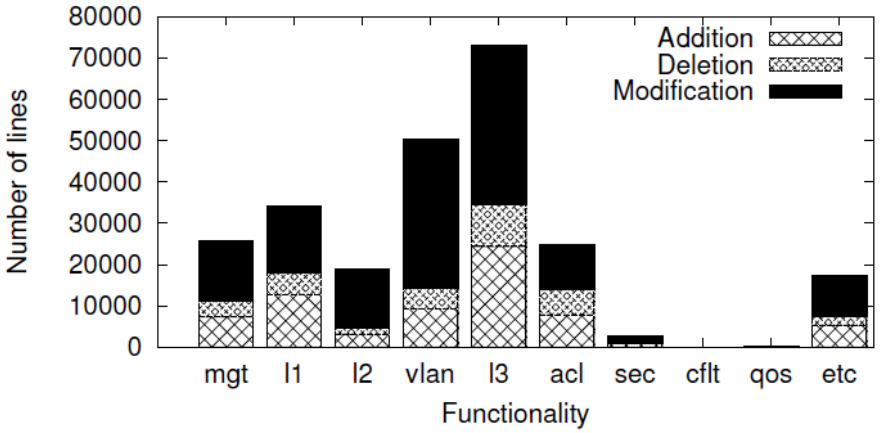
(a) Static analysis of latest snapshot (logscale) - *GT*



(b) Static analysis of latest snapshot (logscale) - *UW*

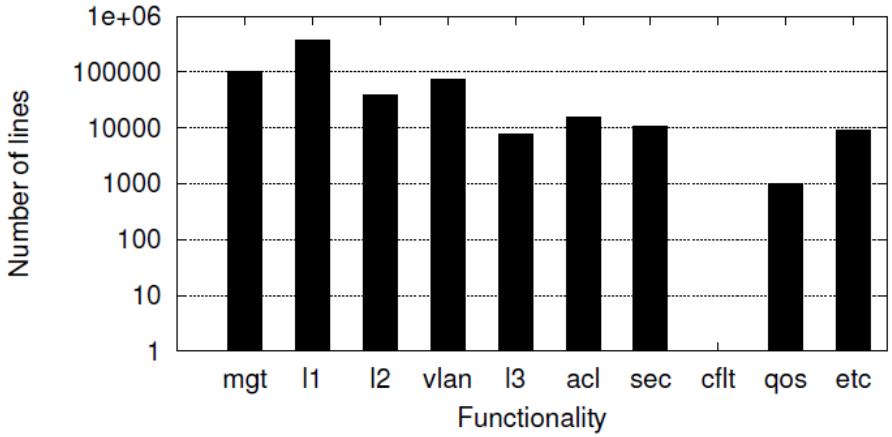


(c) Change characteristic over five years - *GT*

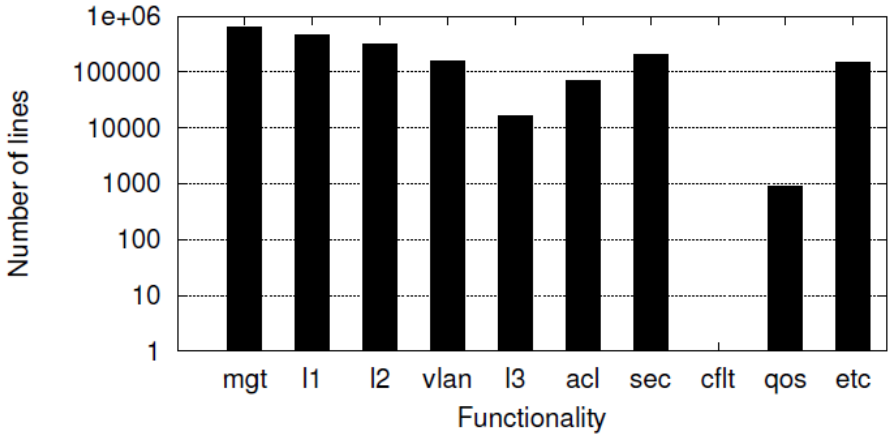


(d) Change characteristic over five years - *UW*

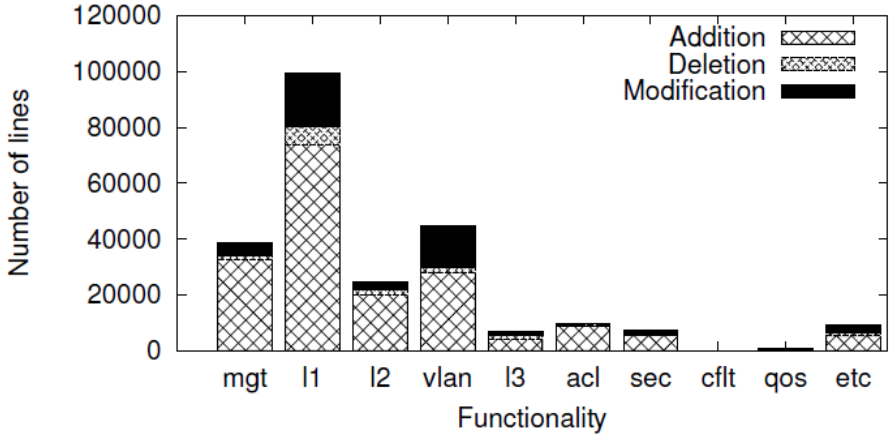
Switch Configuration Changes



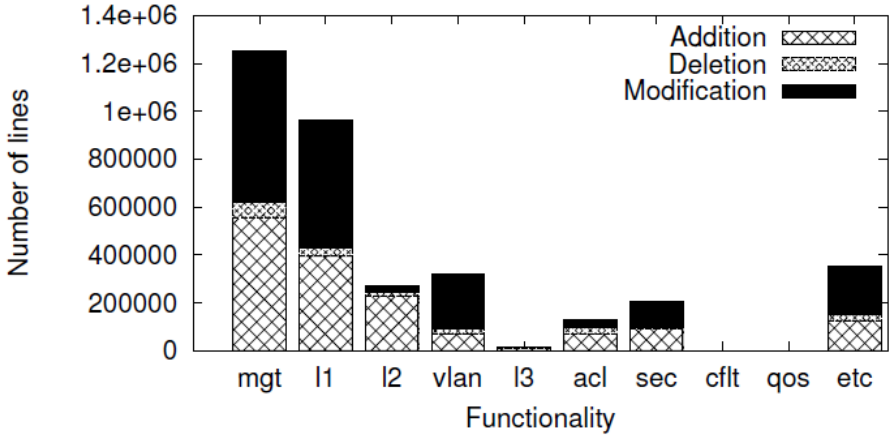
(a) Static analysis of latest snapshot (logscale) - *GT*



(b) Static analysis of latest snapshot (logscale) - *UW*

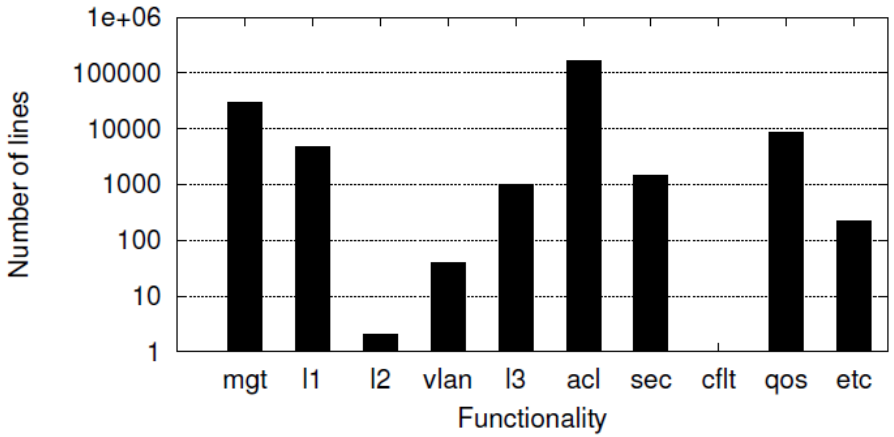


(c) Change characteristic over five years - *GT*

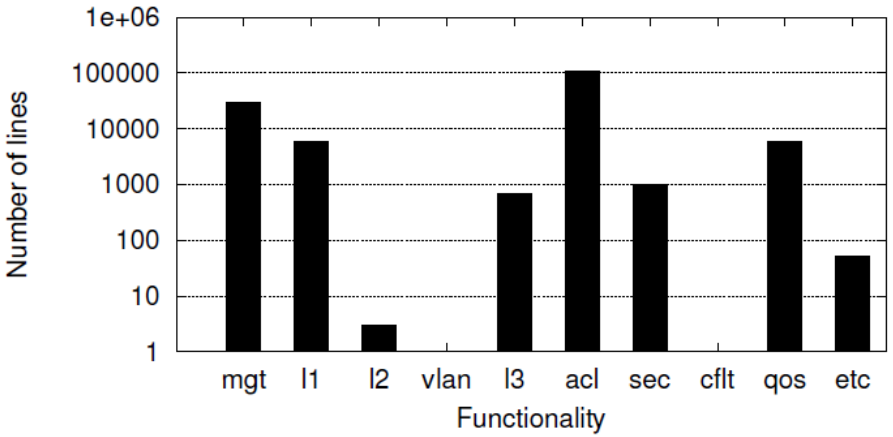


(d) Change characteristic over five years- *UW*

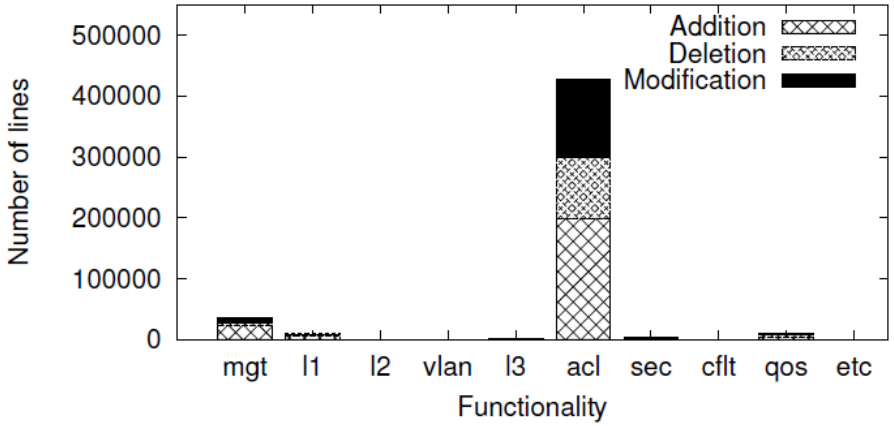
Firewall Configuration Changes



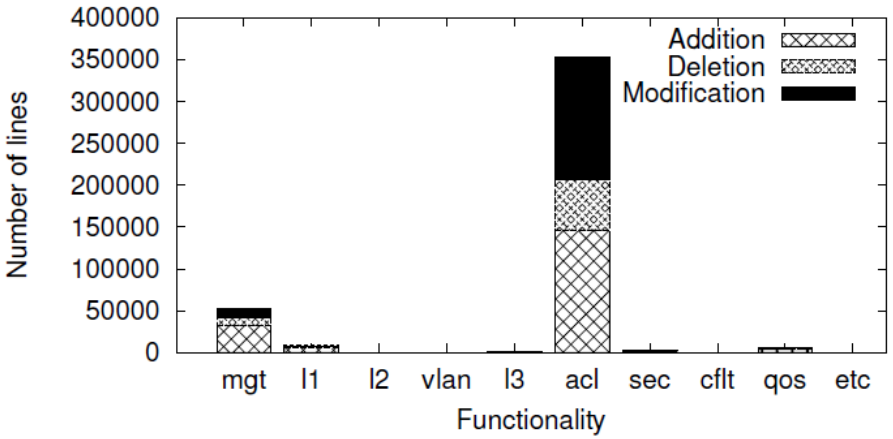
(a) Static analysis of latest snapshot (logscale) - *GT*



(b) Static analysis of latest snapshot (logscale) - *UW*



(c) Change characteristic over five years- *GT*



(d) Change characteristic over five years- *UW*

References

- H. Kim, et al., **The Evolution of Network Configuration: A Tale of Two Campuses**, ACM IMC 2011
- **Snort**, www.snort.org/
- **Bro**, http://bro-ids.org