CSCE 465 Computer & Network Security

Instructor: Sungmin Kevin Hong

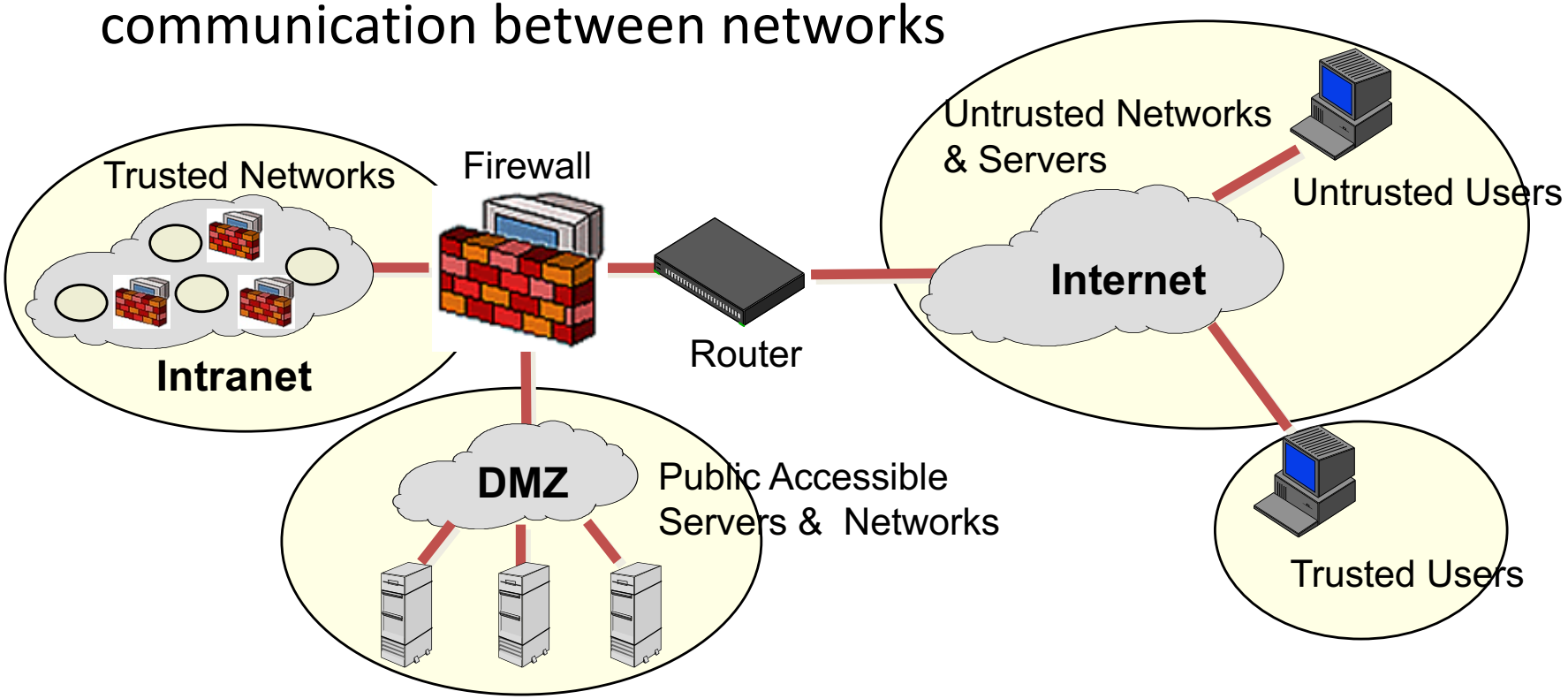
Firewalls

Roadmap

- Basic firewall concept
- Filtering firewall
- Proxy firewall
- Network Address Translation

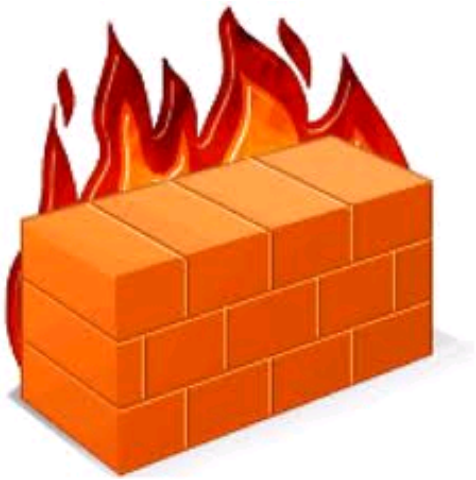
What is a firewall?

- Device that provides secure connectivity between networks (internal/external; varying levels of trust)
- Used to implement and enforce a security policy for communication between networks



Firewalls

- From Webster's Dictionary: *a wall constructed to prevent the spread of fire*
- Internet firewalls are more the moat around a castle than a building firewall
- Controlled access point



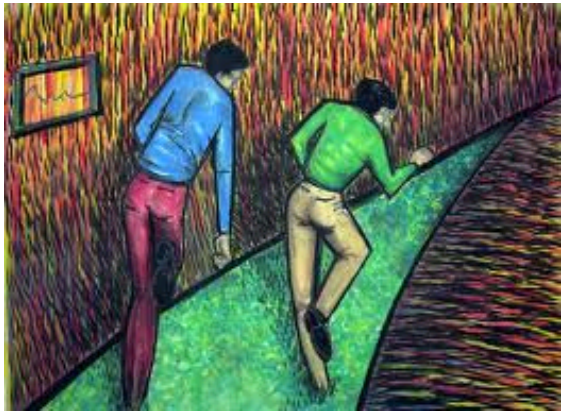
Firewalls can:

- Restrict incoming and outgoing traffic by IP address, ports, or users
- Block invalid packets

- Convenient
  - Give insight into traffic mix via logging
  - Network Address Translation
  - Encryption

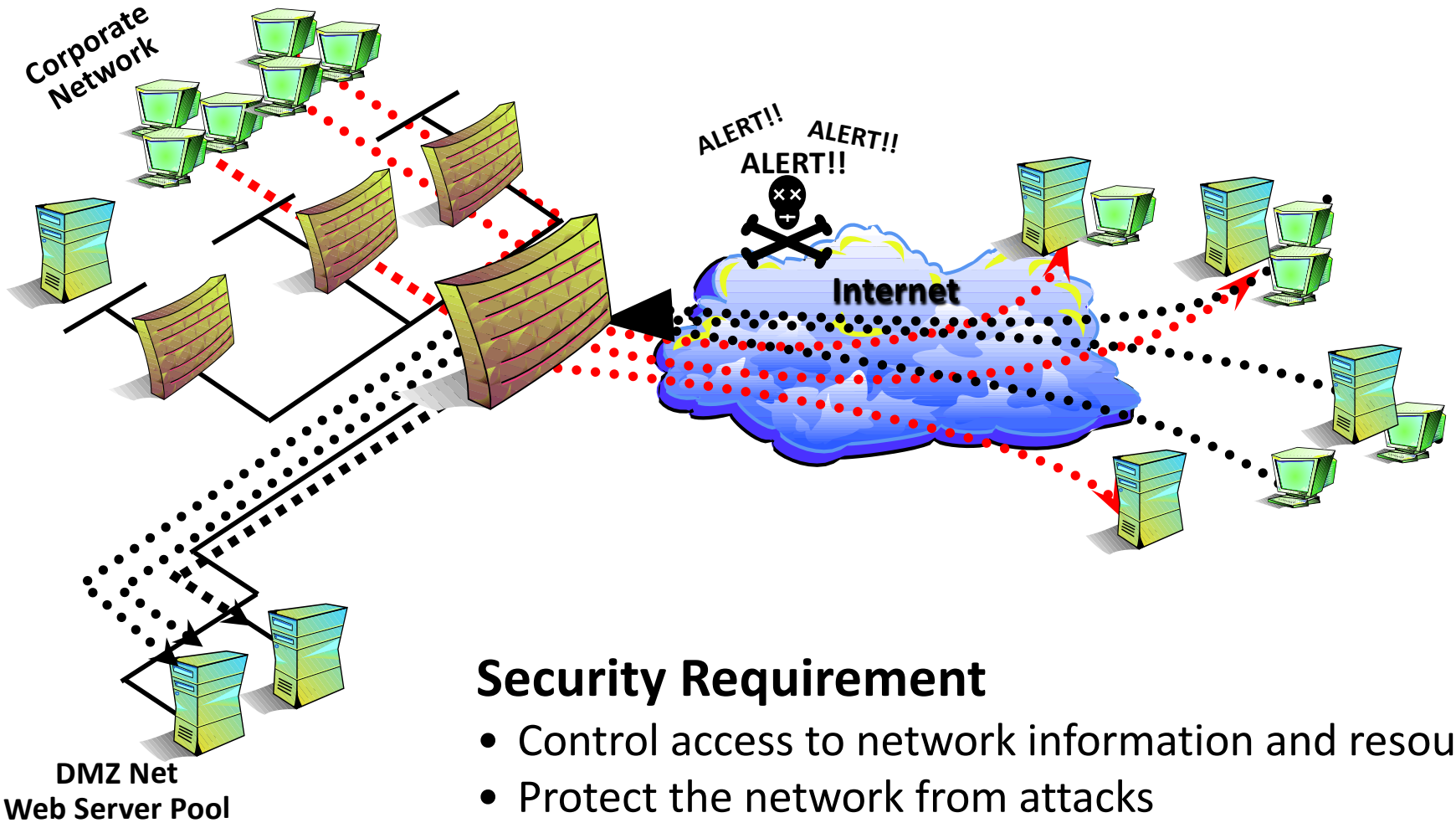
Firewalls Cannot Protect…

- Traffic that does not cross it
  - routing around
  - Internal traffic



- When misconfigured

Access Control



Security Requirement

- Control access to network information and resources
- Protect the network from attacks

FILTERING FIREWALL

Filtering Firewall

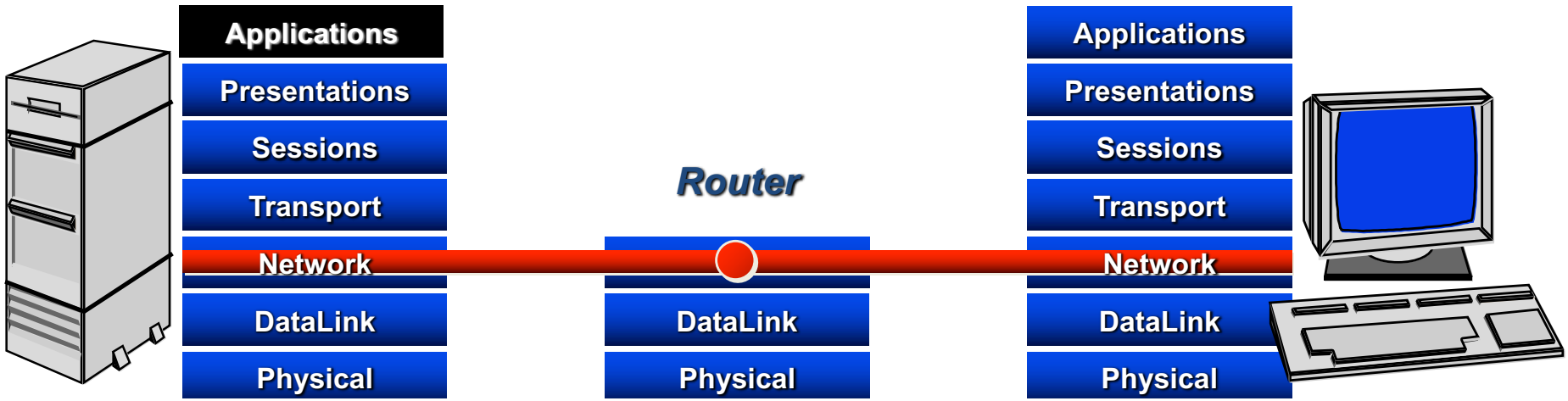
- Packets checked then passed
- Inbound & outbound affect when policy is checked

Filtering

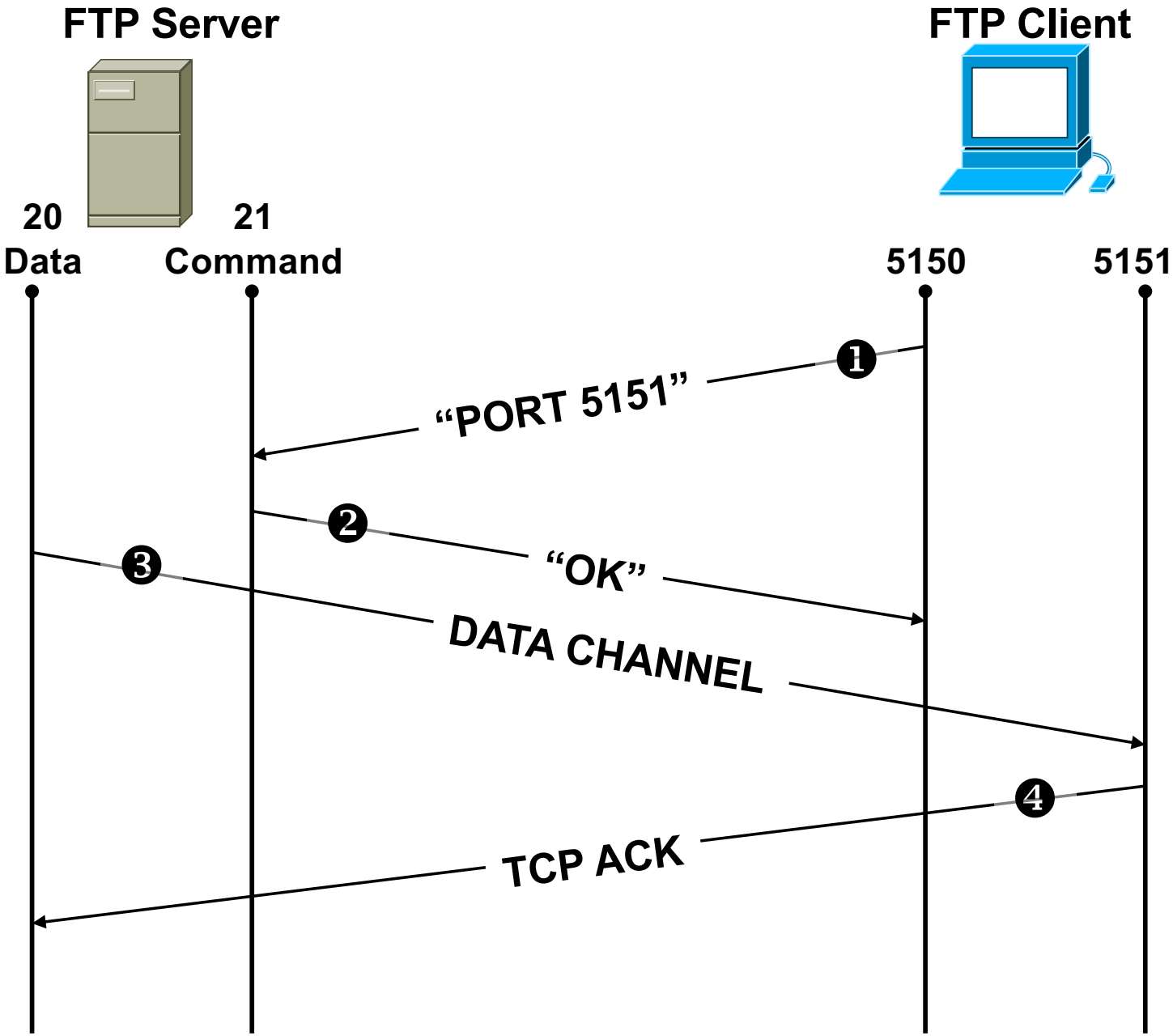
- Packet filtering
  - Access Control Lists
- Session filtering
  - Dynamic Packet Filtering
  - Stateful Inspection
  - Context Based Access Control

Packet Filtering

- Decisions made on a per-packet basis
- No state information saved
- Typical Configuration
  - Ports > 1024 left open
  - If dynamic protocols are in use, *entire ranges of ports must be allowed* for the protocol to work.



Example: FTP Protocol



Example FTP – Packet Filter

Format:

access-list *<rule number> <permit|deny> <protocol> <SOURCE host with IP address| any|IP address and mask> [<gt|eq port number>] <DEST host with IP address| any|IP address and mask> [<gt|eq port number>]*

The following allows a user to FTP (not passive FTP) from any IP address to the FTP server (172.168.10.12) :

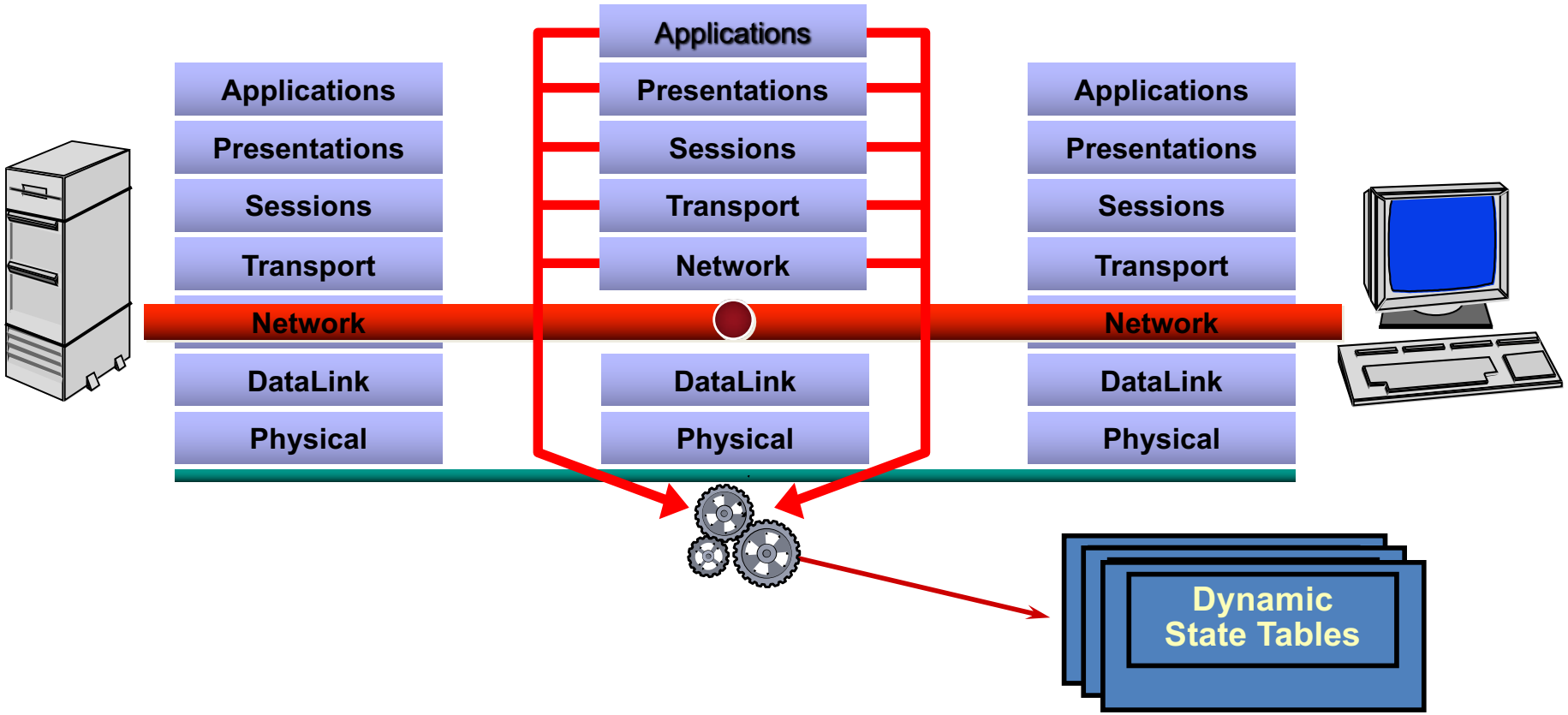
```
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 21
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 20
  ! Allows packets from any client to the FTP control and data ports
access-list 101 permit tcp host 172.168.10.12 eq 21 any gt 1023
access-list 101 permit tcp host 172.168.10.12 eq 20 any gt 1023
  ! Allows the FTP server to send packets back to any IP address with TCP ports > 1023

interface Ethernet 0
 access-list 100 in     ! Apply the first rule to inbound traffic
 access-list 101 out    ! Apply the second rule to outbound traffic
!
```

Session Filtering

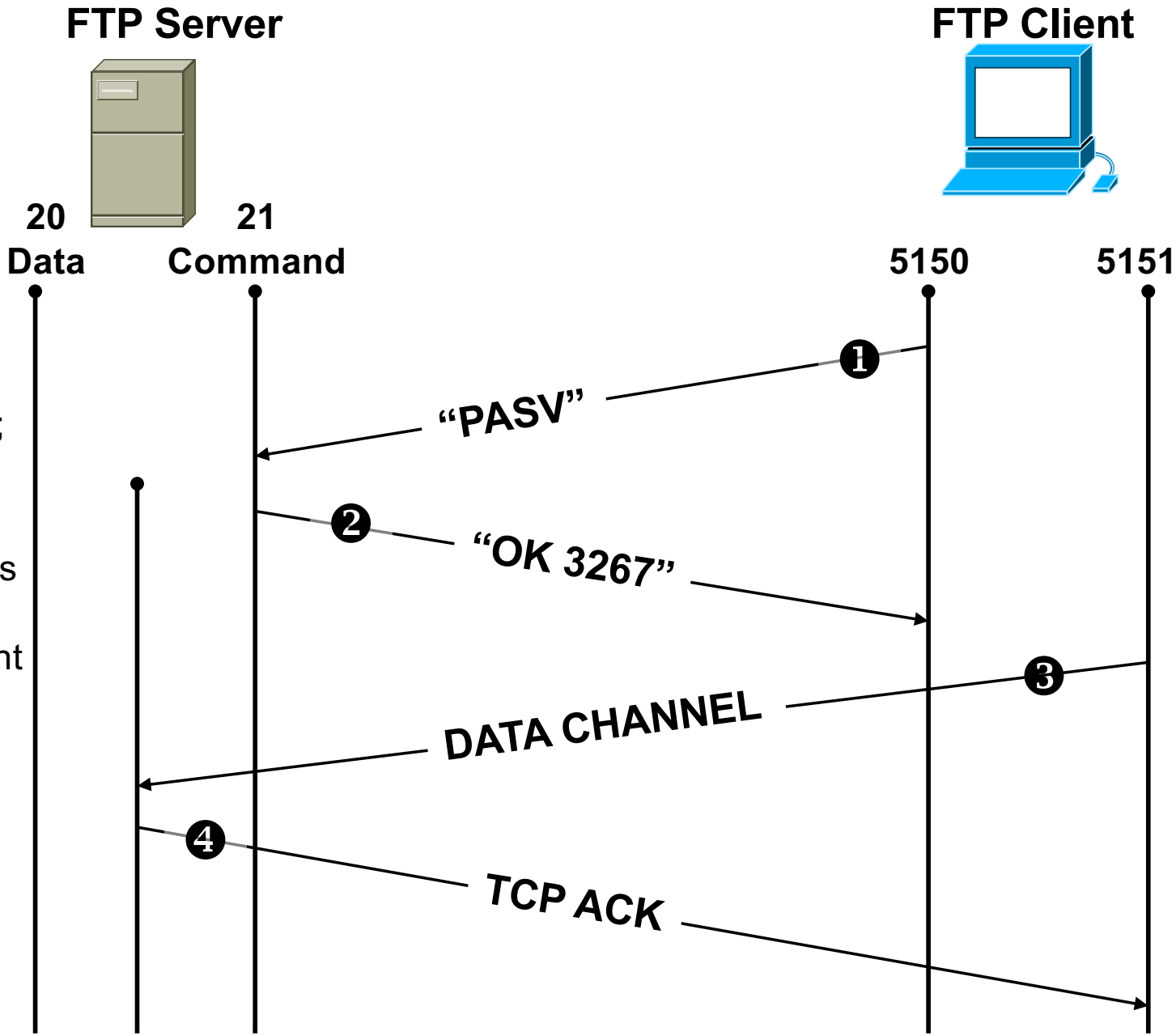
- Packet decision made in the context of a connection
- If packet is a new connection, check against security policy
- If packet is part of an existing connection, match it up in the state table & update table
- Typical Configuration
  - All denied unless specifically allowed
  - Dynamic protocols (FTP, RealAudio, etc.) allowed only if supported

Session Filtering

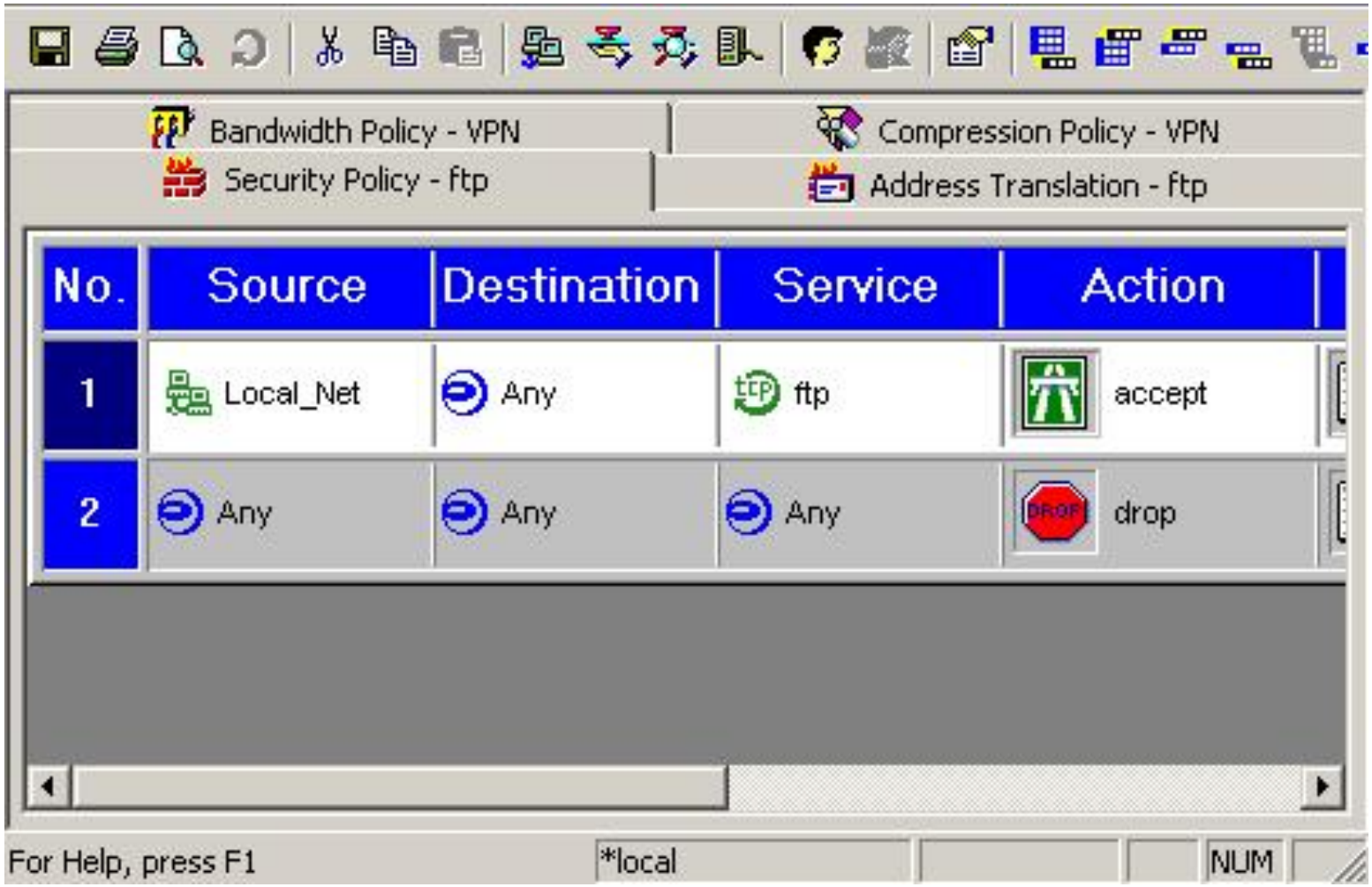
- **Screens** ALL attempts, **Protects** All applications
- **Extracts** & **maintains** 'state' information
- **Makes** an intelligent **security** / **traffic** decision



FTP – Passive Mode



Example FTP : Session Filter



PROXY FIREWALL

Proxy Firewalls

- Relay for connections
- Client ↔Proxy ↔Server
- Two flavors
  - Application level
  - Circuit level

Application Level Gateways

- Understands specific applications
  - Limited proxies available
  - Proxy ‘impersonates’ both sides of connection
- Resource intensive
  - process per connection
- HTTP proxies may cache web pages

Application Level Gateways

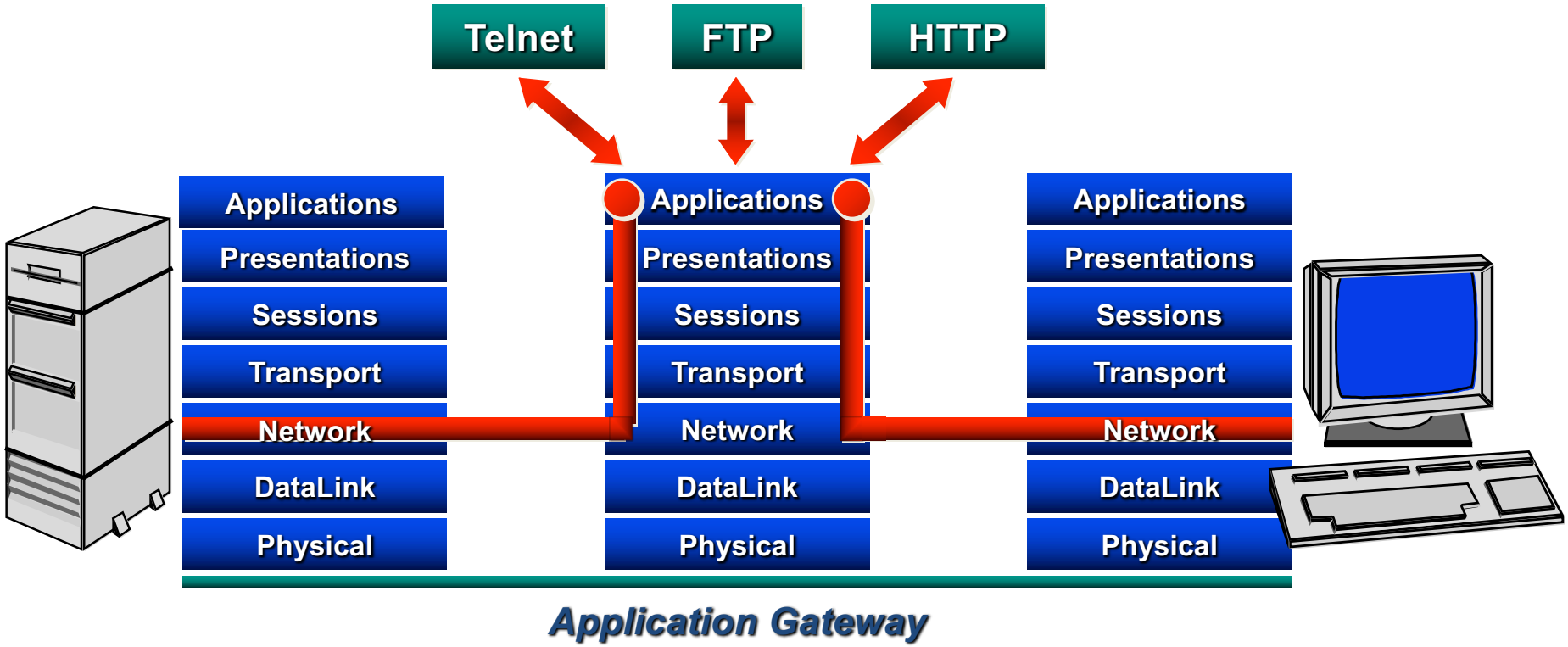
- Also called a Proxy Firewall
- Acts as a relay for application level traffic
  - Typical applications:
    - Telnet
    - FTP
    - SMTP
    - HTTP
- More secure than packet filters
  - Bad packets won't get through the gateway
  - Only has to deal with application level packets
- Simplifies rules needed in packet filter

Application Level Gateways

- Client connects
- Gateway does in depth inspection of the application level packet, if connection meets criteria on the gateway rule base, packet will be proxied to the server
- Proxy firewall is directly between the client and the server on an application by application basis

Application Gateways

- Clients configured for proxy communication
- Transparent Proxies



Application Level Gateways

- More appropriate to TCP
- ICMP difficult
- *Block all unless specifically allowed*
- Must write a new proxy application to support new protocols
  - Not trivial!

ALG Use

- Many application clients can be configured to use a specific ALG (proxy) by the end user
  - Firefox-Options-Advanced-Network-Connections-Proxy
  - WS/FTP-Connect-Firewall-Proxy
- Router can be set to forward all application packets to specific proxy
  - Benefit is all user traffic is forced to a proxy
  - User cannot bypass the proxy

Additional ALG Benefits

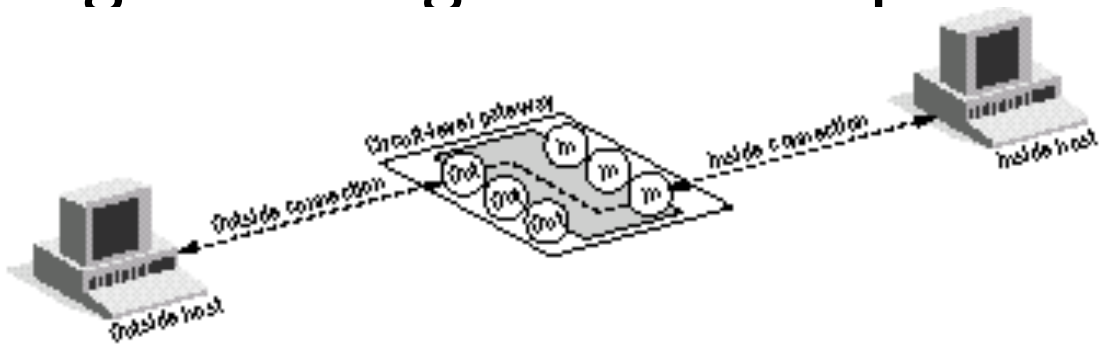
- Privacy
  - Outside world only sees the IP of the gateway not the IPs of the end users
  - Prevents foreign hosts from harvesting user addresses for later use in SPAM
    - Especially important for HTTP
- Ideal place to do logging

Circuit Level Gateways

- Also known as a Stateful Inspection Firewall
- Session layer of OSI
- Shim between transport and application layer of TCP/IP
- Monitors handshake used to establish connections
- Hides information about internal network
- Breaks the TCP connection
  - Proxies the TCP connection

Circuit-Level Gateways

- Support more services than Application-level Gateway
  - less control over data
- Hard to handle protocols like FTP
- Clients must be aware they are using a circuit-level proxy
- Protect against fragmentation problem



Example: SOCKS

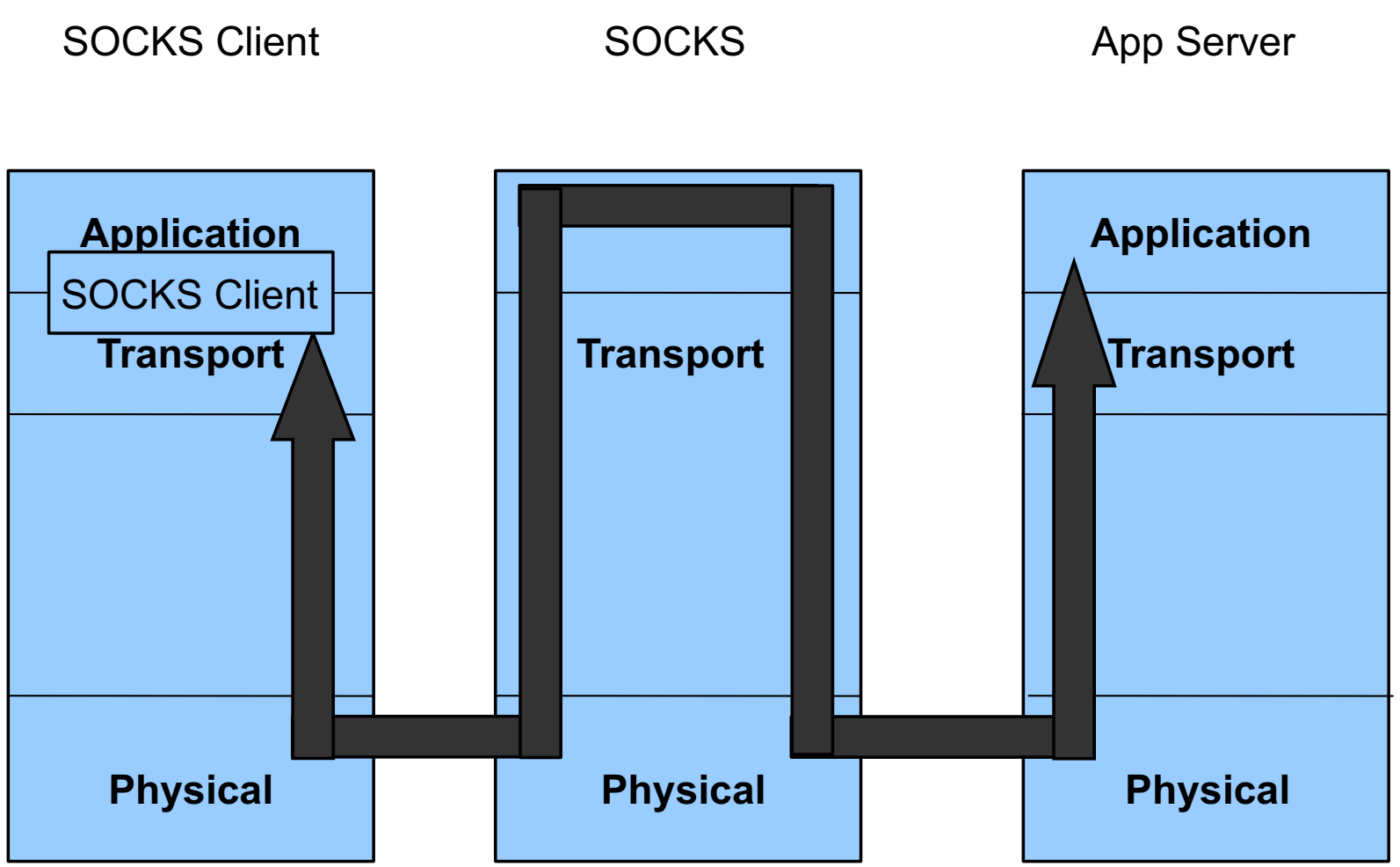
- Circuit level Gateway
- Support TCP
- SOCKS v5 supports UDP, earlier versions did not
- See http://www.socks.nec.com

SOCKS (SOCKetS)

- RFC1928
- Generic proxy protocol for TCP/IP
- Provides a framework for developing secure communications by easily integrating other security technologies
- Works for both TCP and UDP (ver. 5)

How Does SOCKS Work

- Client wants to connect to an application server
- Connects to SOCKS proxy using SOCKS protocol
- SOCKS proxy connects to application server using SOCKS protocol
- To the application server the SOCKS server is the client


SOCKS Client
SOCKS
App Server
Application
SOCKS Client
Transport
Physical
Transport
Physical
Application
Transport
Physical

The SOCKS Protocol

- SOCKS ver 5 IETF Approved (RFC 1928)
- Two components
  - Client – sits between the Application and Transport layers
  - Server – application layer
- Purpose is to enable a client on one side of the SOCKS server to talk to a server on the other side without requiring IP reachability

SOCKS Functions

- Make Connection Requests
- Set up proxy circuits
- Relay Application Data
- Perform user authentication

SOCKS Features

- Transparent network access across multiple proxy servers
- Easy deployment of authentication and encryption
- Rapid deployment of new network applications
- Simple network security policy management

SOCKS Benefits

- Single protocol authenticates and establishes the communication channel
- Is application independent
- Can be used with TCP or UDP
  - Supports redirection of ICMP
- Bi-directional support and intrinsic NAT for added security and anti-spoofing

Comparison

	Security	Performance
Packet Filter	3	1
Session Filter	2	2
Circuit GW	2	3
App. GW	1	4

Lower is better for security & performance

Comparison

	Modify Client Applications?
Packet Filter	No
Session Filter	No
Circuit GW	Typical, SOCKS-ify client applications
App. GW	Unless transparent, client application must be proxy-aware & configured

Comparison

	ICMP	Fragmen-tation
Packet Filter	Yes	No
Session Filter	Yes	Maybe
Circuit GW	(SOCKS v5)	Yes
App. GW	No	yes

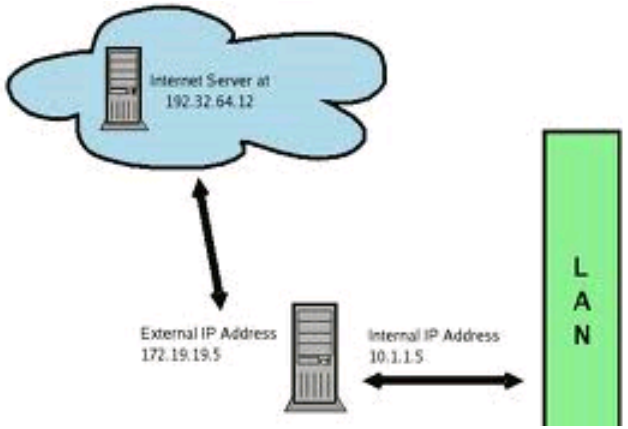
Proxying UDP/ICMP

- Why isn't UDP or ICMP proxied as much as TCP?
- TCP's connection-oriented nature easier to proxy
- UDP & ICMP harder (but not impossible) since each packet is a separate transaction
- Session filters determine which packets appear to be replies

NETWORK ADDRESS TRANSLATION

NAT: Network Address Translation

- Useful if organization does not have enough real IP addresses
- Extra security measure if internal hosts do not have valid IP addresses (harder to trick firewall)
- Only really need real IP addresses for services outside networks will originate connections to



NAT

- Many-to-1 (n-to-m) mapping
- 1-to-1 (n-to-n) mapping
- Proxies provide many-to-1
- NAT not required on filtering firewalls

Encryption (VPNs)

- Allows trusted users to access sensitive information while traversing untrusted networks
- Useful for remote users/sites
- IPSec