# CS489/698
# Privacy, Cryptography, Network and Data Security

Network Security Primer

Spring 2024, Monday/Wednesday 11:30am-12:50pm

# Today's Class

- Networking background
  - Internet's architecture
  - UDP/TCP
  - IP

- How to protect the network at the system level
  - Barriers to protect entry
    - Firewalls
    - DMZ
  - Detection systems to support these barriers
    - Honeypots
    - Intrusion Detection

# Architecture of the Internet
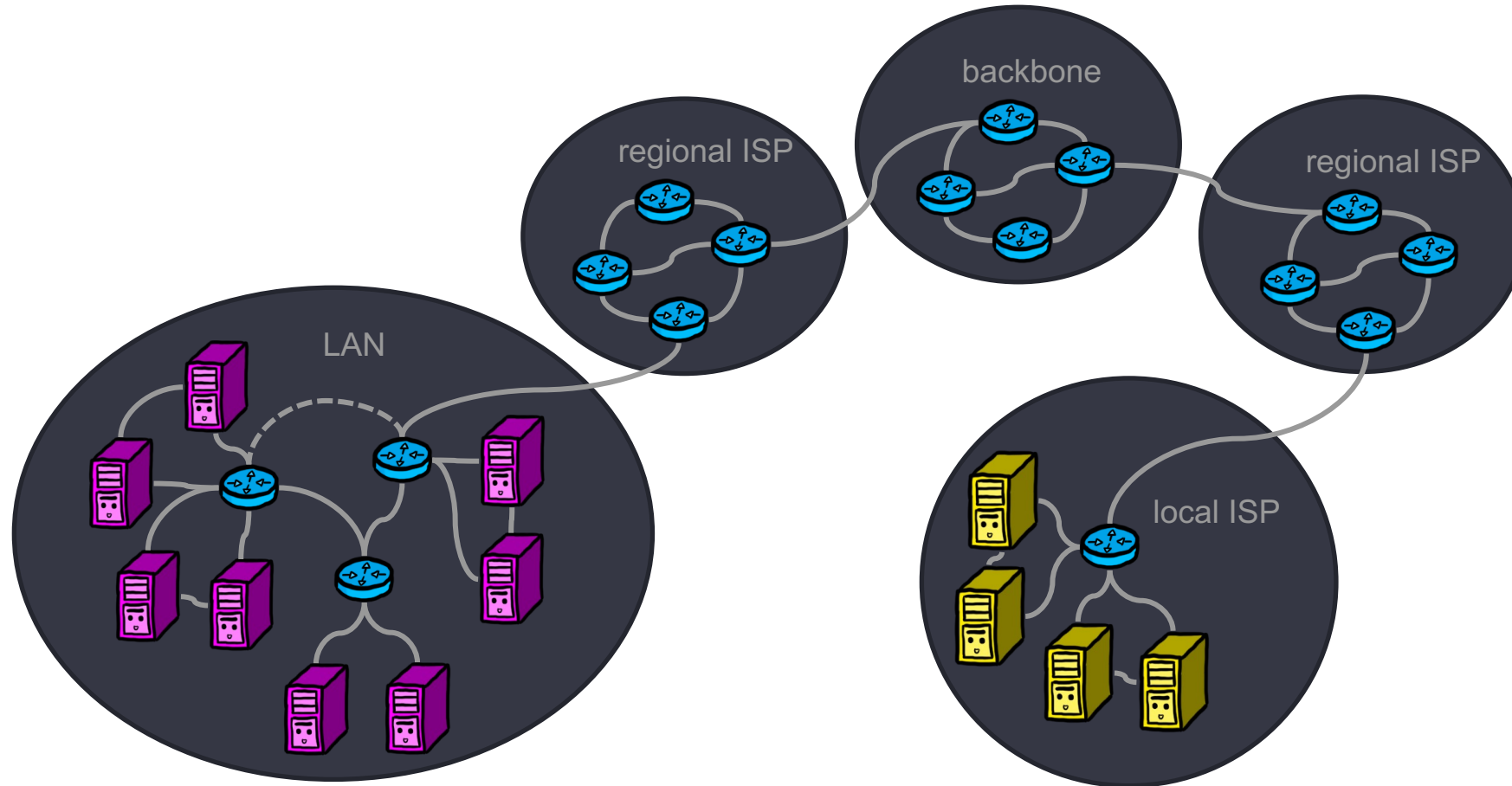
# Characteristics of the Internet (that make security hard)

- Traffic from a source to a destination flows through nodes controlled by different entities
- End nodes cannot control through which nodes traffic flows
  - Worse, all traffic is split up into individual packets, and each packet could be routed along a different path
- Different types of nodes
  - Server, laptop, router, UNIX, Windows,. . .
- Different types of communication links
  - Wireless vs. wired
- TCP/IP suite of protocols
  - Packet format, routing of packets, dealing with packet loss,. . .
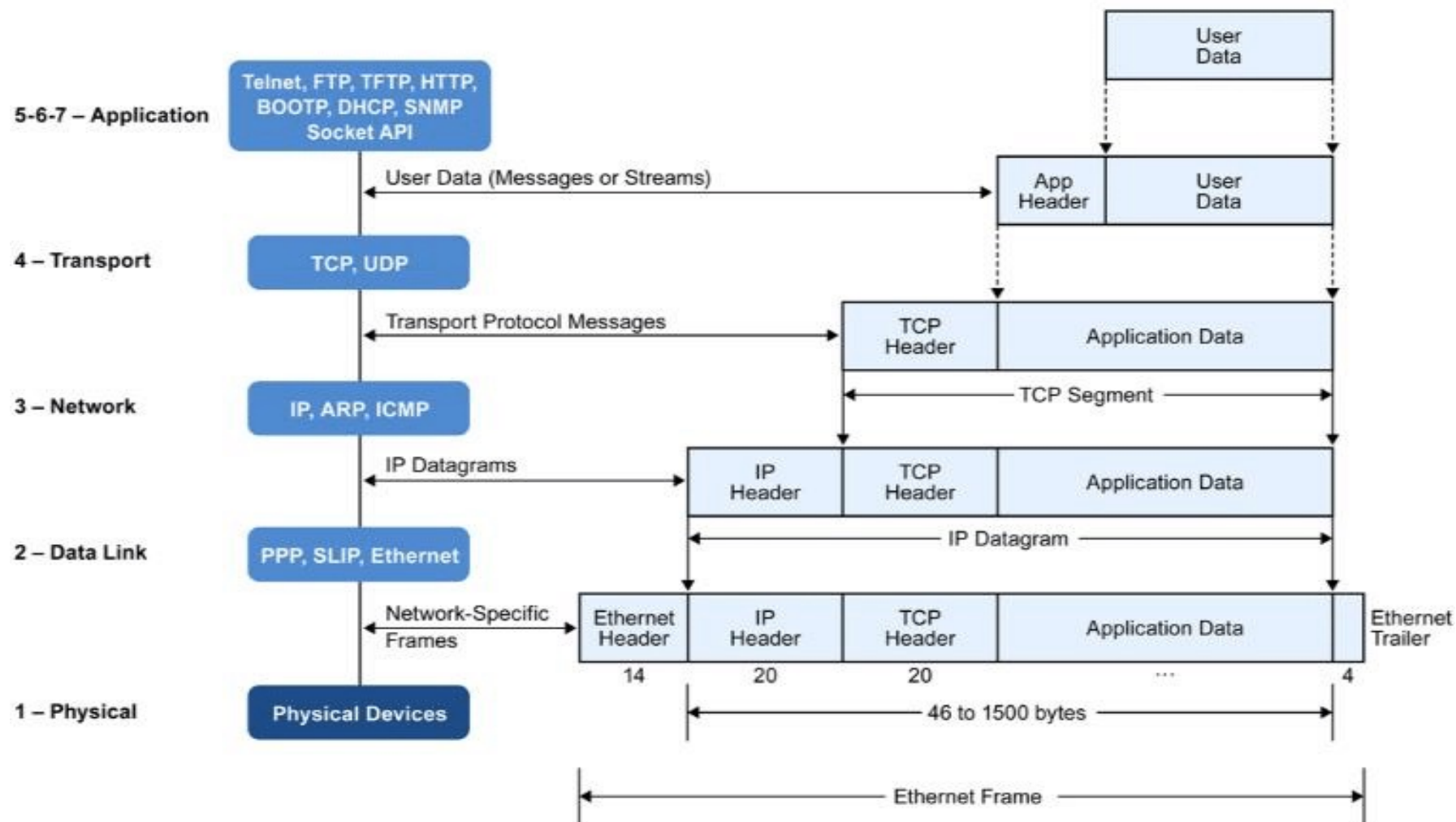
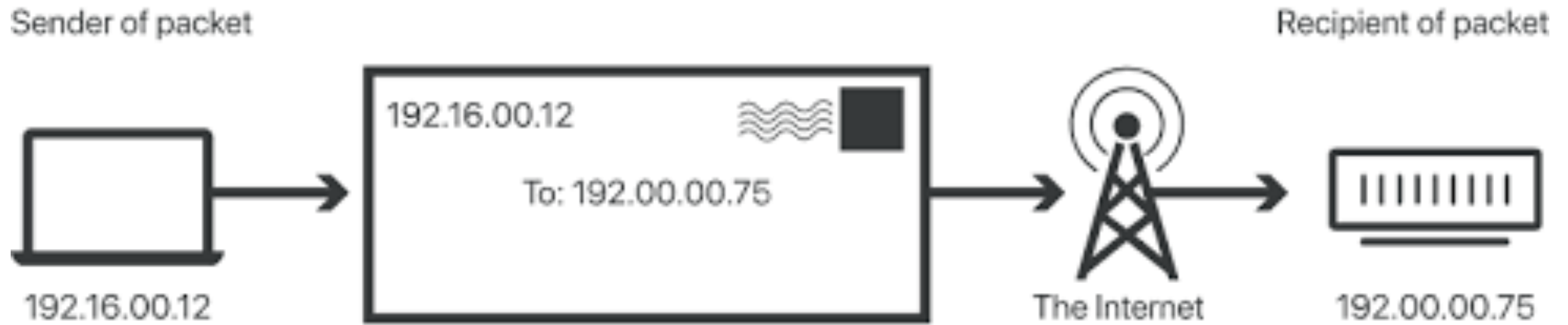# How can we be safe?
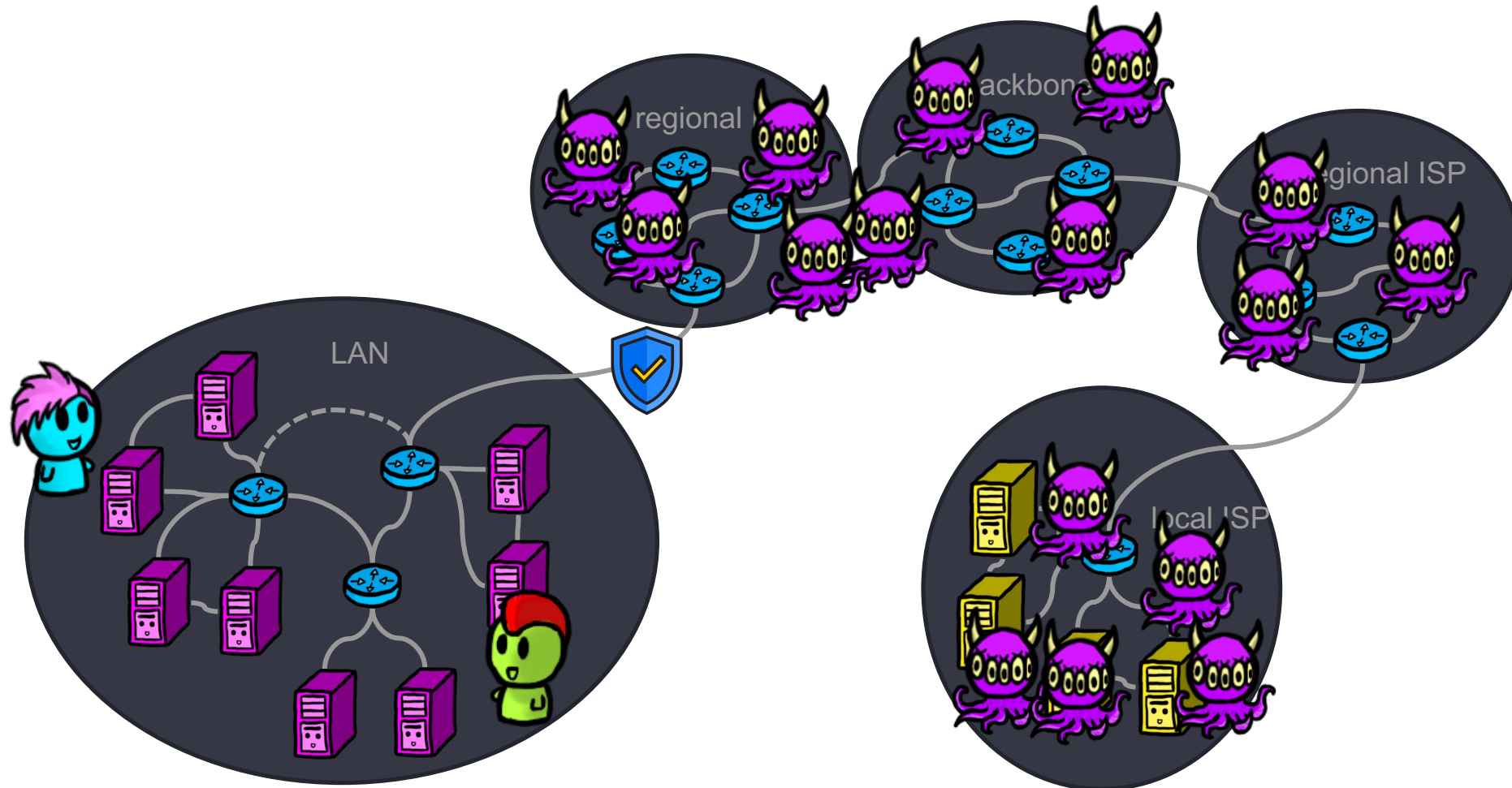
# Networking Basics

# TCP/IP Model
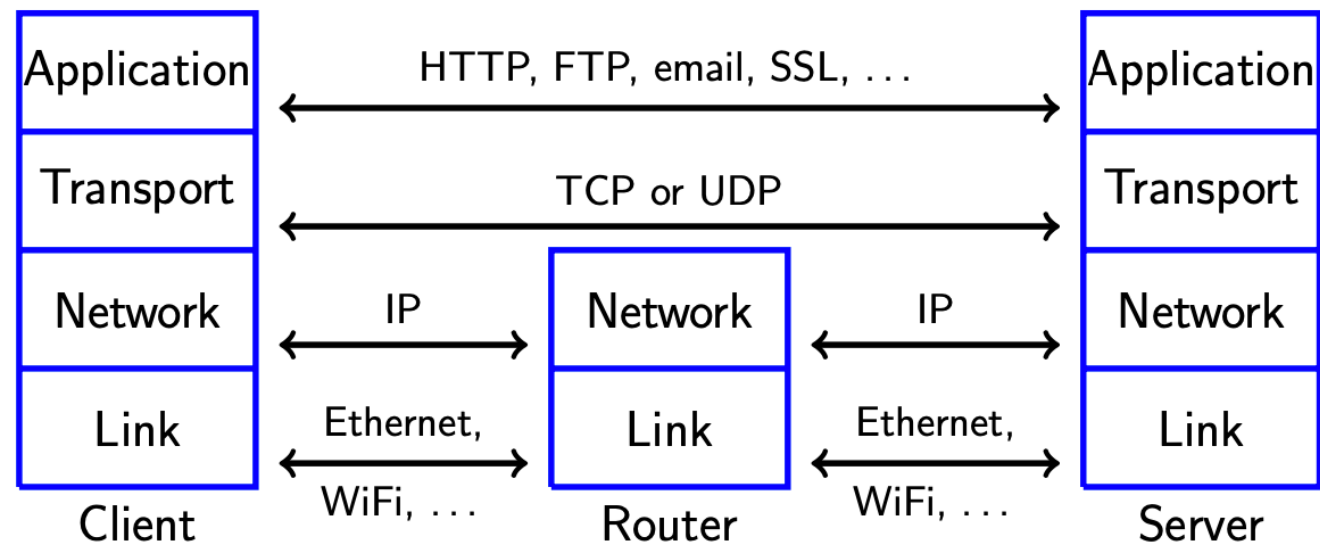


- Transport and network layer designed in the 1970s to connect local networks at different universities and research labs
- Participants knew and **trusted** each other
- Design addressed **non-malicious errors** (e.g., packet drops), but not malicious errors

# Packet Formats

# IP

# UDP vs TCP



**TCP**
- Slower but more reliable transfers
- Typical Applications:
  - File Transfer Protocol (FTP)
  - Web Browsing
  - Email

unicast

**UDP**
- Faster but not guaranteed transfers ("best effort")
- Typical Applications:
  - Live Streaming
  - Online Games
  - VoIP

unicast    multicast    broadcast

# Sockets
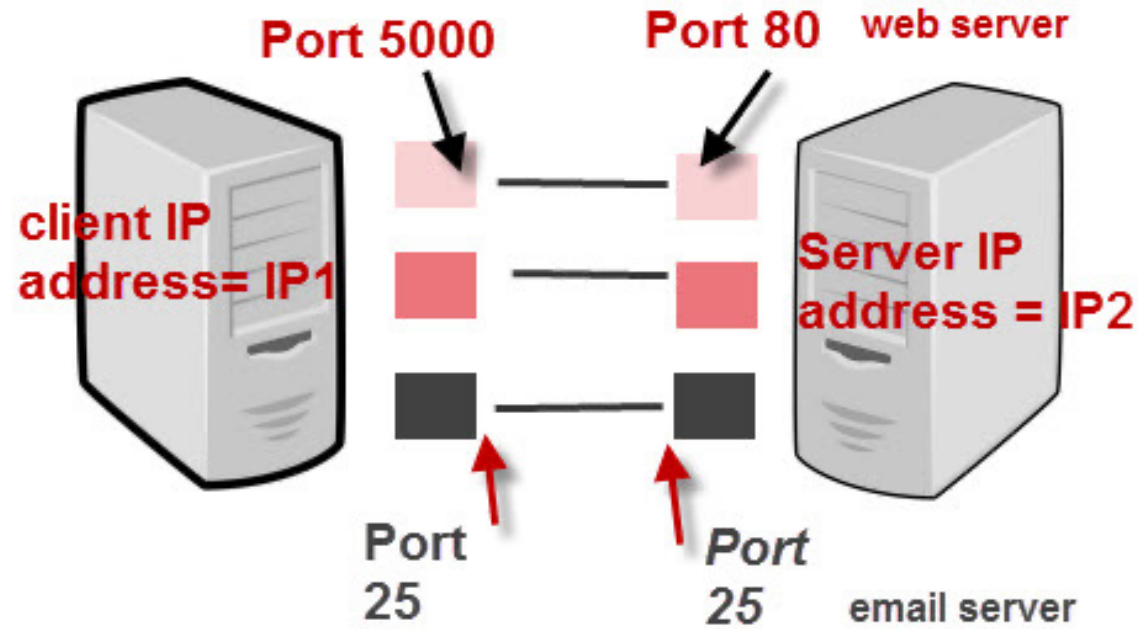


IP Address + Port number = Socket

**TCP/IP Ports And Sockets**

# The TCP Handshake

IP: 129.97.124.21
Port: 5297

Alice

Bob

IP: 142.251.41.46
Port: 443

# The TCP Handshake

IP: 129.97.124.21
Port: 5297

Alice

Bob

IP: 142.251.41.46
Port: 443

SrcIP: 129.97.124.21    TCPSrc: 5297
DstIP: 142.251.41.46    TCPDst: 443

SYN

# The TCP Handshake



IP: 129.97.124.21
Port: 5297

**Alice**

**Bob**

IP: 142.251.41.46
Port: 443

SrcIP: 129.97.124.21    TCPSrc: 5297
DstIP: 142.251.41.46    TCPDst: 443

SYN

SrcIP: 142.251.41.46    TCPSrc: 443
DstIP: 129.97.124.21    TCPDst: 5297

SYN/ACK
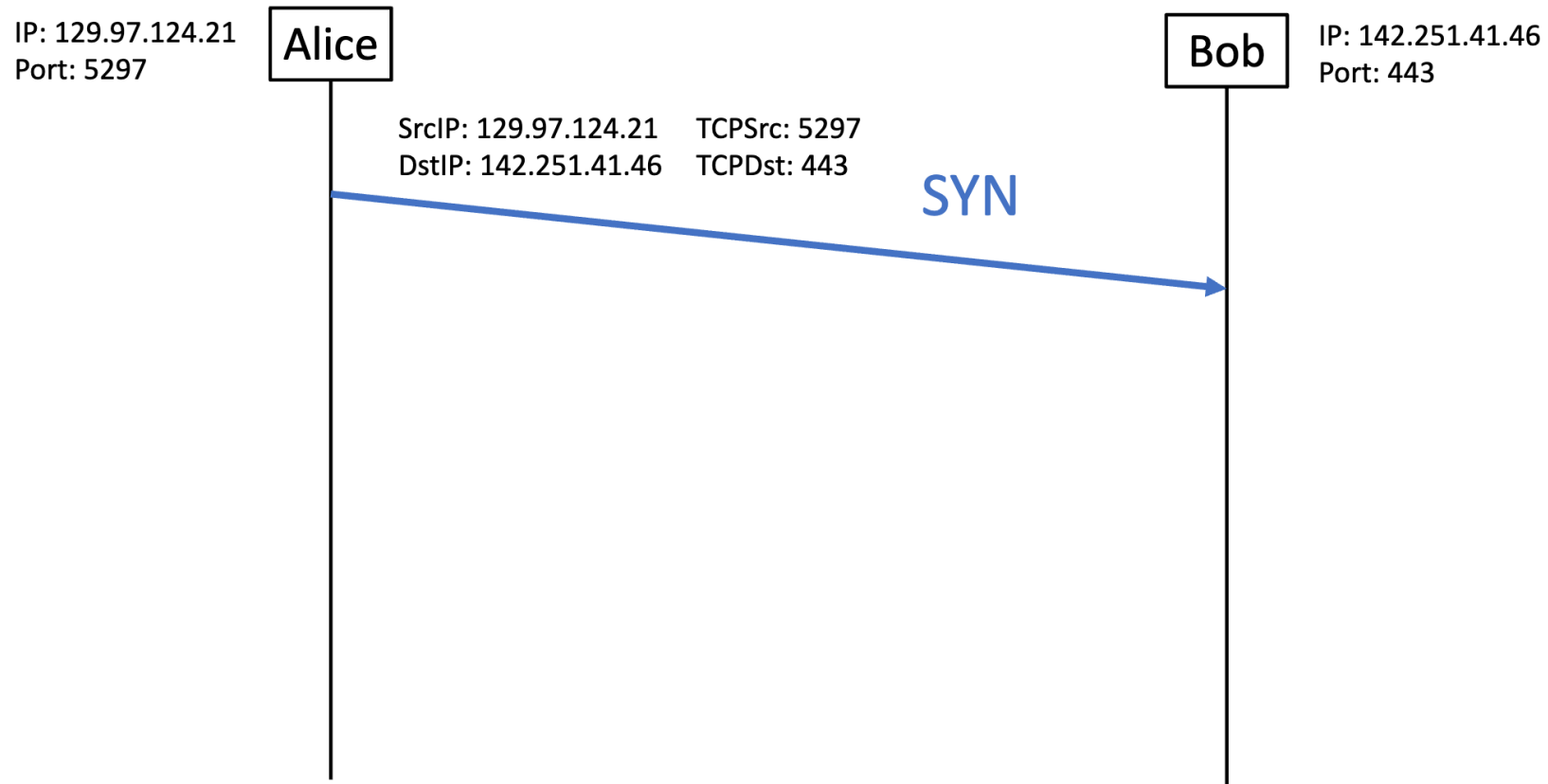
# The TCP Handshake

# The TCP Handshake

IP: 129.97.124.21
Port: 5297

**Alice**

**Bob**

IP: 142.251.41.46
Port: 443

SrcIP: 129.97.124.21     TCPSrc: 5297
DstIP: 142.251.41.46     TCPDst: 443

SYN

SrcIP: 142.251.41.46     TCPSrc: 443
DstIP: 129.97.124.21     TCPDst: 5297

SYN/ACK

SrcIP: 129.97.124.21     TCPSrc: 5297
DstIP: 142.251.41.46     TCPDst: 443

ACK
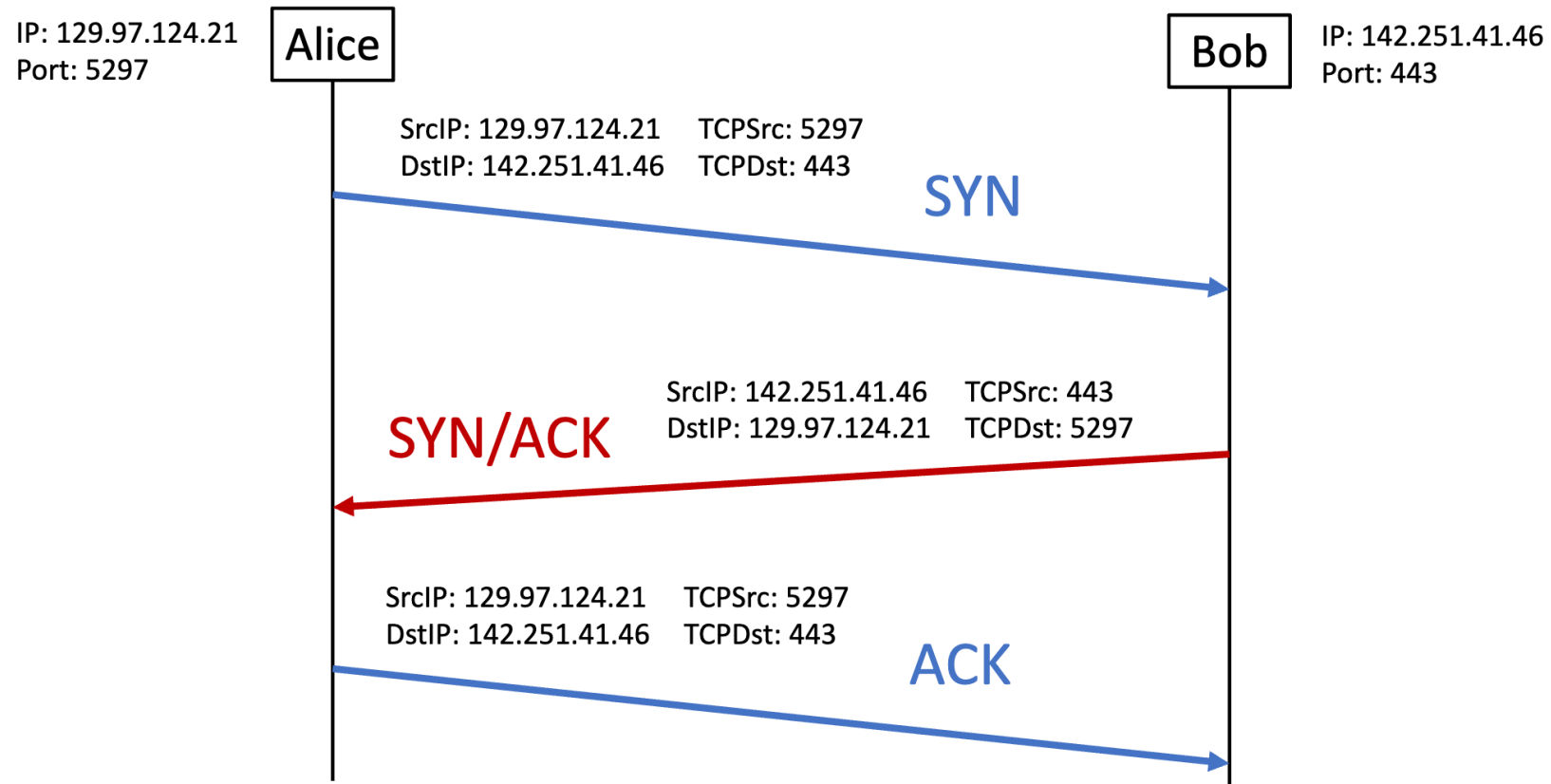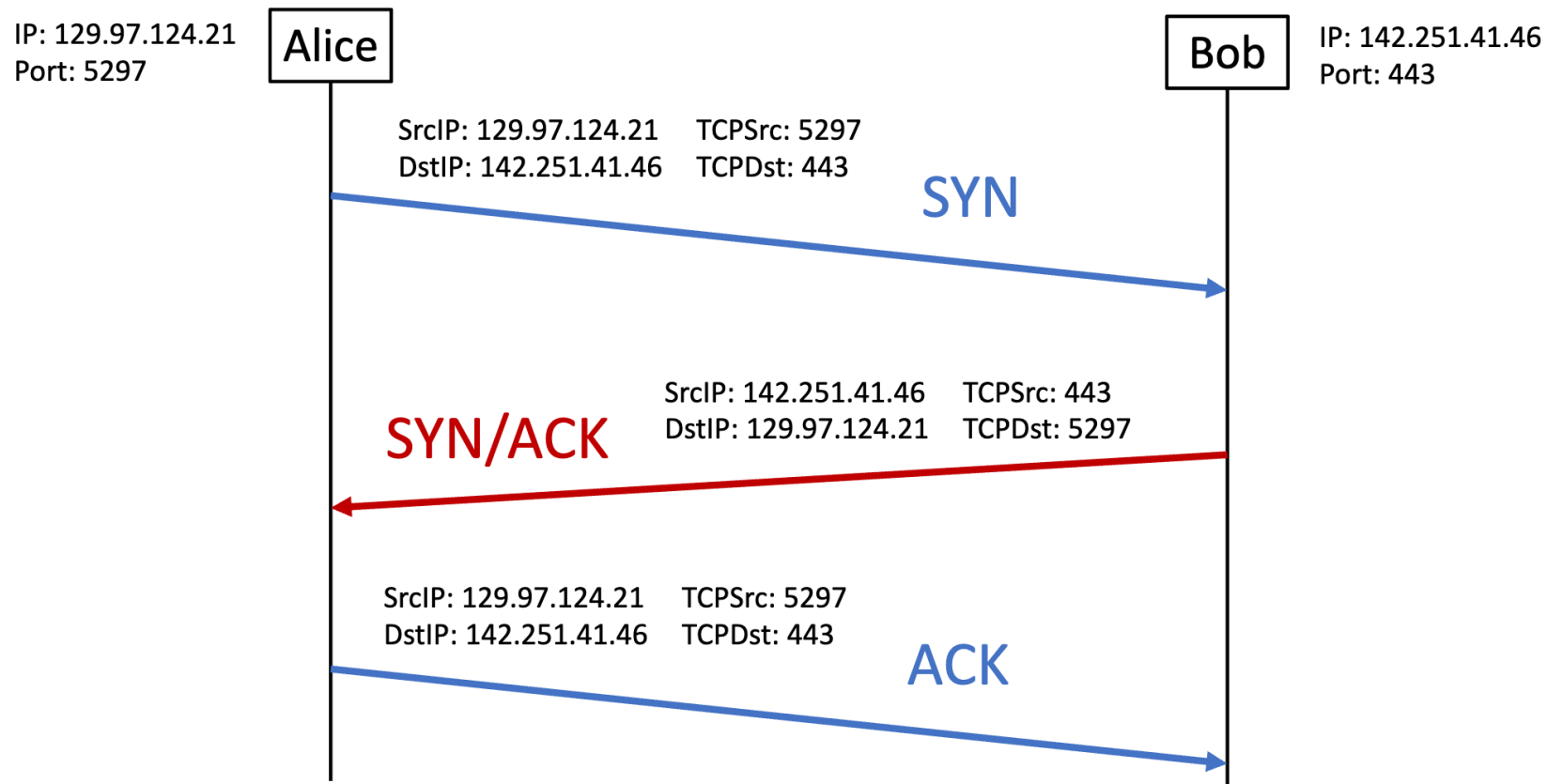
Disclaimer: omitting several other details (SEQ numbers, MSS, window size, additional flags, etc.)

# Working Example



regional ISP

mail server

mail server

Alice

Bob
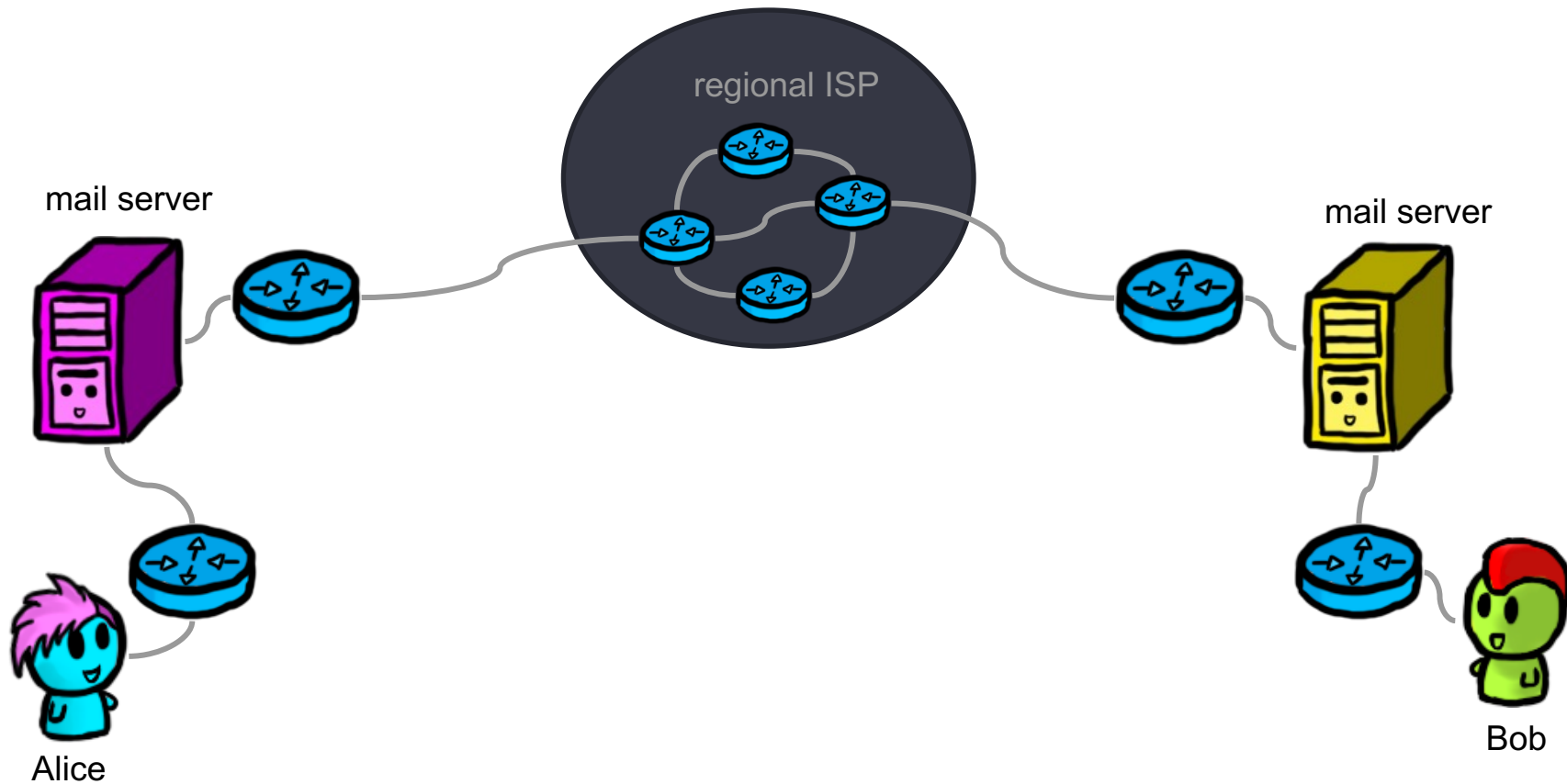
# Keep Unauthorized Individuals Out of the LAN



mail server

mail server

Alice

Bob

regional ISP
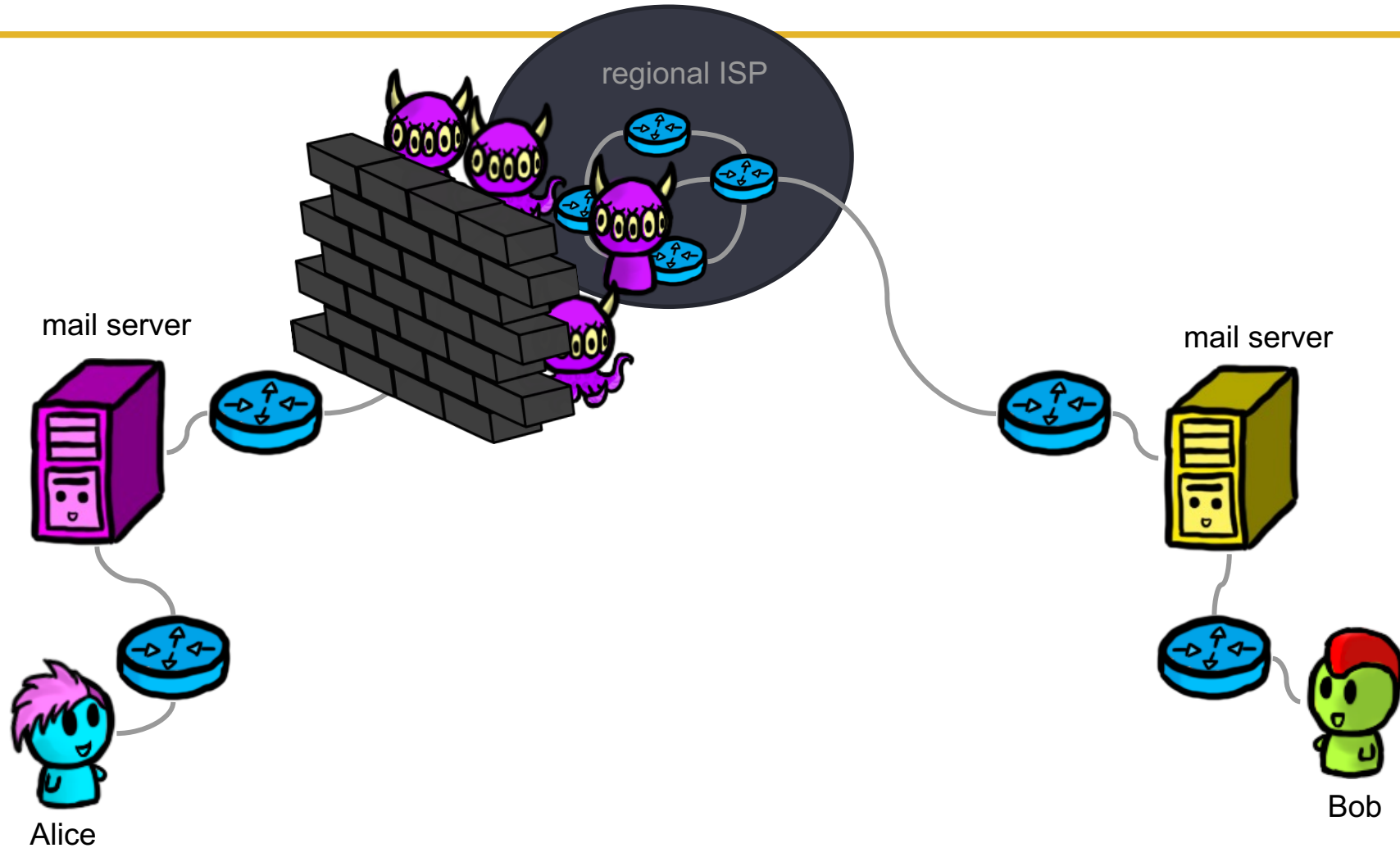
# Redundancy (in case we fail)

- Avoid **single points of failure**
  - Even if you don't have to worry about attackers
  - Disk crash, power failure, earthquake...

- Servers should be deployed in a **redundant** way on multiple machines, ideally with different software to get **genetic diversity** and at **different locations**

- Redundant servers should be kept in (close) sync so that backup servers can take over easily
  - Test this! Beware of Schrödinger backups

# Access controls

- Access Control Lists (ACLs) on routers
  - All traffic to a company typically goes through a single (or a few) routers
  - Filter traffic based on allow/block list
  - In case of flooding attack, define router ACL that drops packets with particular source and destination address
  - ACLs are expensive for high-traffic routers
  - Source addresses of packets in flood are typically spoofed and dynamic

- Need something more sophisticated.

# Firewalls

# Firewalls



mail server

mail server

Alice

Bob

regional ISP
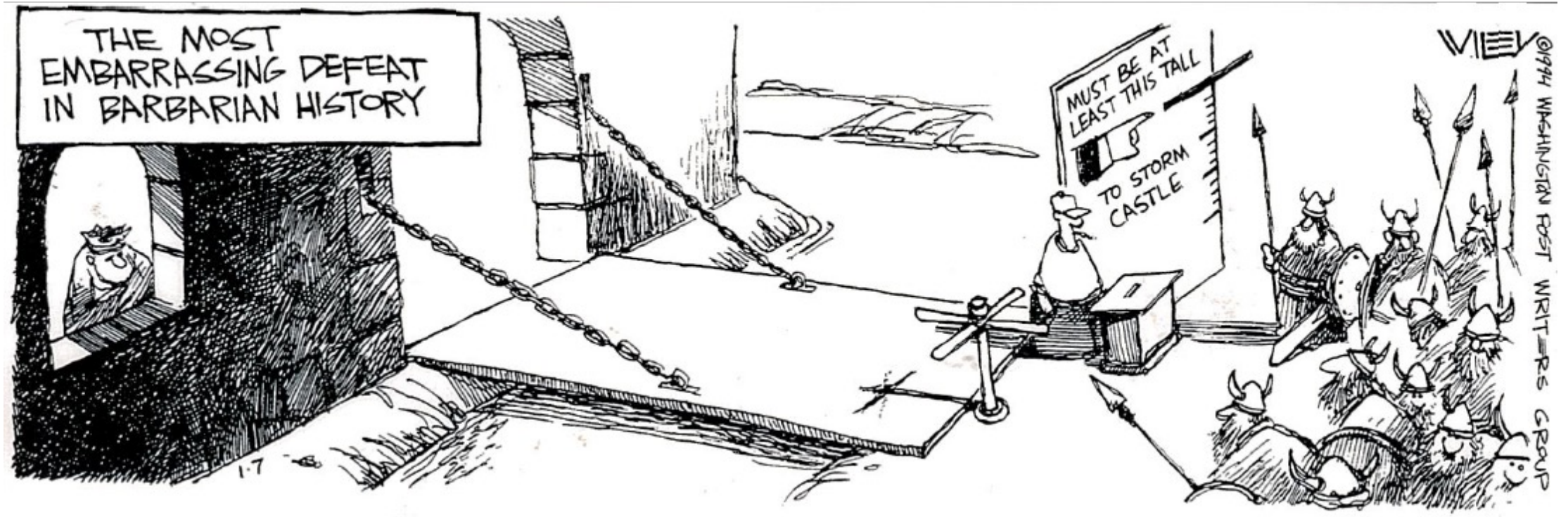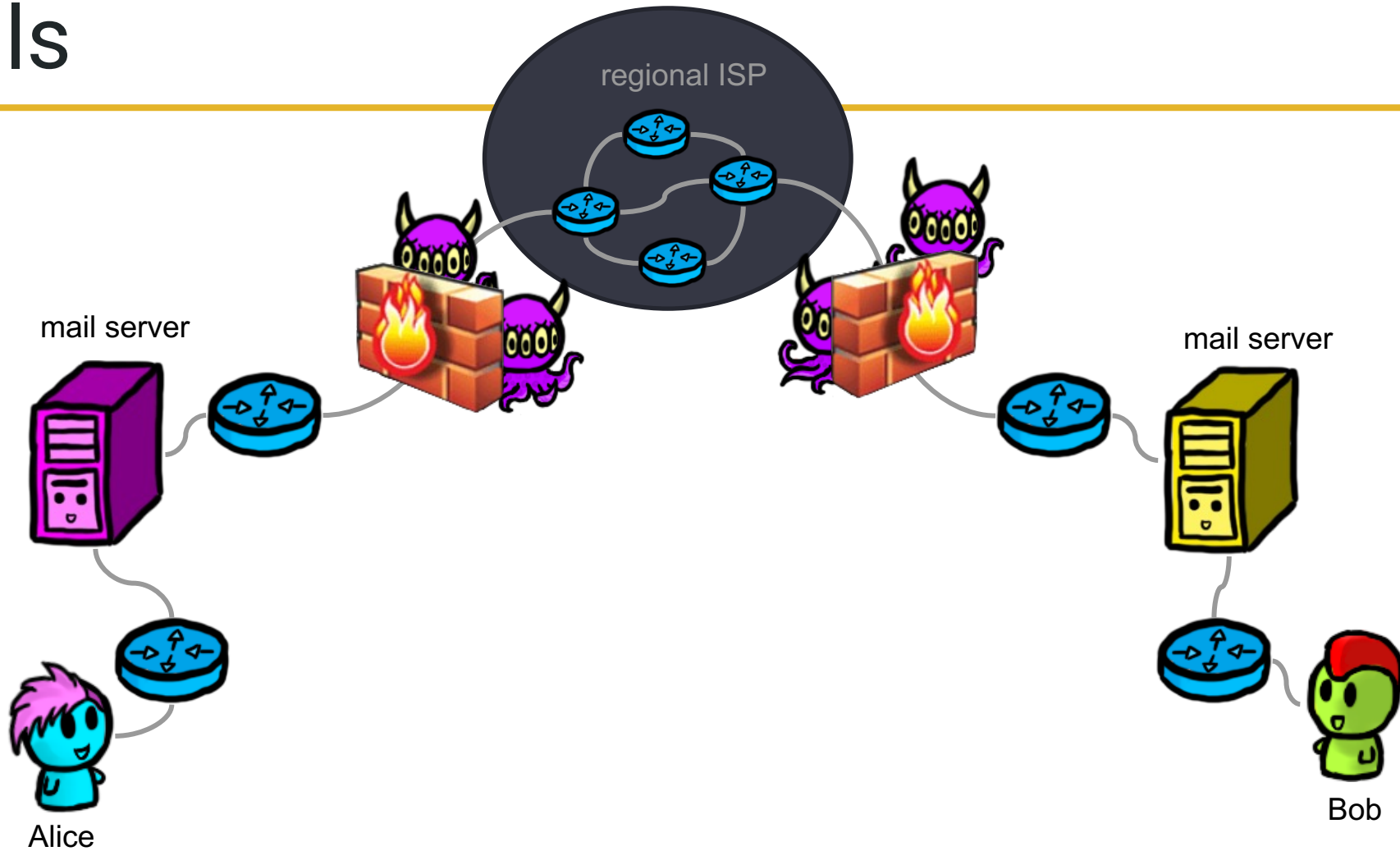
# Firewalls

- Firewalls are the castles of the Internet age

- All traffic into/out of a company has to go through a small number of gates (choke points)
  - Wireless access point should be outside of firewall

- Choke points carefully examine traffic, especially incoming, and might refuse it access
  - Two strategies: "permit everything unless explicitly forbidden" or "forbid everything unless explicitly allowed"

# Blocklist vs allowlist

# Limitations of Firewalls

- Firewalls do not protect against internal attacks
  - Employee who colludes with external party
  - Laptop that is infected while outside the LAN
  - Poorly secured WIFI allowing an attacker inside the LAN.

- Vulnerable to IP Spoofing (more later)

- Need multiple layers of defense / defense in depth

# Types of firewalls

- Packet filtering gateways / screening routers
- Stateful inspection firewalls
- Application proxies
- Personal firewalls

- Firewalls are attractive targets for attackers; they are typically deployed on designated computers that have been stripped of all unnecessary functionality to limit attack surface

# Packet filtering gateways

- Simplest type – very fast and transparent

- Make decision based on <span style="color:red">network header of a packet</span>

- Header contains source and destination addresses and port numbers, port numbers can be used to infer type of packet
  - 80->Web,22->SSH
  - E.g., allow Web, but not SSH

- Ignore payload of packet

# Example

Table 12.1   Packet-Filtering Example

| Rule | Direction | Source Address | Destination Address | Protocol | Destination Port | Action |
|------|-----------|----------------|---------------------|----------|------------------|--------|
| A | In | External | Internal | TCP | 25 | Permit |
| B | Out | Internal | External | TCP | > 1023 | Permit |
| C | Out | Internal | External | TCP | 25 | Permit |
| D | In | External | Internal | TCP | > 1023 | Permit |
| E | Either | Any | Any | Any | Any | Deny |

# Example

Table 12.1   Packet-Filtering Example

| Rule | Direction | Source Address | Destination Address | Protocol | Destination Port | Action |
|------|-----------|----------------|---------------------|----------|------------------|--------|
| A | In | External | Internal | TCP | 25 | Permit |
| B | Out | Internal | External | TCP | > 1023 | Permit |
| C | Out | Internal | External | TCP | 25 | Permit |
| D | In | External | Internal | TCP | > 1023 | Permit |
| E | Either | Any | Any | Any | Any | Deny |

**Q:** Problems?

# Example

Table 12.1    Packet-Filtering Example

| Rule | Direction | Source Address | Destination Address | Protocol | Destination Port | Action |
|------|-----------|----------------|---------------------|----------|------------------|--------|
| A | In | External | Internal | TCP | 25 | Permit |
| B | Out | Internal | External | TCP | > 1023 | Permit |
| C | Out | Internal | External | TCP | 25 | Permit |
| D | In | External | Internal | TCP | > 1023 | Permit |
| E | Either | Any | Any | Any | Any | Deny |

| Rule | Direction | Source Address | Source Port | Dest Address | Protocol | Dest Port | Flag | Action |
|------|-----------|----------------|-------------|--------------|----------|-----------|------|--------|
| D | In | External | 25 | Internal | TCP | > 1023 | ACK | Permit |

# What about spoofing

- Can drop spoofed traffic
  - uWaterloo's firewall could drop all packets originating from uWaterloo whose source address is not of the form 129.97.x.y
  - And traffic originating from outside of uWaterloo whose source address is of the form 129.97.x.y

**Q:** Does this eliminate spoofed traffic completely?

# What about spoofing

- Can drop spoofed traffic
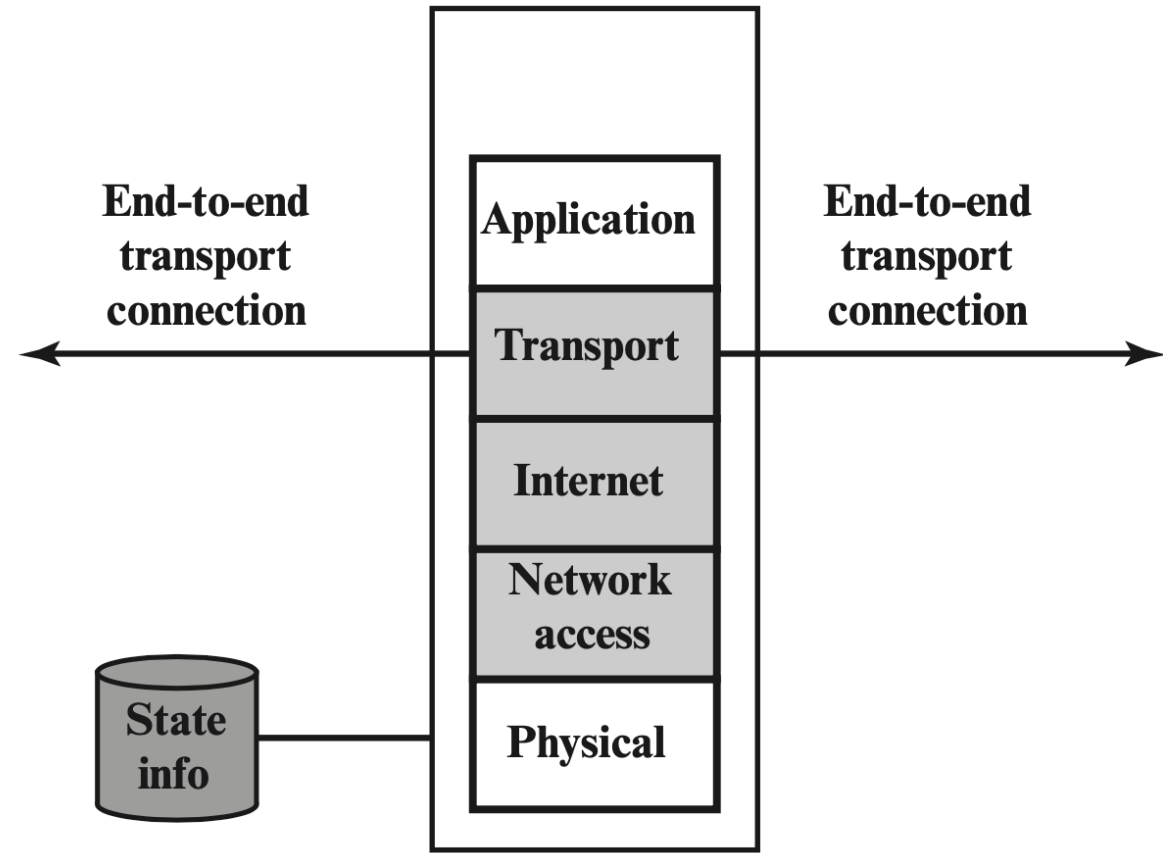  - uWaterloo's firewall could drop all packets originating from uWaterloo whose source address is not of the form 129.97.x.y
  - And traffic originating from outside of uWaterloo whose source address is of the form 129.97.x.y

**Q:** Does this eliminate spoofed traffic completely?

**A:** No. <u>Internal</u> spoofing may still occur.
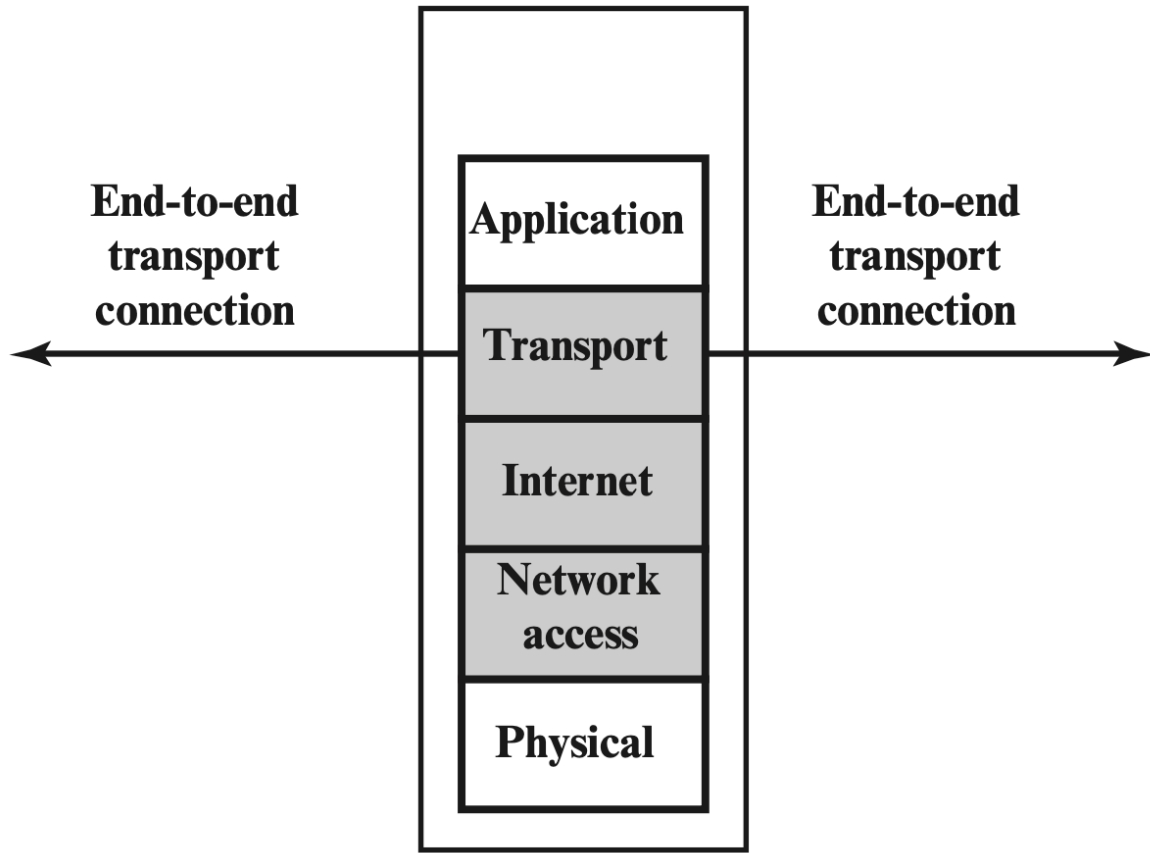
(b) Packet filtering firewall

(c) Stateful inspection firewall

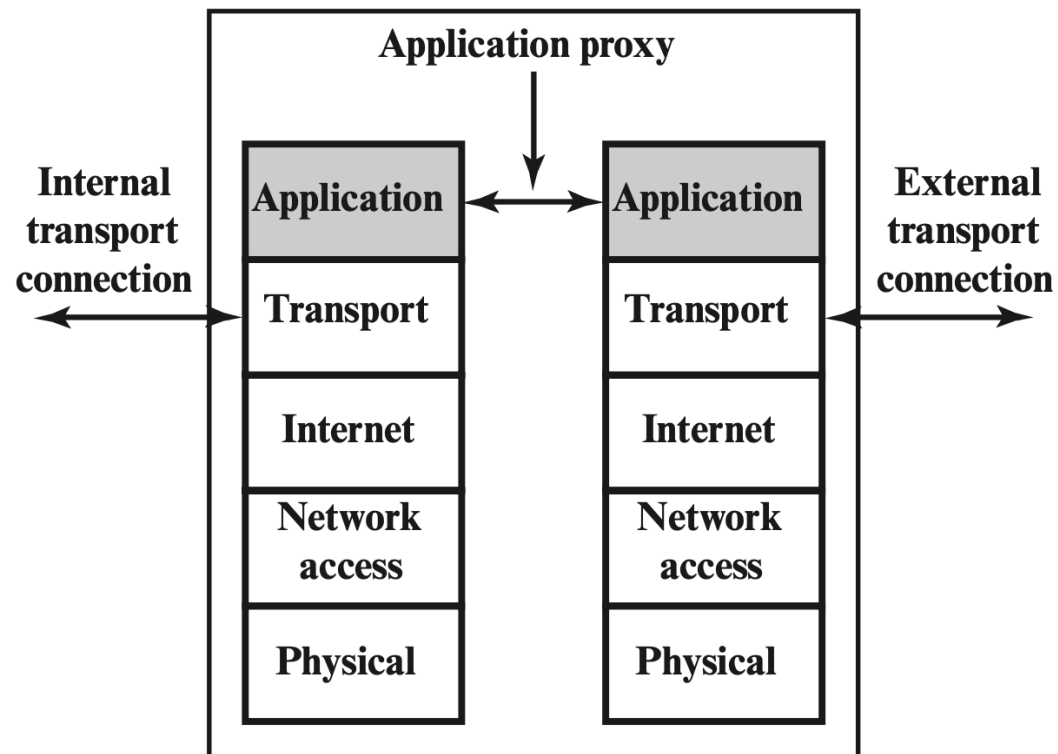# Stateful inspection firewalls

- More expensive than packet filtering

- Keep <span style="color:red">state</span> to identify packets that belong together
  - When a client within the company opens a TCP connection to a server outside the company, firewall must recognize response packets from server and let (only) them through
  - Some application-layer protocols (e.g., FTP) require additional (expensive) inspection of packet content to figure out what kind of traffic should be let through

- IP layer can fragment packets, so firewall might have to re-assemble packets for stateful inspection

**Table 12.2** Example Stateful Firewall Connection State Table (SP 800-41-1)

| Source Address | Source Port | Destination Address | Destination Port | Connection State |
|---|---|---|---|---|
| 192.168.1.100 | 1030 | 210.22.88.29 | 80 | Established |
| 192.168.1.102 | 1031 | 216.32.42.123 | 80 | Established |
| 192.168.1.101 | 1033 | 173.66.32.122 | 25 | Established |
| 192.168.1.106 | 1035 | 177.231.32.12 | 79 | Established |
| 223.43.21.231 | 1990 | 192.168.1.6 | 80 | Established |
| 2122.22.123.32 | 2112 | 192.168.1.6 | 80 | Established |
| 210.922.212.18 | 3321 | 192.168.1.6 | 80 | Established |
| 24.102.32.23 | 1025 | 192.168.1.6 | 80 | Established |
| 223.21.22.12 | 1046 | 192.168.1.6 | 80 | Established |

# Application Proxy

- **Client talks to proxy, proxy talks to server**

  - Specific for an application (email, Web,...)

  - Not as transparent as packet filtering or stateful inspection

  - Intercepting proxy requires no explicit configuration by client (or knowledge of this filtering by client)

  - All other traffic is blocked



**Application proxy**

Internal transport connection

External transport connection

| Application | Application |
| Transport | Transport |
| Internet | Internet |
| Network access | Network access |
| Physical | Physical |

**(d) Application proxy firewall**

# Application Proxy

- For users within the company wanting to access a server outside the company (forward proxy) and vice versa (reverse proxy)

- Proxy has full knowledge about communication and can do sophisticated processing
  - Limit types of allowed database queries, filter URLs, log all emails, scan for viruses

- Can also do strong user authentication

# Personal Firewalls



regional ISP

mail server

mail server

Alice

Bob
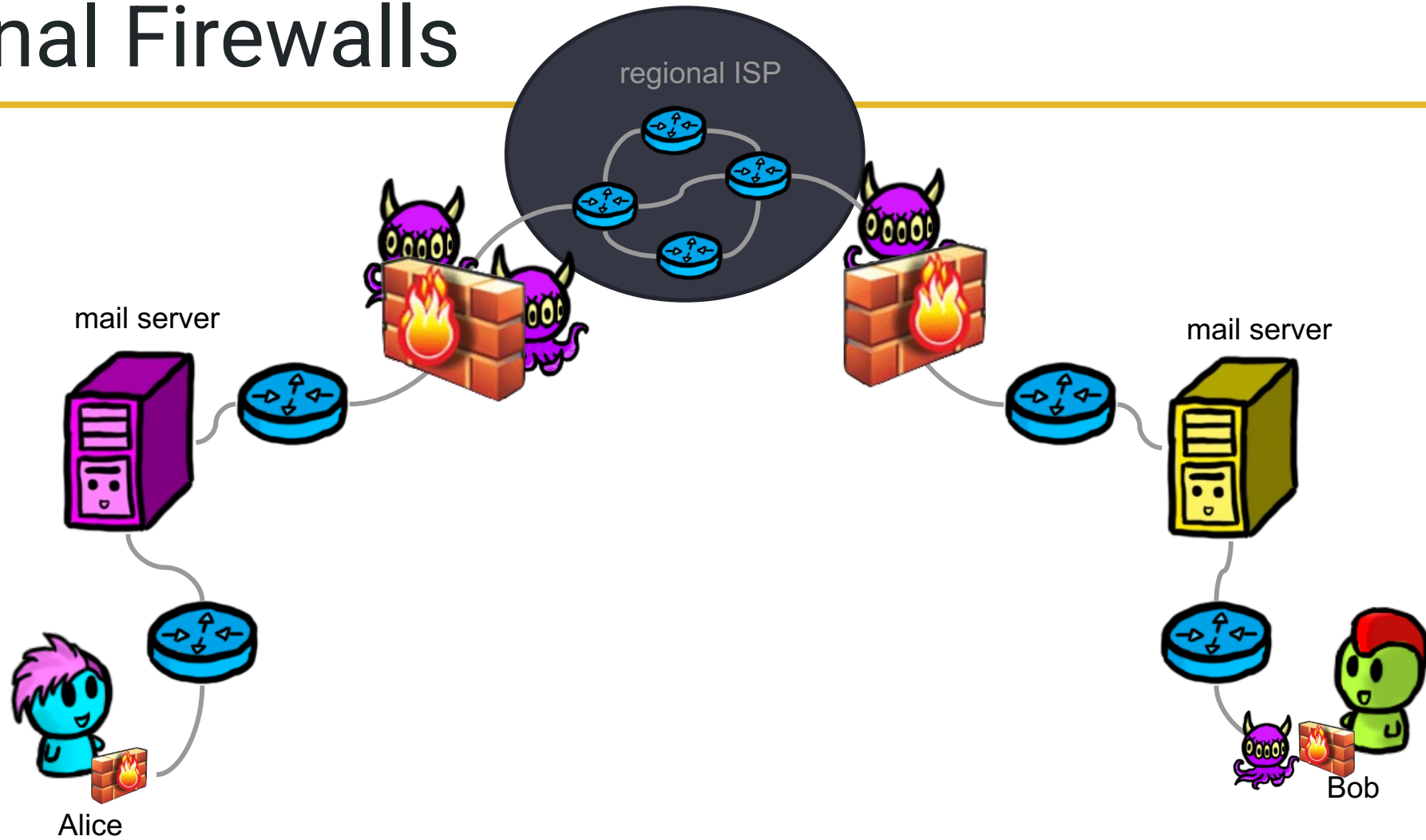
# Personal firewalls

- Firewall that runs on a (home) user's computer
    - Usually software based
    - Especially important for computers that are always online

- Primarily for denying unauthorized remote access

- Typically "forbid everything unless explicitly allowed"
    - Definitely for communication originating from other computers
    - Maybe also for communication originating on the user's computer
        - Why? What's the problem here?

# Segmentation and separation

- Don't put all a company's servers on a single machine
  - Deploy them on **multiple** machines, depending on their functional and access requirements

- If a machine gets broken into, only some services will be affected
  - E.g., the web server of a company needs to be accessible from the outside and is therefore more vulnerable

- Therefore, it shouldn't be trusted by other servers of the company, and it should be deployed outside the company firewall

# Demilitarized Zone (DMZ)

- Subnetwork that contains an organization's external services, accessible to the Internet
  - E.g. Webserver, email sever, DNS server
- Deploy external and internal firewall
  - External firewall protects DMZ
  - Internal firewall protects internal network from attacks lodged in DMZ

DMZ

Internal network

regional ISP

Internal Firewall

External Firewall
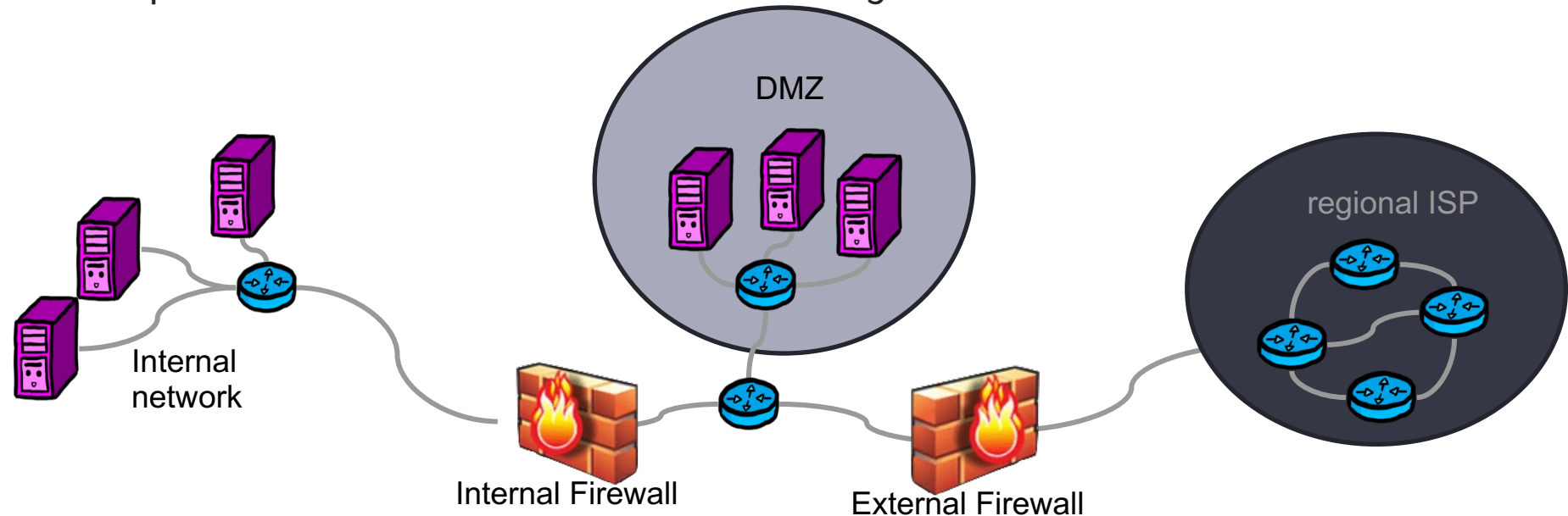
# DMZ in our example
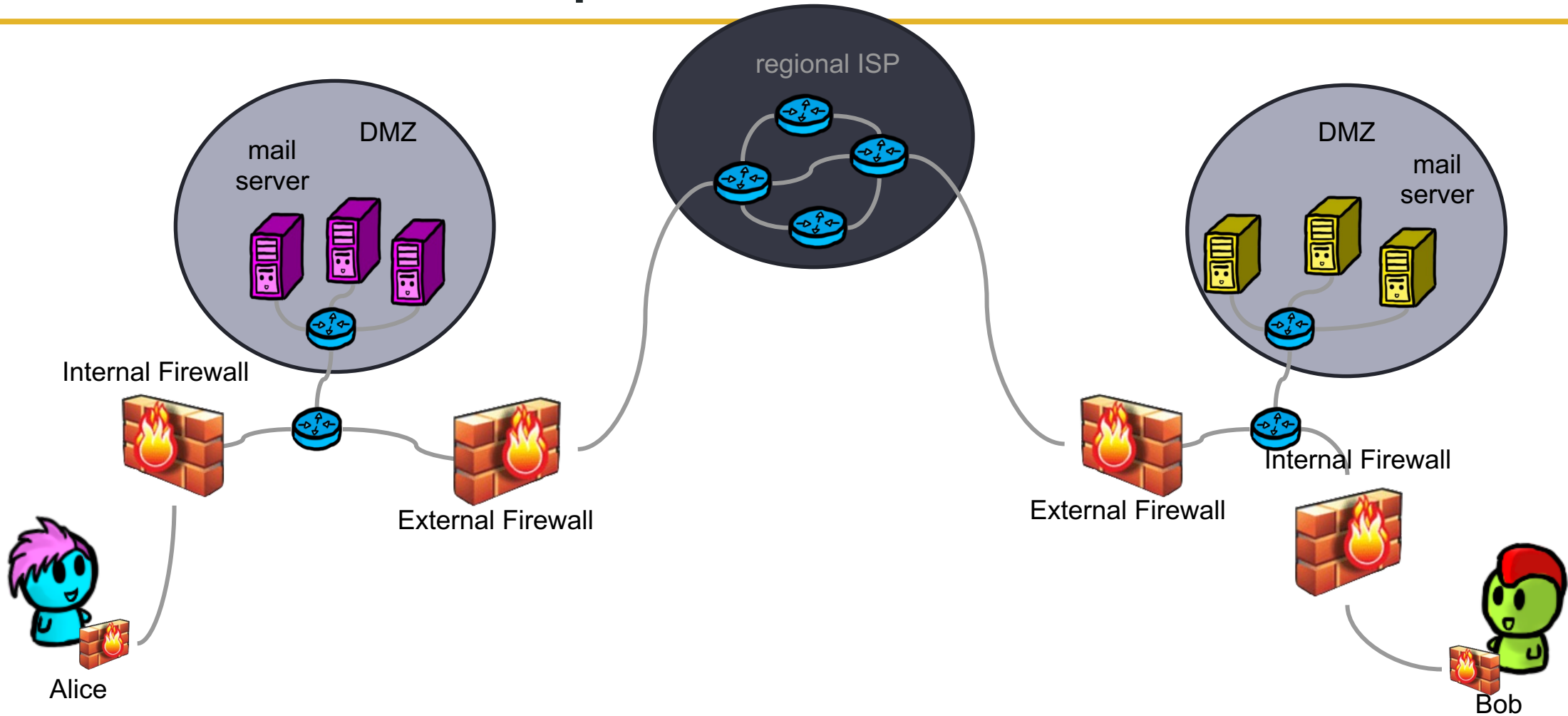
# A real-life DMZ

# Recall redundancy?

- Don't want to just rely on the firewall.

- What if something gets through?

- **Q:** How can we improve our firewall rules over time?
- **A:** Detection-based defenses as a second line of defence

# Intrusion Detection Advantages

- If detected fast enough can remove the threat

- Can act as a deterrent as it can be a pain to get around

- Gather intel about the attackers to improve first line of defense

# Honeypots / honeynets

- Set up an (unprotected) computer or an entire network as a trap for an attacker

- System has no production value, so any activity is suspicious
    - Any received email is considered spam

- Observe attacker to learn about new attacks, to identify and stop attacker, or to divert attacker from attacking real system

- The attacker should not be able to learn it targeted a honeypot

- The attacker may use the honeypots to break into real system

# Honeypots / honeynets



https://xkcd.com/350/
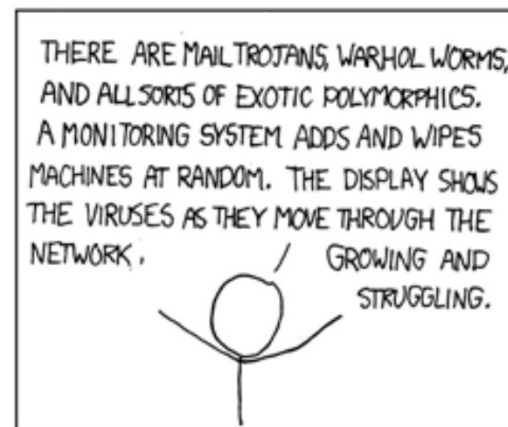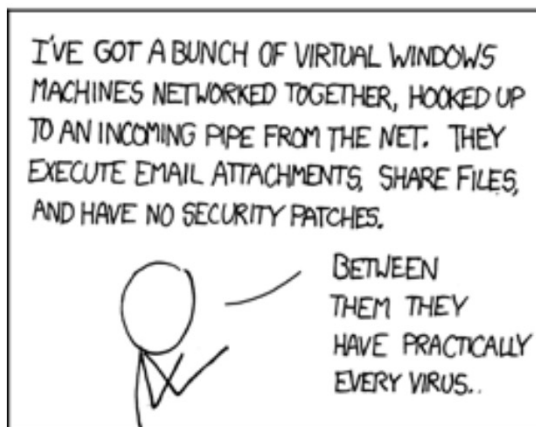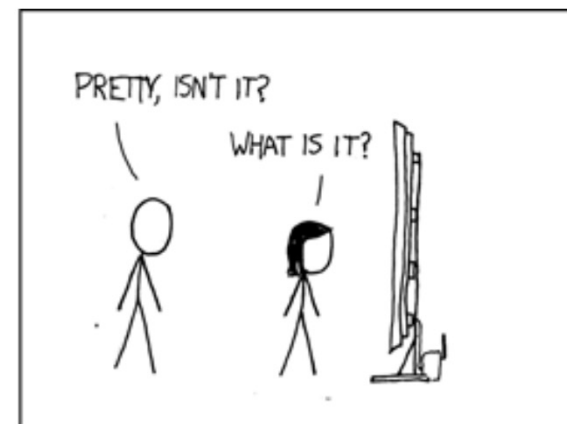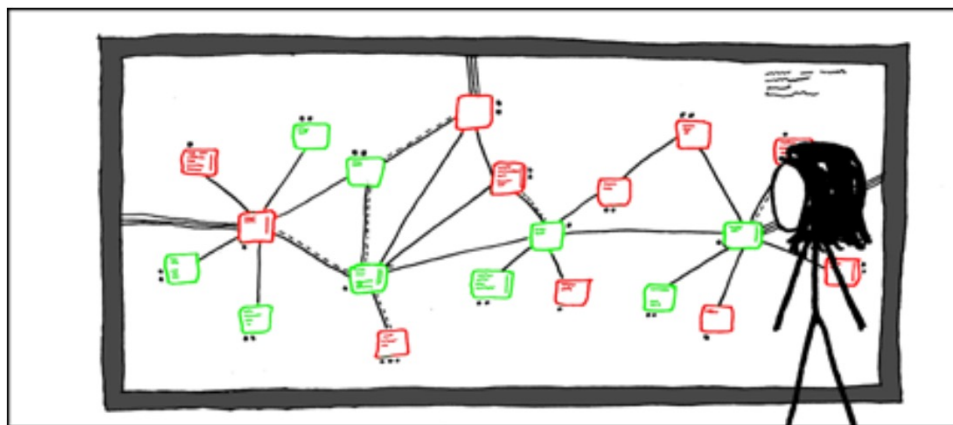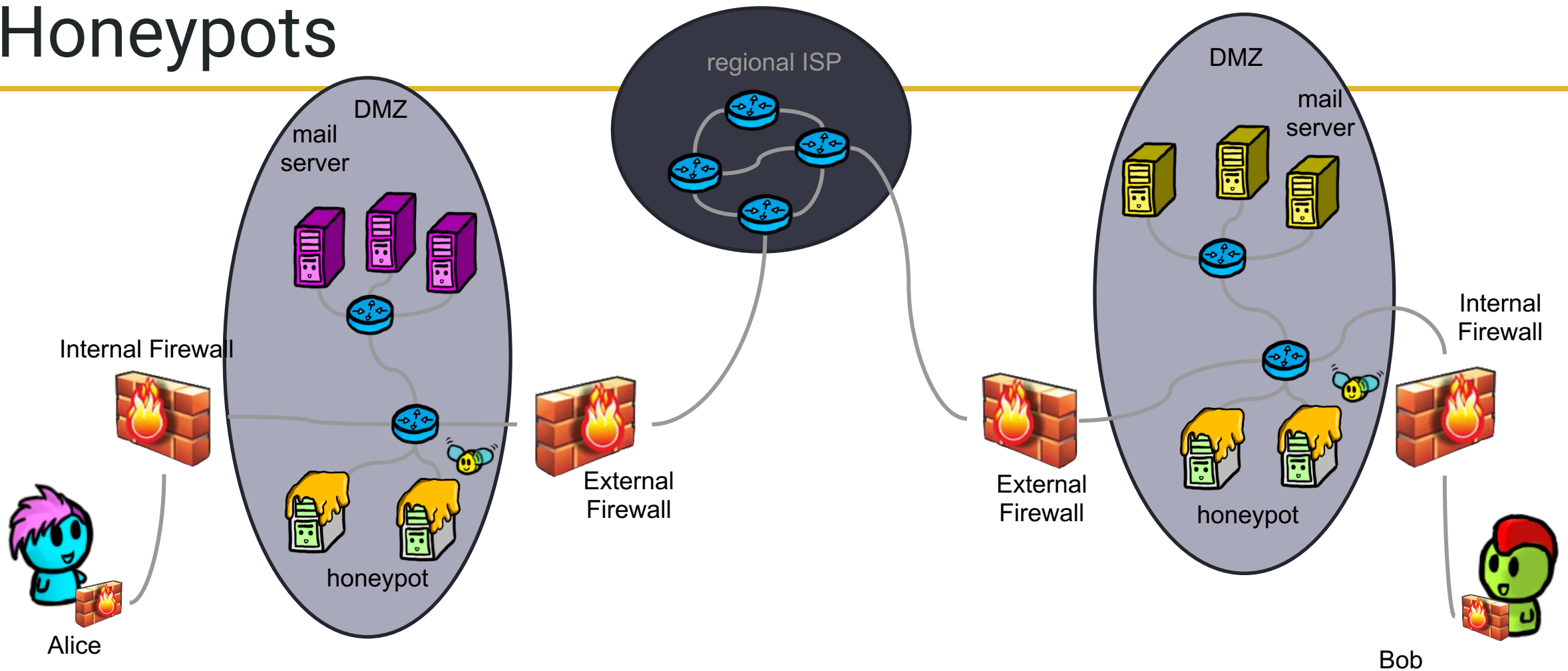
# Honeypots

# Types of honeypots/-nets

- ## Low interaction

  - Server that emulates one or multiple hosts, running different services
  - Easy to install and maintain
  - Limited amount of information gathering possible
  - Easier for the attacker to detect than high interaction honeynets

- ## High interaction

  - Deploy real hardware and software, use stealth network switches or keyloggers for logging data
  - More complex to deploy
  - Can capture lots of information
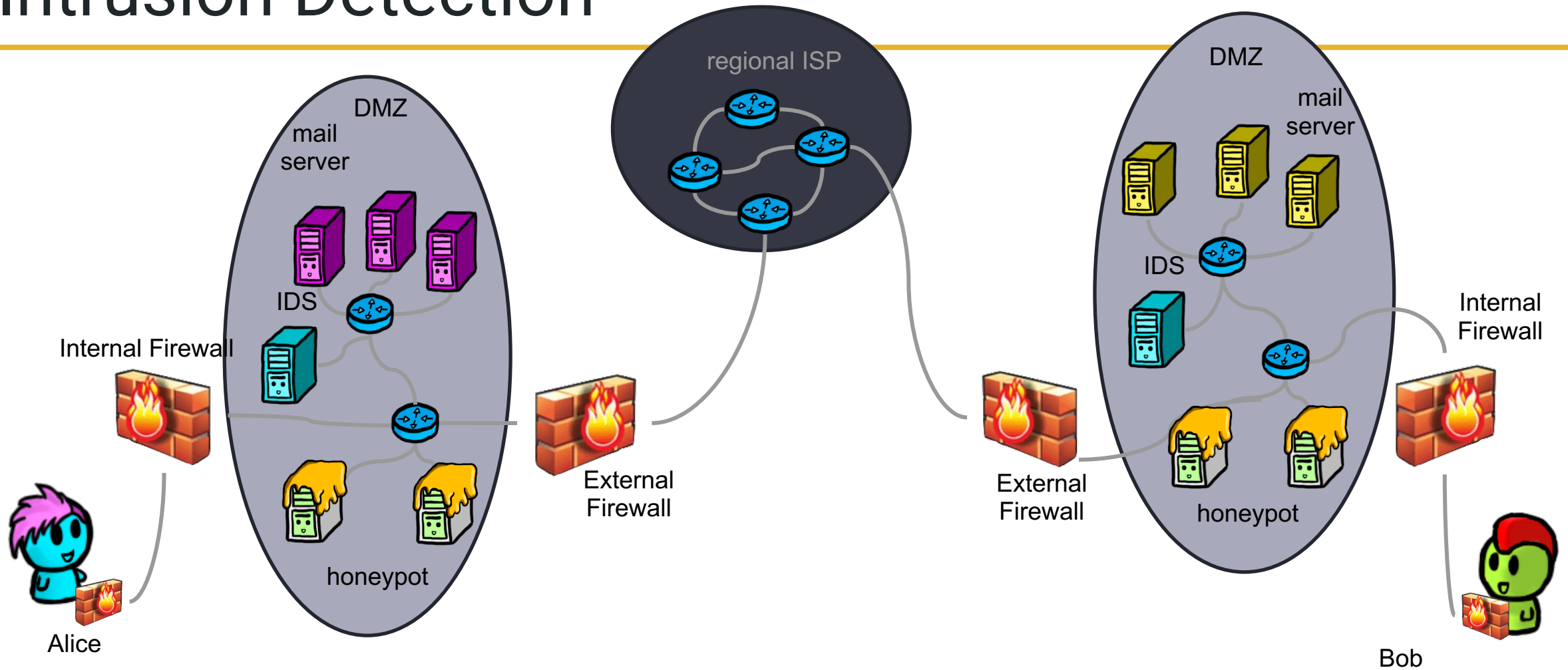  - Can capture unexpected behaviour by attacker

# Types of Honeypots

- ## External
  - Pros: Less risk to internal network, most data collected
  - Cons: Easier to detect

- ## DMZ
  - Pros: More realistic without endangering the internal network
  - Cons: Needs proper protection for other DMZ services

- ## Internal
  - Pros: Detects internal attacks
  - Cons: If compromised it's inside the network

# Intrusion detection systems (IDSs)

- Firewalls do not protect against inside attackers and are not perfect

- IDSs are next line of defense

- Monitor activity to identify malicious or suspicious events
  - Receive events from sensors
  - Store and analyze them
  - Take action if necessary

- Host-based and network-based IDSs

- Signature-based and heuristic/anomaly-based IDSs

# Intrusion Detection



regional ISP

DMZ

mail server

IDS

Internal Firewall

honeypot

Alice

External Firewall

DMZ

mail server

IDS

Internal Firewall

honeypot

External Firewall

Bob

# Host-based and network-based IDSs

- ## Host-based IDSs

  - Run on a host to protect this host
  - Can exploit lots of information (packets, disk, memory,. . . )
  - Miss out on information available to other (attacked) hosts
  - If host gets subverted, IDS likely gets subverted, too

- ## Network-based IDSs

  - Run on dedicated node to protect all hosts attached to a network
  - Have to rely on information available in monitored packets
  - Typically more difficult to subvert

- ## Distributed IDSs combine the two of them

# Signature-based IDSs

- Each (known) attack has its signature
  - E.g., many SYNs to ports that are not open could be part of a port scan
- Signature-based IDSs try to detect attack signatures
- Fail for new attacks or if attacker manages to modify attack such that its signature changes
  - Polymorphic worms
- Might exploit statistical analysis
  - Hypothesis test or threshold for frequency of certain events

# Heuristic/anomaly-based IDSs

- Look for behaviour that is out of the ordinary
- By modelling good behaviour and raising alert when system activity no longer resembles this model
- Or by modelling bad behaviour and raising alert when system activity resembles this model
- All activity is classified as good/benign, suspicious, or unknown
- Over time, IDS learns to classify unknown events as good or suspicious
  - Maybe with machine learning

# ML Based Intrusion Detection
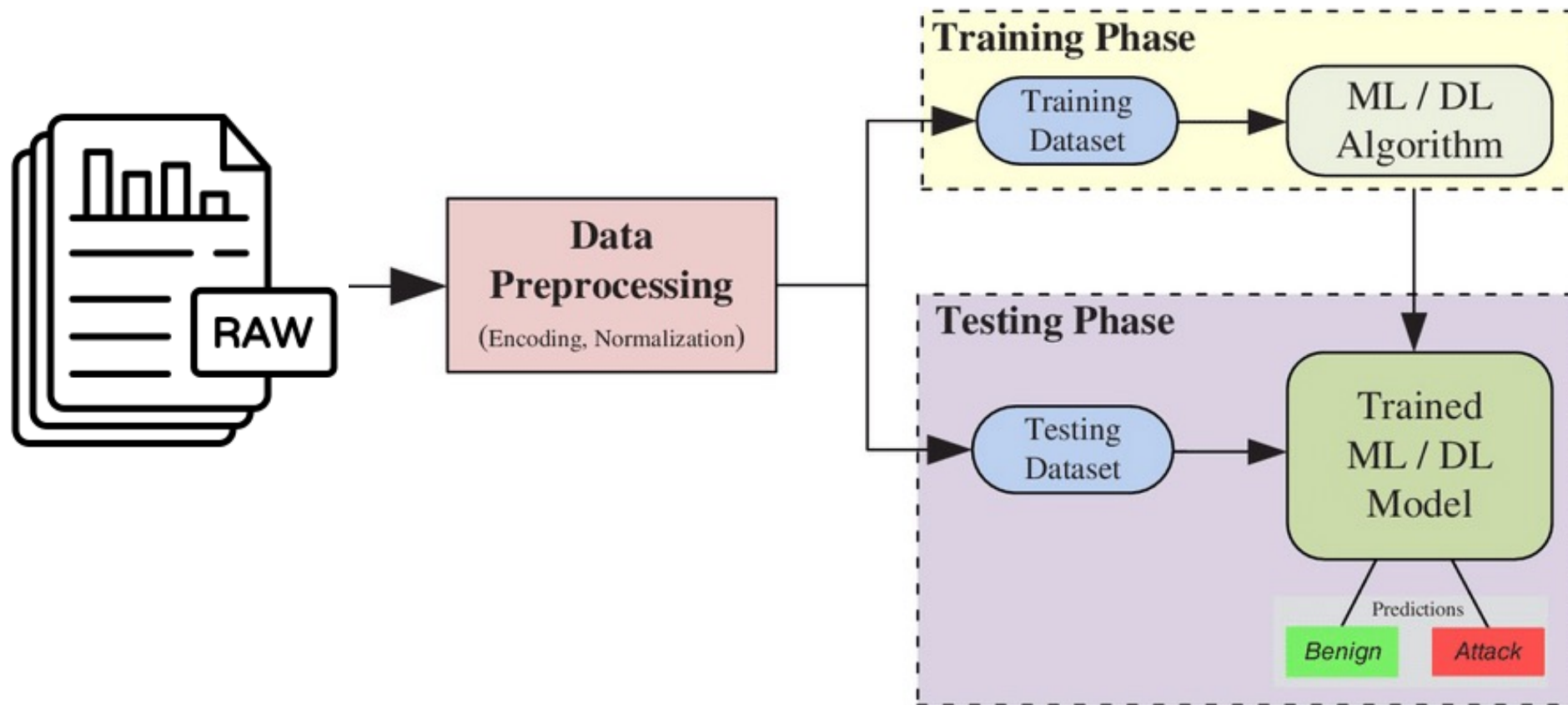
# ML Based Intrusion Detection

- Just because we can does not always mean we should!

- Typically, interpretable classifiers (trees, KNN)

- Need representative training set

  - Out of distribution samples can be a problem

- Keep parameters secret

  - Adversarial Examples (more later)

- Privacy issues (more later)

# IDS discussion

- ## Stealth mode
  - Two network interfaces, one for monitoring traffic, another one for administration and for raising alarms
  - First one has no published address, so it does not exist for routing purposes (passive wiretap)

- ## Responding to alarms
  - Type of response depends on impact of attack
  - From writing a log entry to calling a human

# False positives/negatives

○ Former might lead to real alarms being ignored

○ IDS might be tunable to strike balance between the two

○ In general, an IDS needs to be monitored to be useful
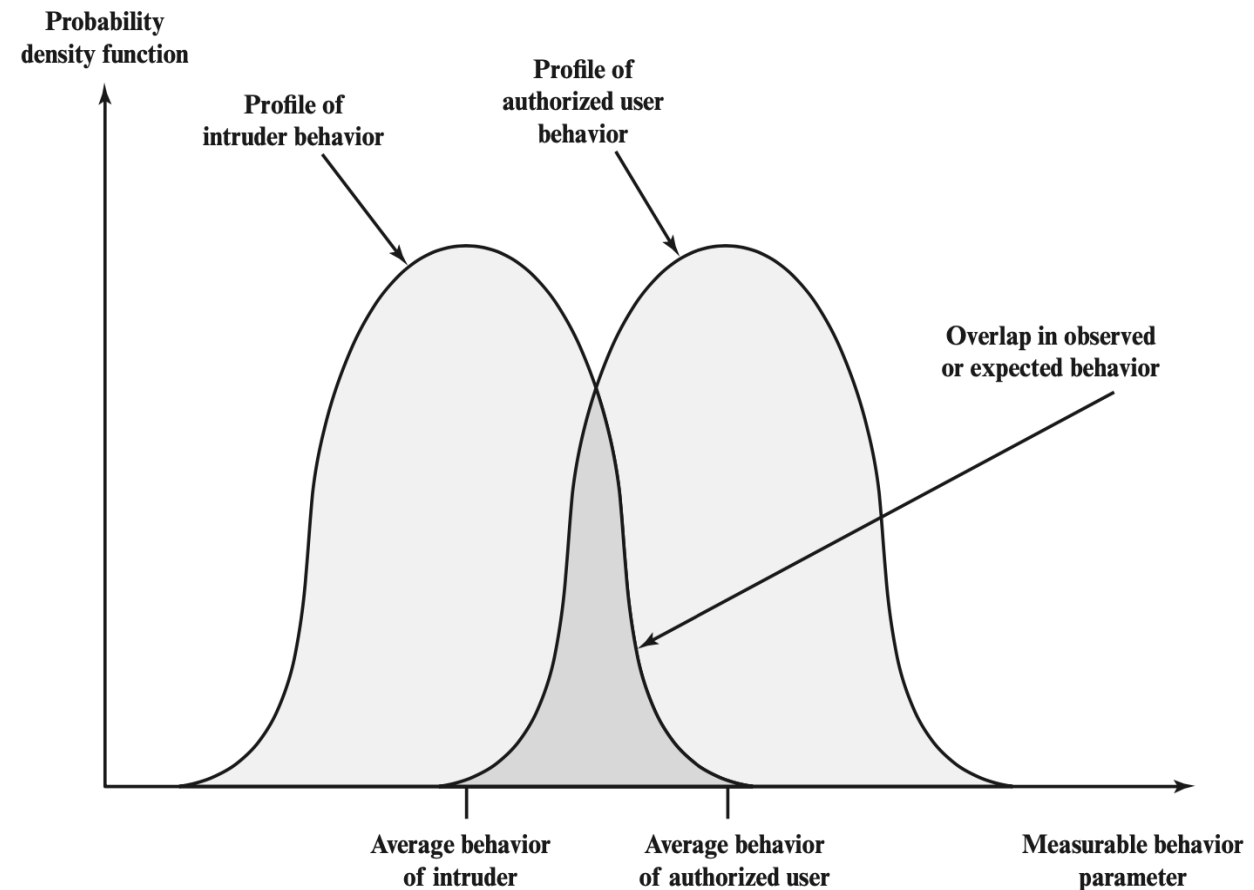


Figure 11.1    Profiles of Behavior of Intruders and Authorized Users

# Example: Snort

- 3 modes
  - Sniffer, Logger, IDS
- IDS is based on a list of rules
  - Can be similar to firewall or more sophisticated
  - Each rule an alert, log, etc
  - Comes with some and repo. Can also define more

```
alert tcp $EXTERNAL_NET 80 -> $HOME_NET any
(
    msg:"Attack attempt!";
    flow:to_client,established;
    file_data;
    content:"1337 hackz 1337",fast_pattern,nocase;
    service:http;
    sid:1;
)
```

# Example: Tripwire

- Anomaly-based, host-based IDS, detects file modifications
- Initially, compute digital fingerprint of each system file and store fingerprints at a safe place
- Periodically, re-compute fingerprints and compare them to stored ones
- (Malicious) file modifications will result in mismatches
- Why is it not a good idea to perform the second step directly on the production system?