CS459/698
Privacy, Cryptography, Network and Data Security

Network Security Primer

Fall 2024, Tuesday/Thursday 02:30pm-03:50pm

Security controls using cryptography

- We use cryptography as security control in situations where trust cannot be assumed.

- We will focus on network security (link layer, network layer, transport layer, and application layer).

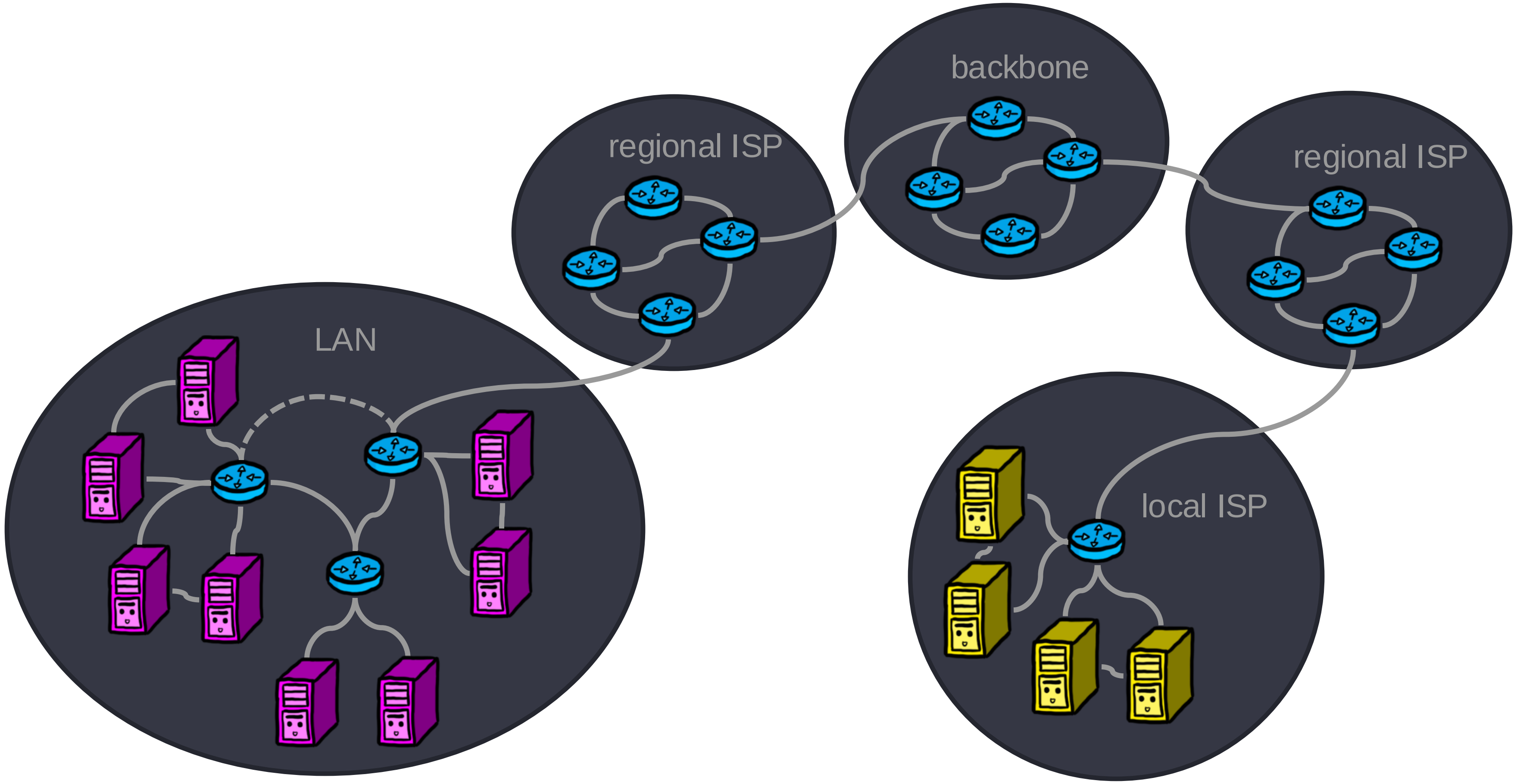
- Where entities you can only communicate with over a network are inherently less trustworthy. This makes networking a primary scenario for cryptography.

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

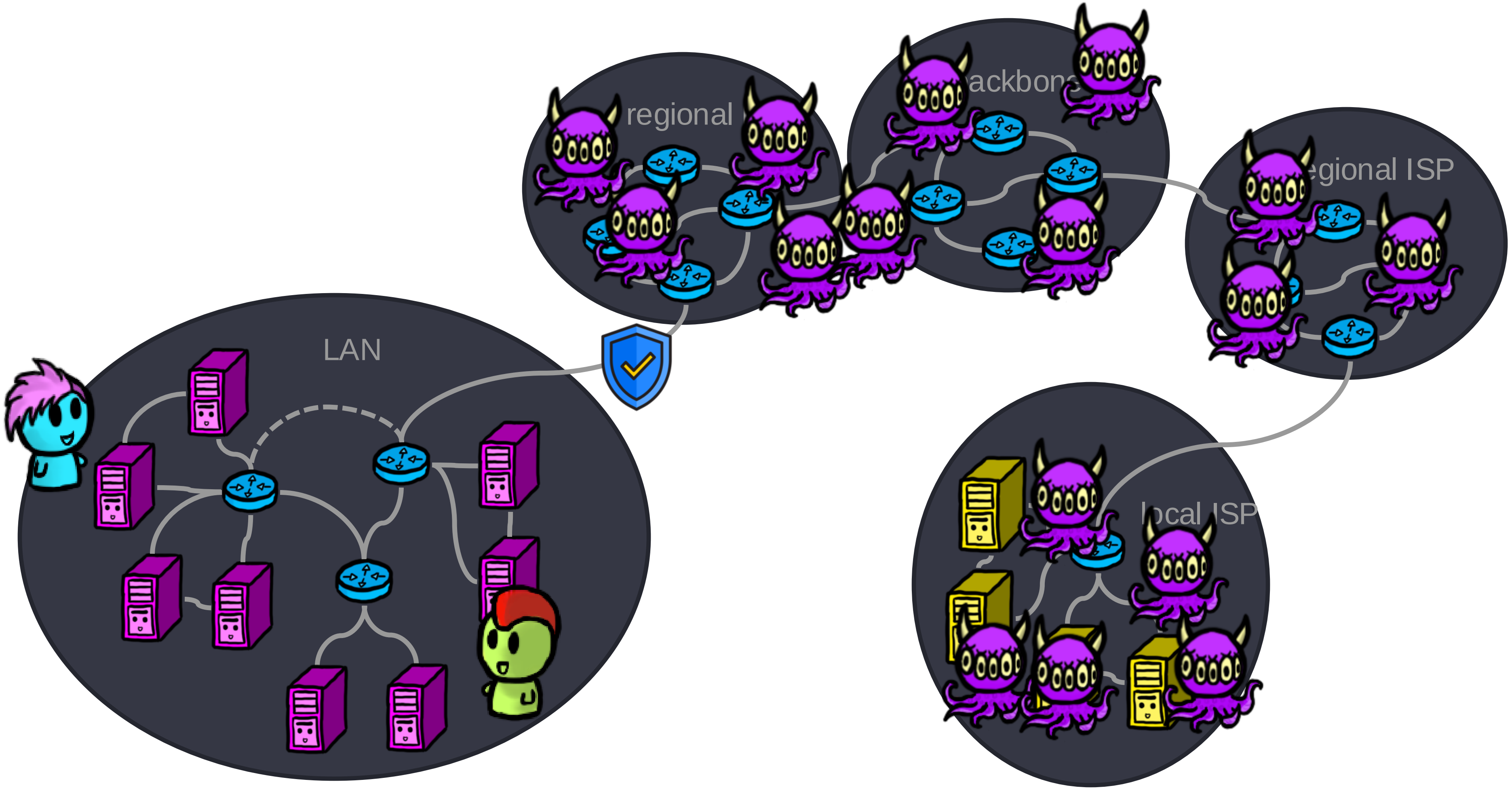
Architecture of the Internet



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,. . .

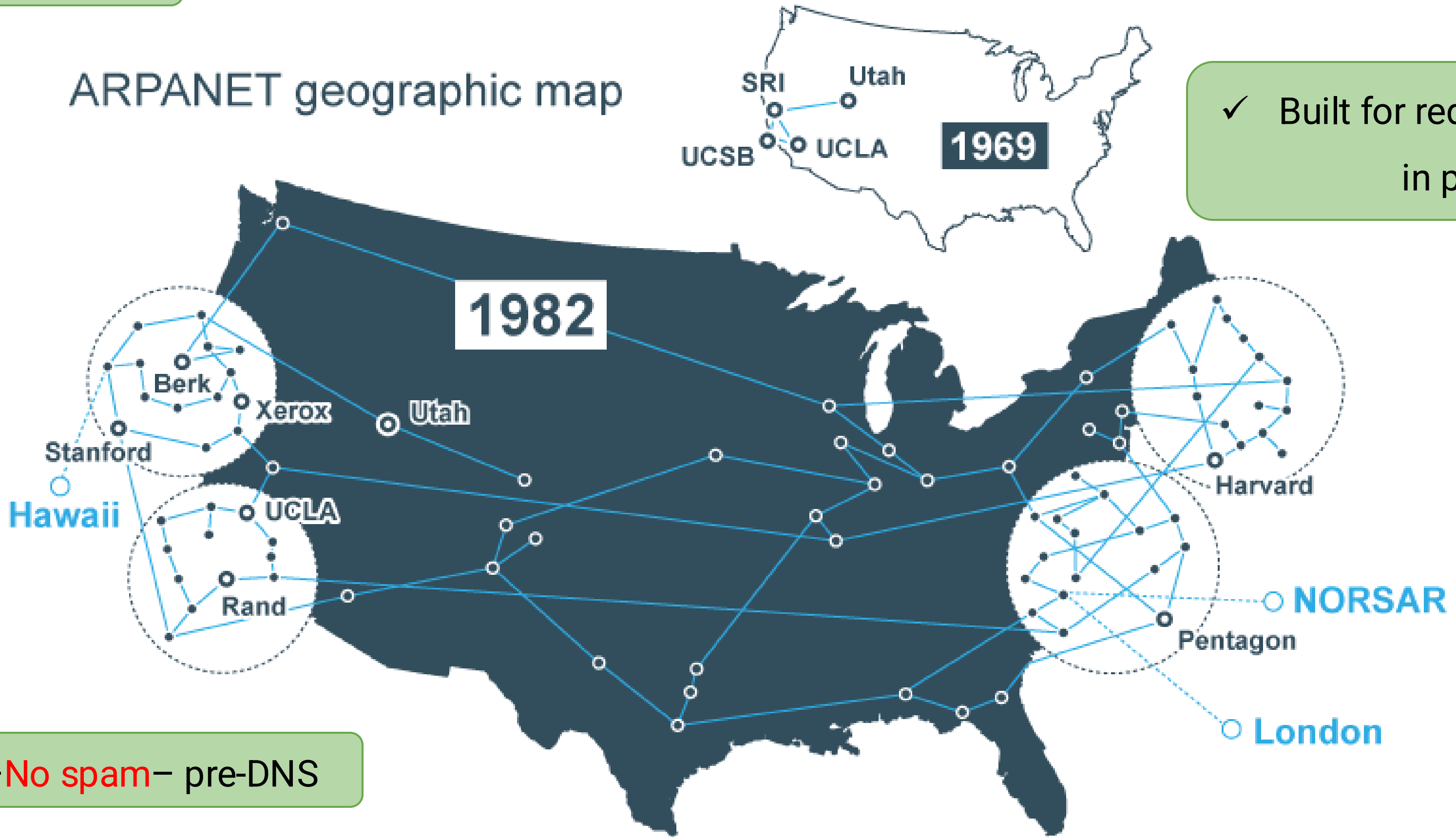
How can we be safe?



Networking Basics

ARPA-Network

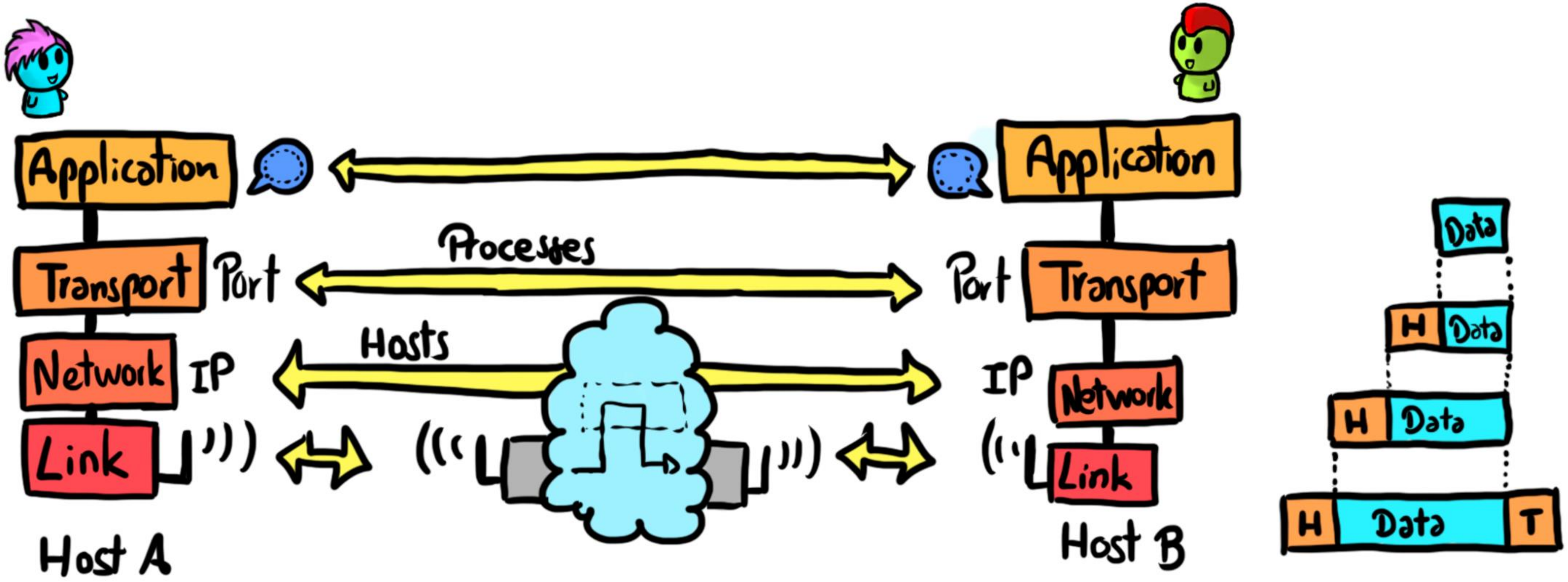
- ✓ The precursor to today's Internet



- ✓ Built for redundant communication in post-nuclear war.

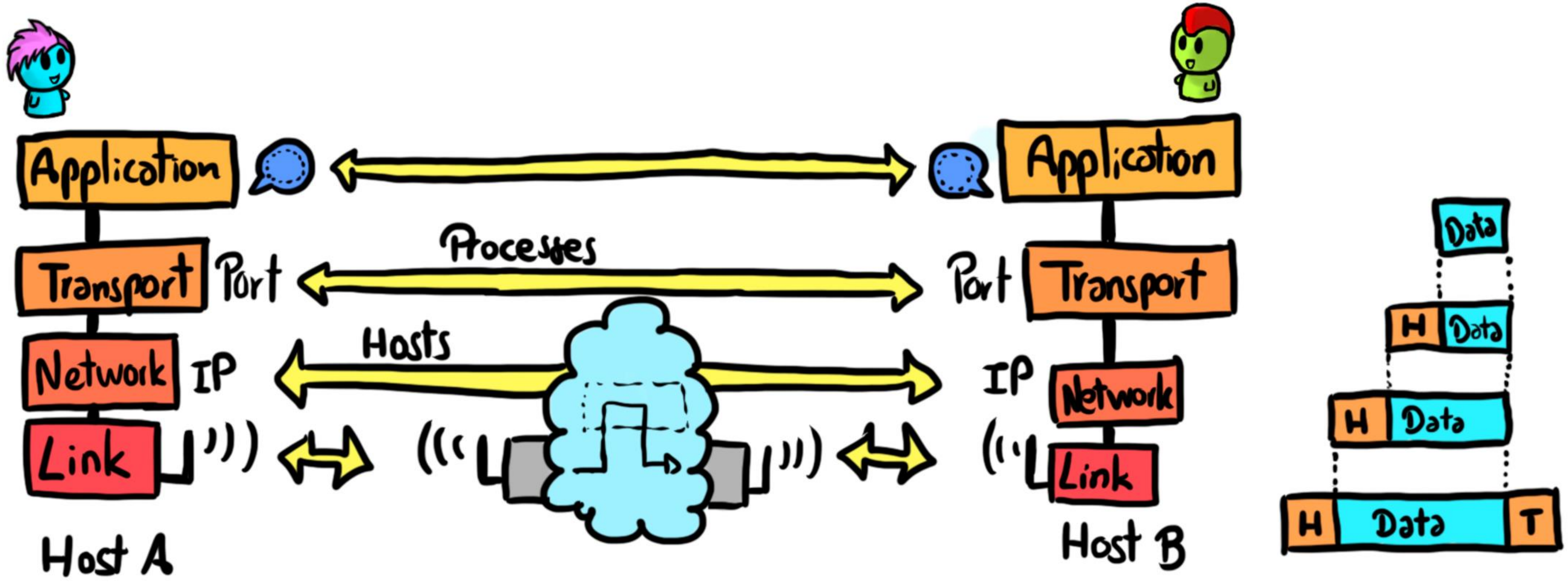
- ✓ Internet was small –No spam– pre-DNS

Simplified Network stack



- 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

Simplified Network stack



Q: Where do we need to apply crypto?(confidentiality, integrity, authentication)

(A) Link layer is enough

(B) Application layer is enough

(C) We need it in all layers

(D) Who needs crypto anyway ?

Network security and privacy

Cryptography is used at every layer of the network stack for both security and privacy applications. We will see some examples:

Link

- ➢ WEP, WAP, WAP2

Network

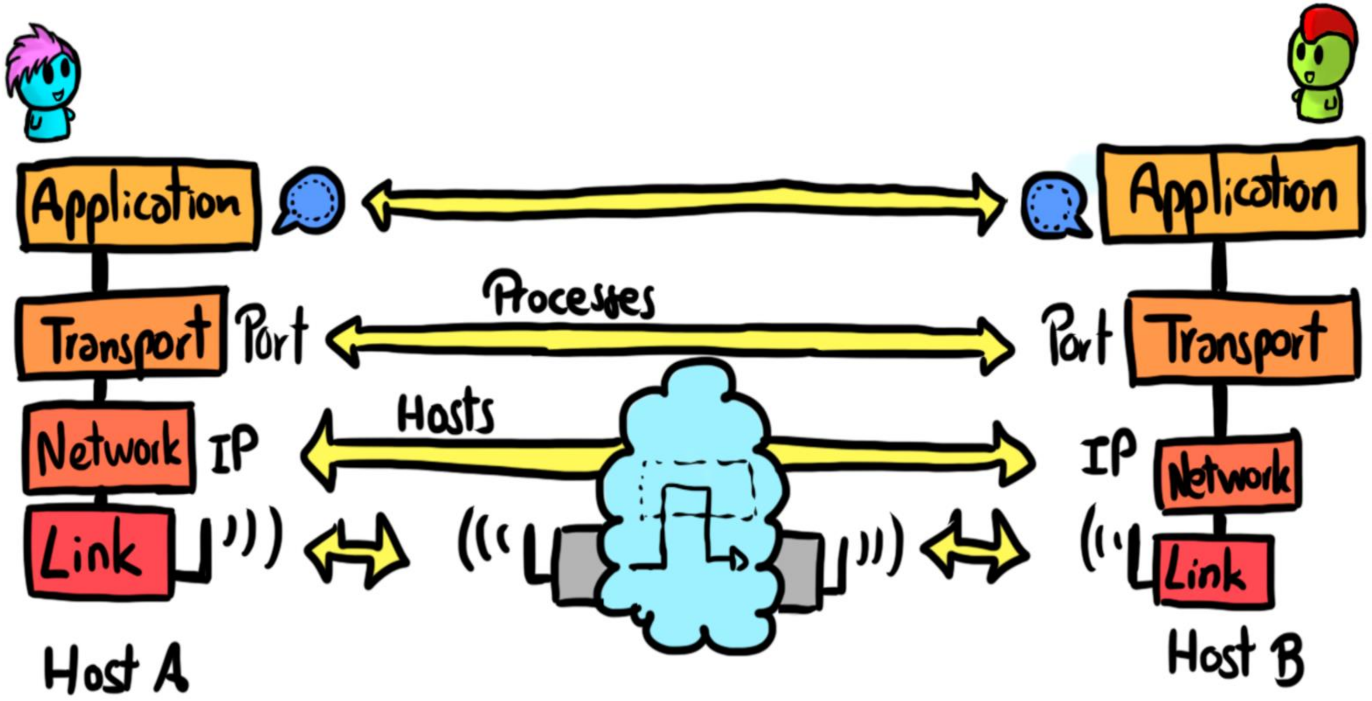
- ➢ VPN, IPsec

Transport

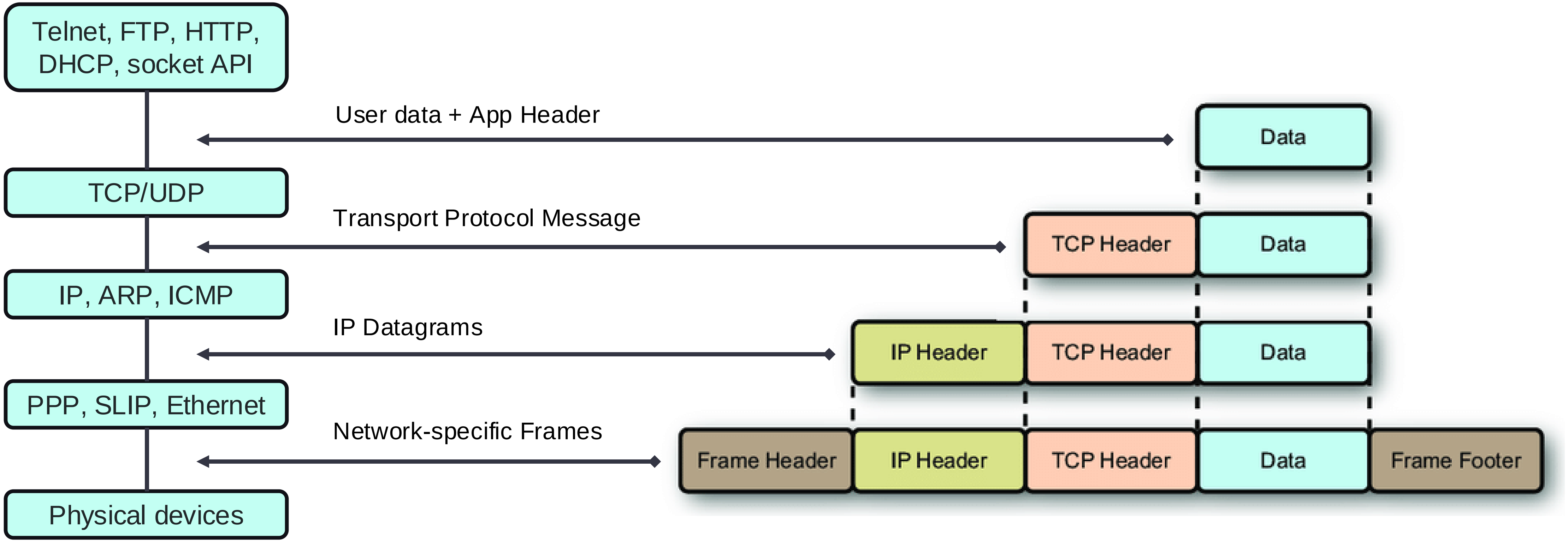
- ➢ TLS/SSL, Tor

Application

- ➢ SSH, PGP, OTR, Signal, Mixminion

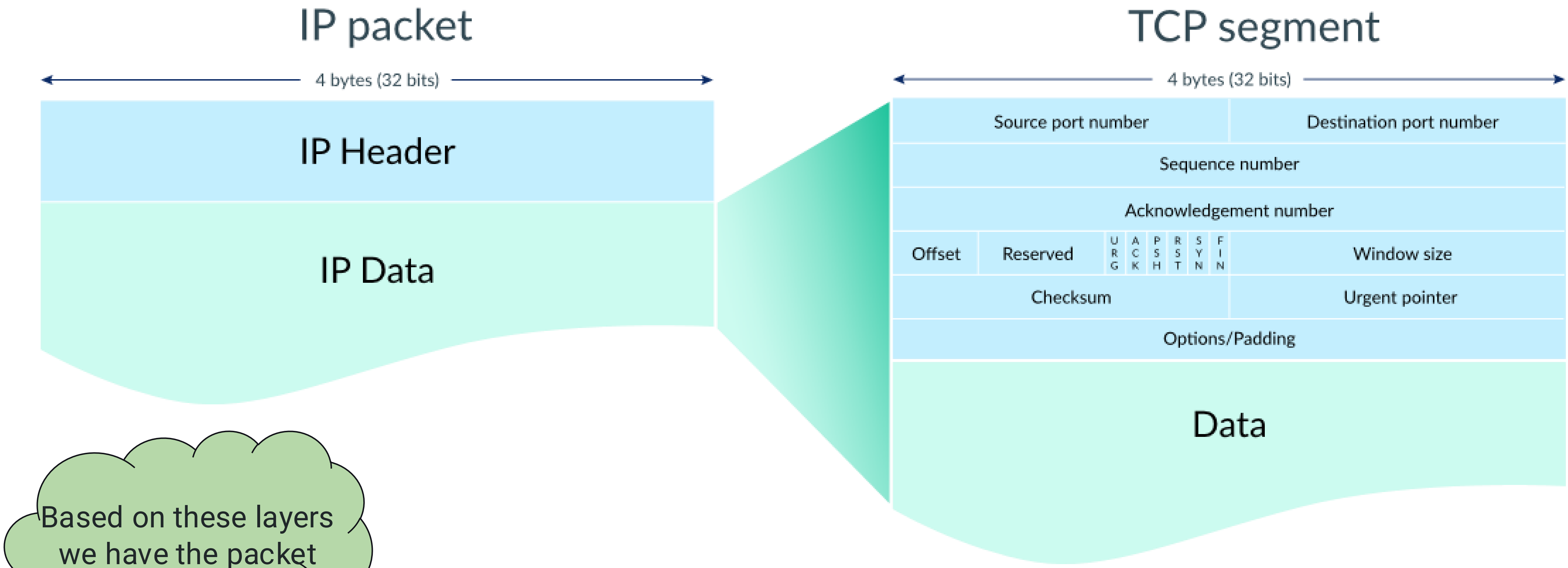


Packet Formats



Source: https://shorturl.at/BfxSt

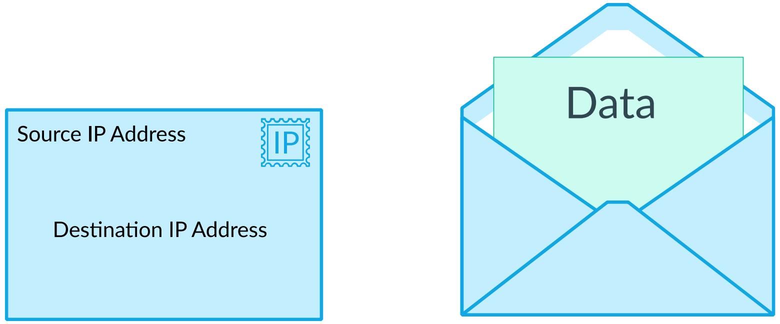
Packet Formats



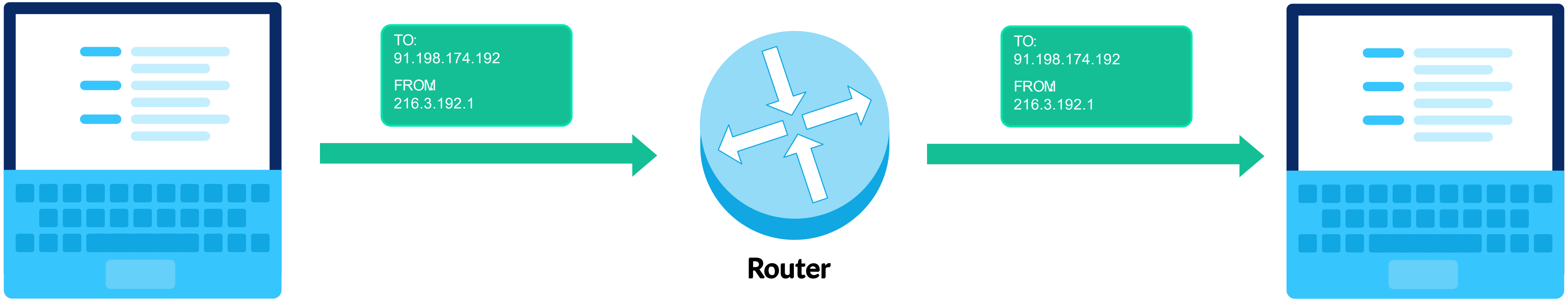
Based on these layers we have the packet decomposition.

IP

Header
(routing information)

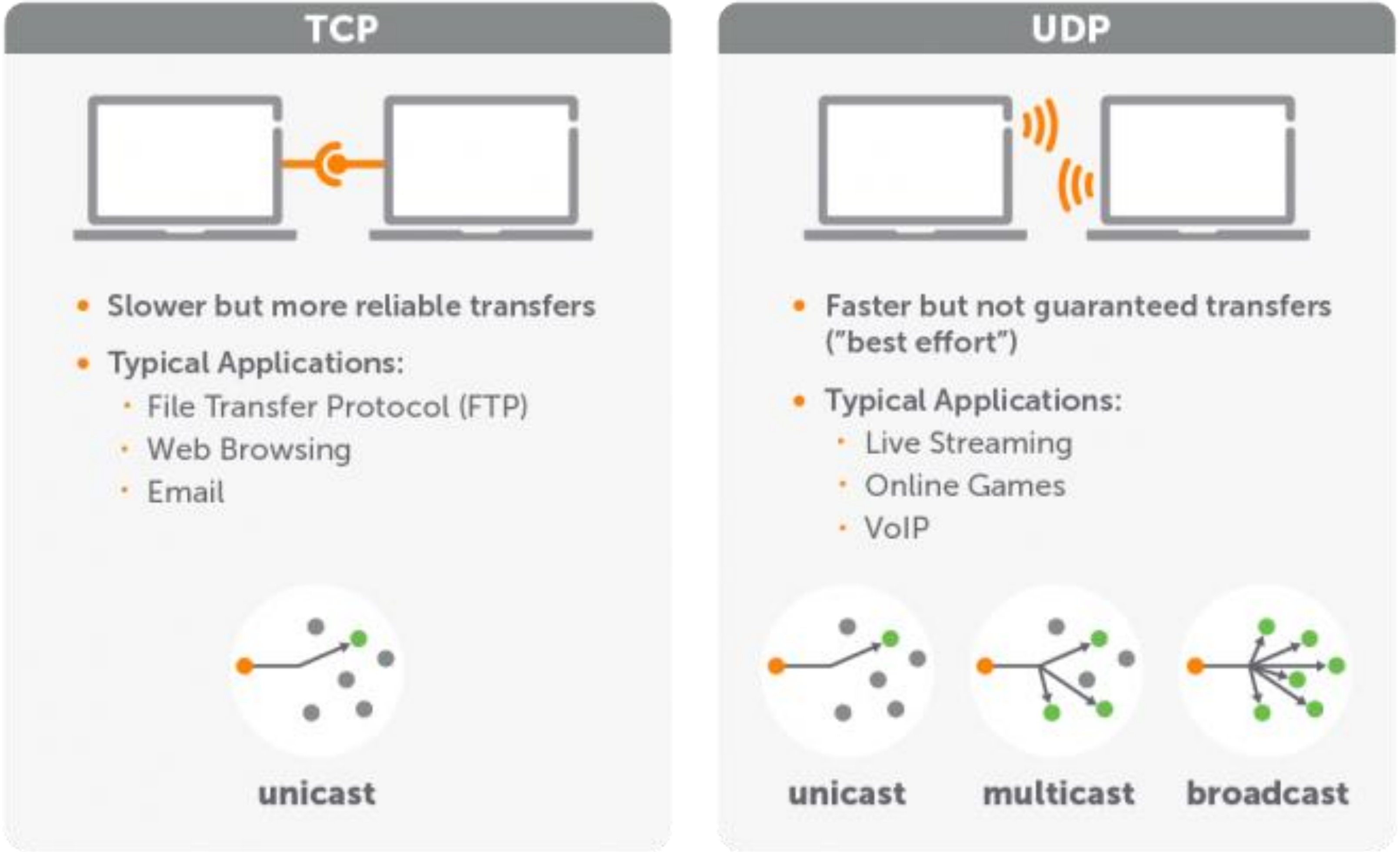


Payload
(Data sent to the recipient)



Source: https://shorturl.at/BfxSt

UDP vs TCP



TCP

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

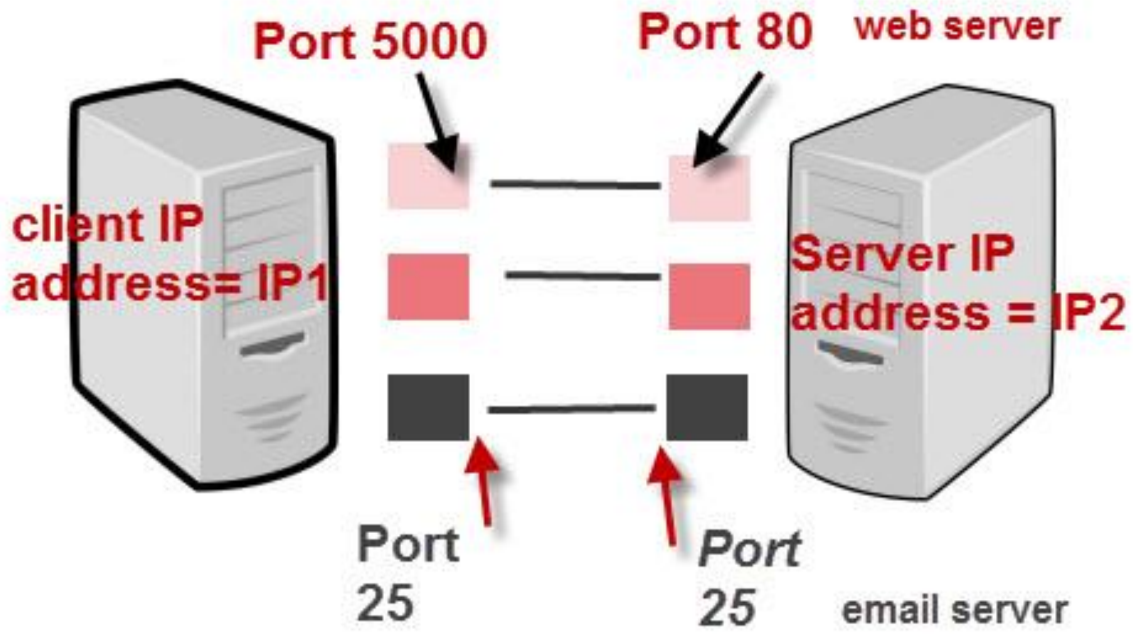
unicast

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

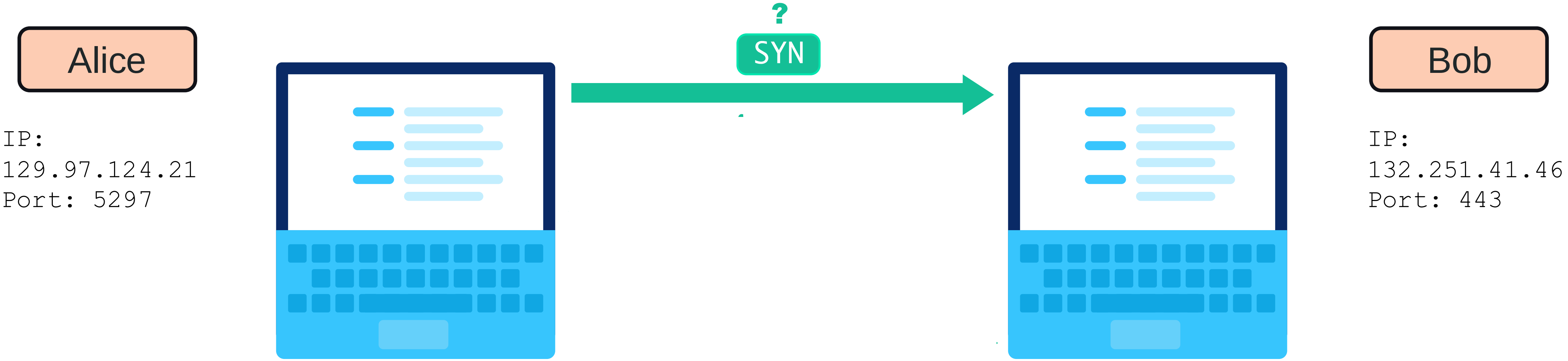
Alice

IP:
129.97.124.21
Port: 5297

Bob

IP:
132.251.41.46
Port: 443

The TCP Handshake

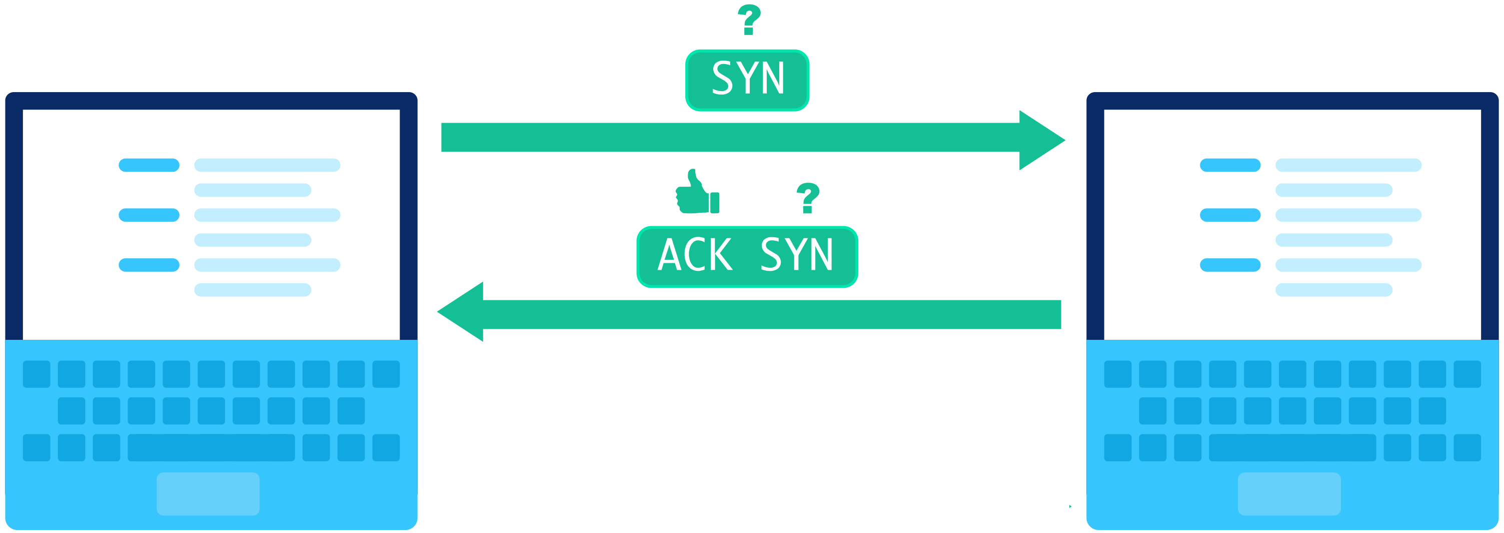


Alice sends a packet with the SYN bit set to 1
(SYN = "synchronize?").

The TCP Handshake

Alice

```
IP:
129.97.124.21
Port: 5297
```



Bob

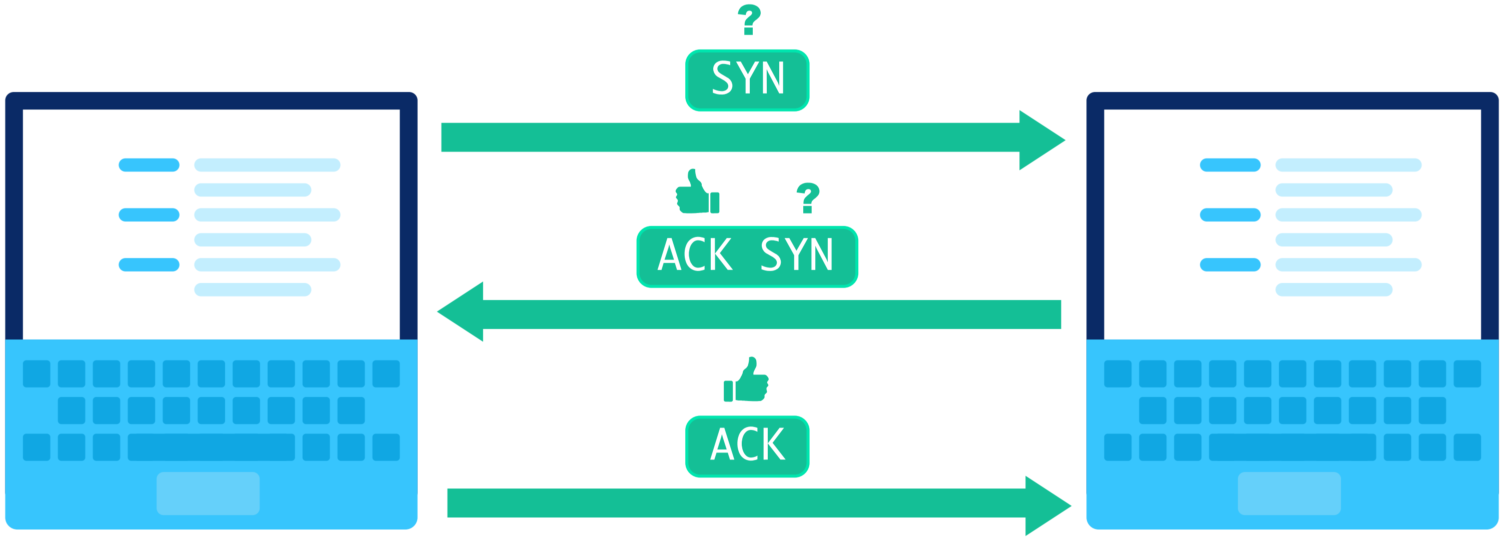
```
IP:
132.251.41.46
Port: 443
```

Bob sends back a packet with the ACK bit set to 1 (ACK = ”acknowledged!").

The TCP Handshake

Alice

```
IP:
129.97.124.21
Port: 5297
```



Bob

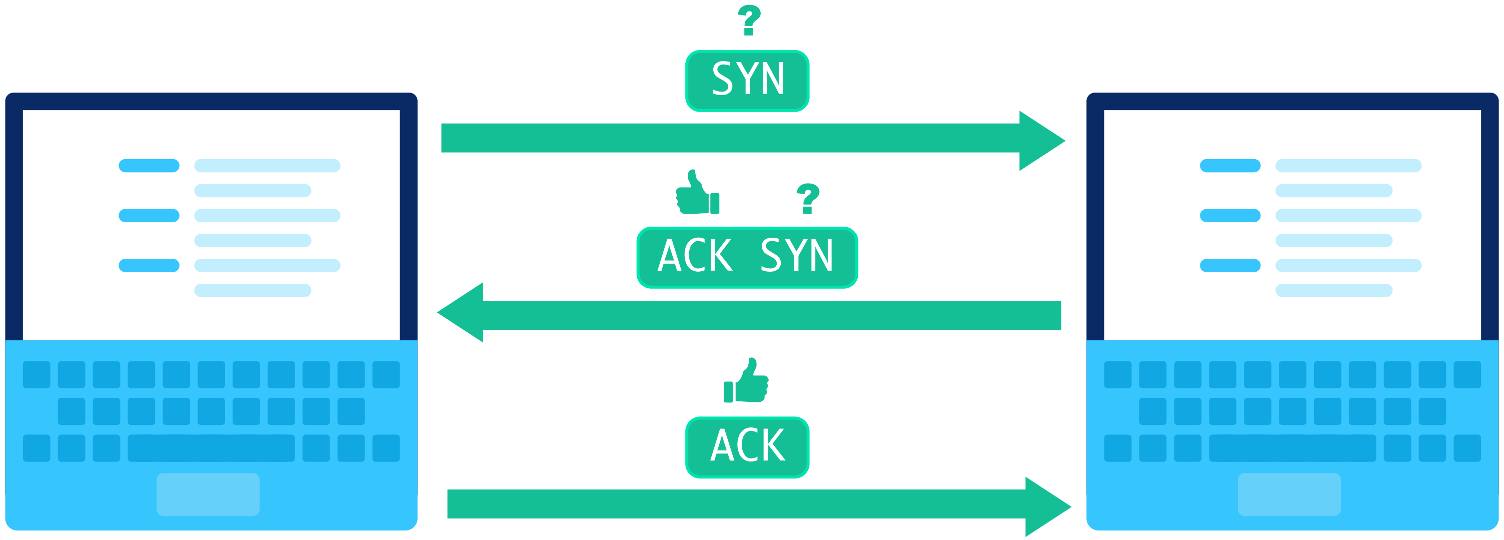
```
IP:
132.251.41.46
Port: 443
```

Alice replies back with an ACK

The TCP Handshake

Alice

```
IP:
129.97.124.21
Port: 5297
```



Bob

```
IP:
132.251.41.46
Port: 443
```

4 bytes (32 bits)

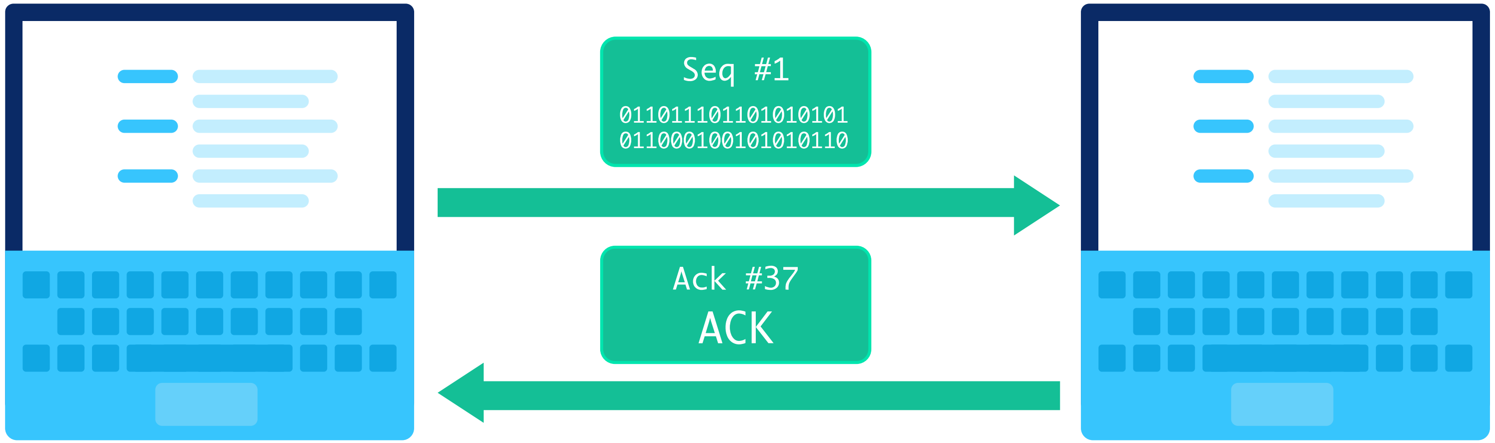
Source port number								Destination port number
Sequence number								
Acknowledgement number								
Offset	Reserved	URG	ACK	PSH	RST	SYN	FIN	Window size
Checksum								Urgent pointer
Options/Padding								

The SYN and Ack bits are both part of the TCP header.

The TCP Packet

Alice

IP:
129.97.124.21
Port: 5297



Bob

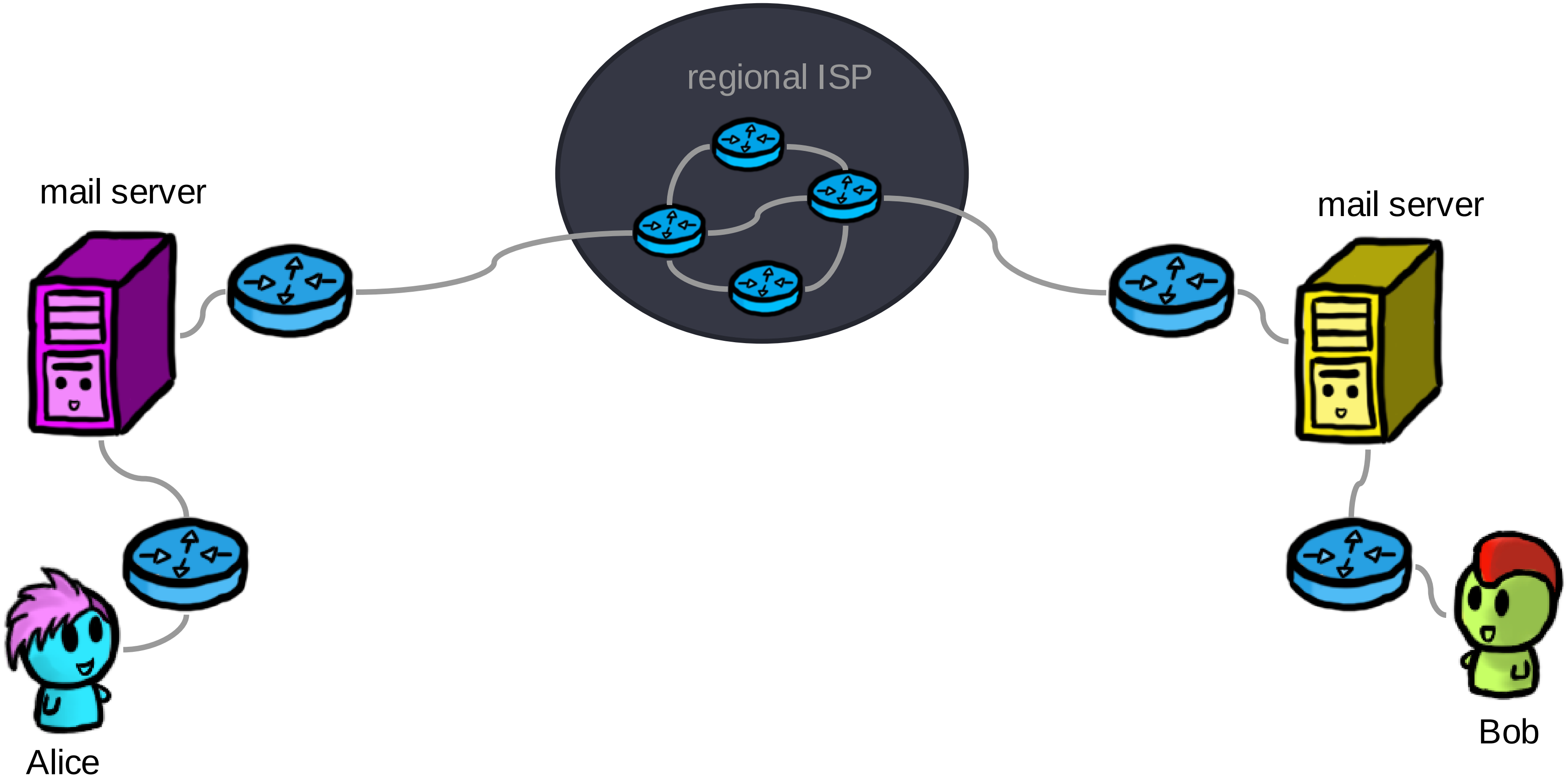
IP:
132.251.41.46
Port: 443

4 bytes (32 bits)

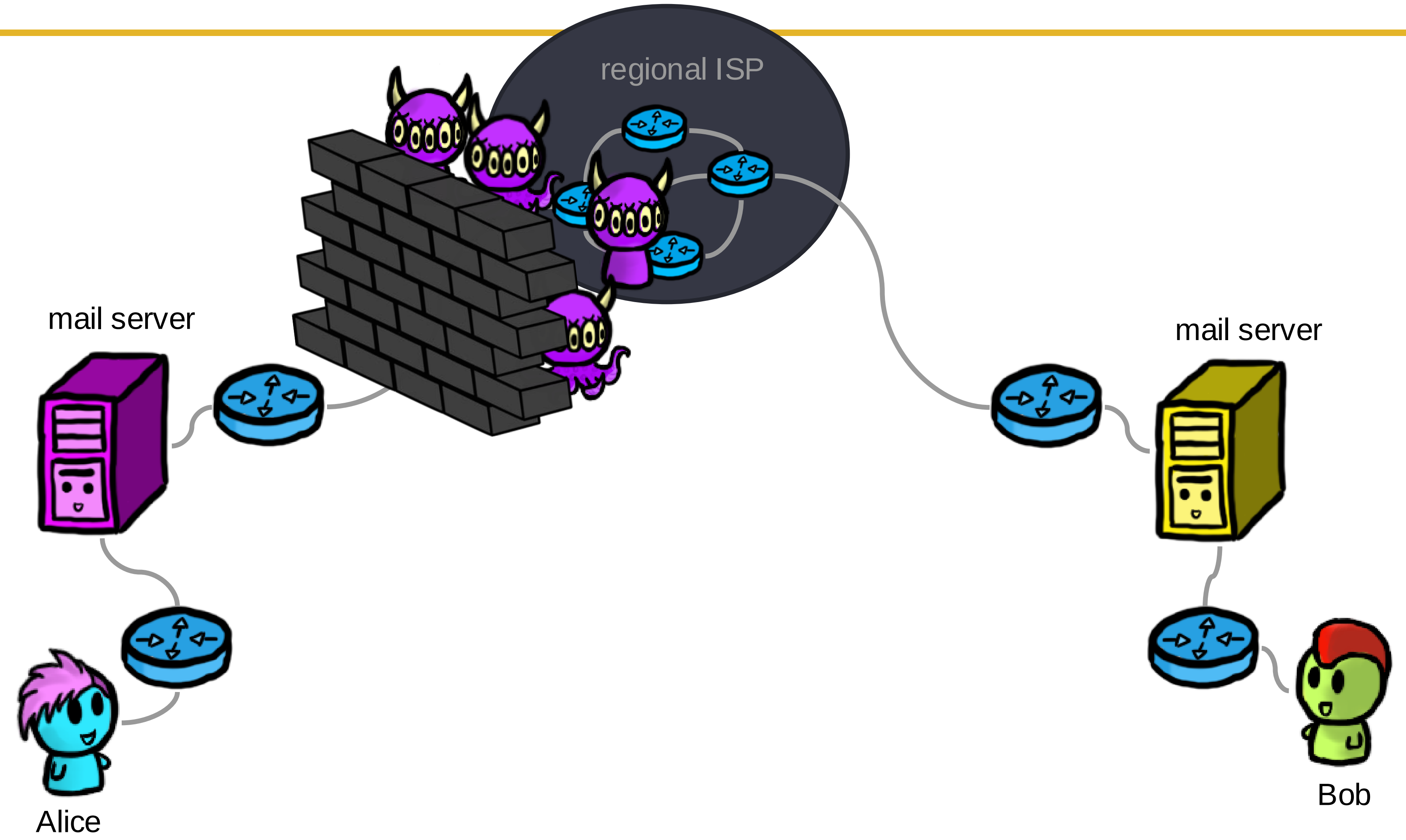
<table>
<tr><td colspan="8">Source port number</td><td>Destination port number</td></tr>
<tr><td colspan="9">Sequence number</td></tr>
<tr><td colspan="9">Acknowledgement number</td></tr>
<tr><td>Offset</td><td>Reserved</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window size</td></tr>
<tr><td colspan="8">Checksum</td><td>Urgent pointer</td></tr>
<tr><td colspan="9">Options/Padding</td></tr>
</table>

Keep track of which data was successfully received.

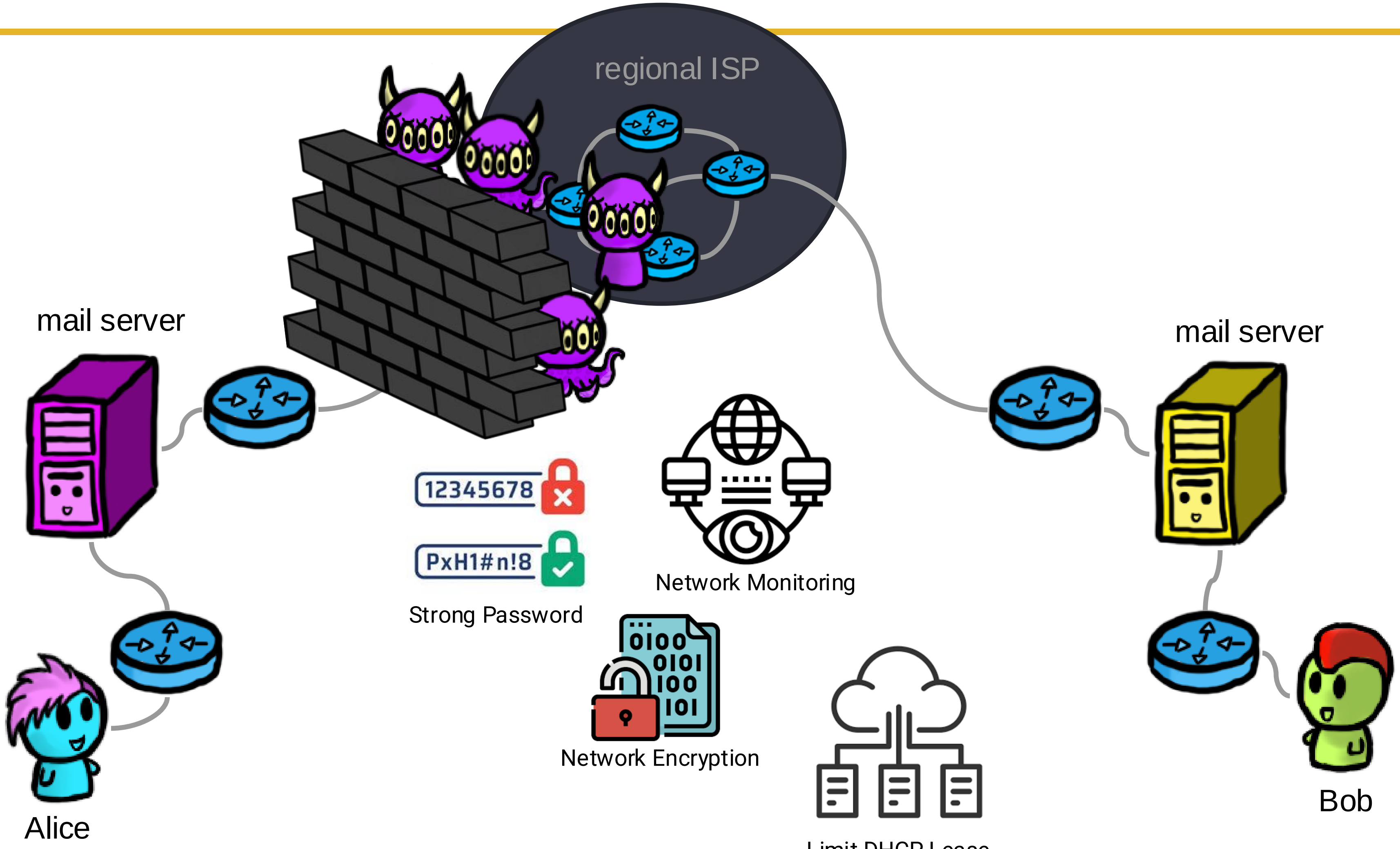
Working Example



Keep Unauthorized Individuals Out of the LAN



Keep Unauthorized Individuals Out of the LAN



Strong Password

Network Monitoring

Network Encryption

Limit DHCP Lease

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…

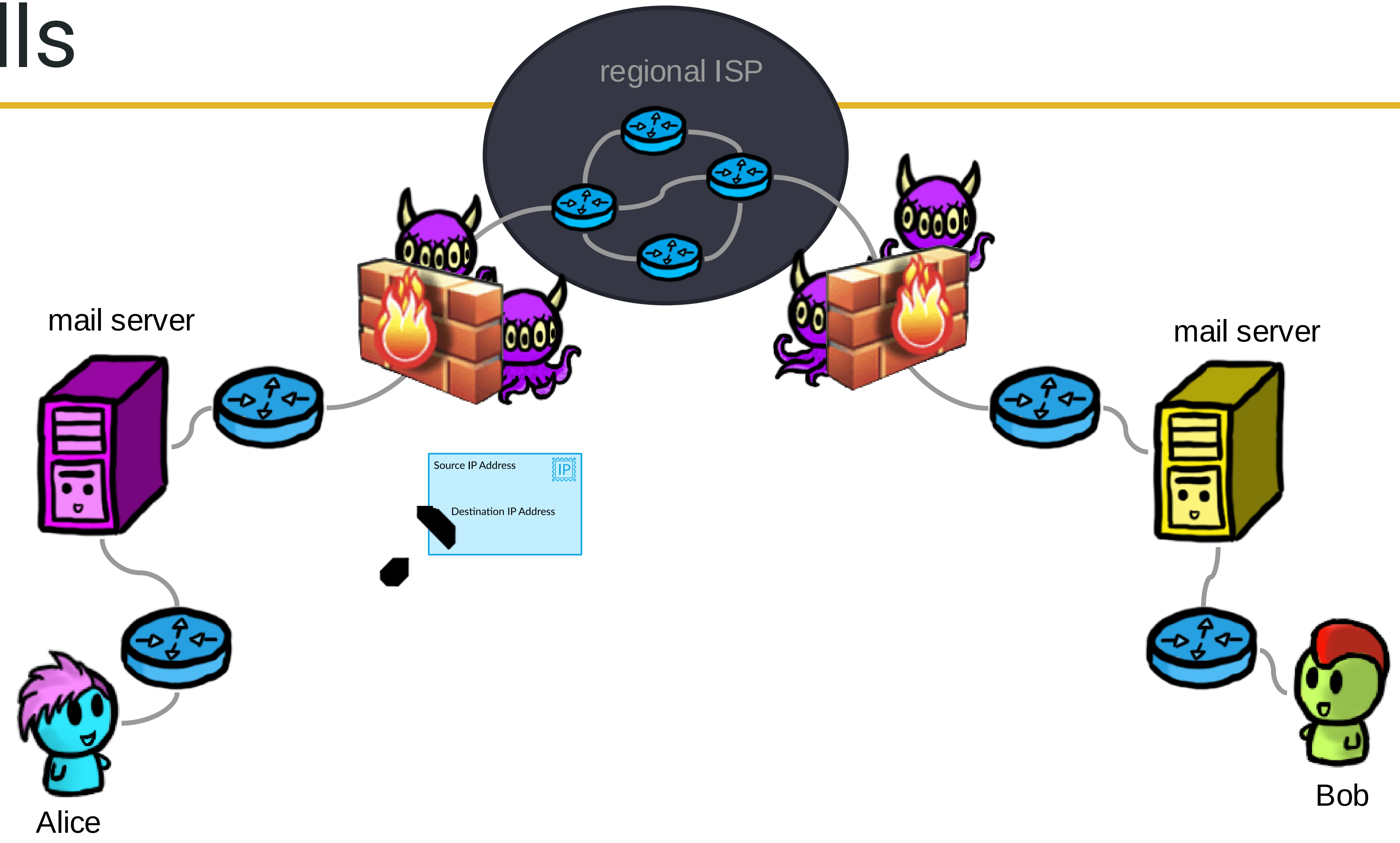
- Servers should be deployed in a **redundant** way on multiple machines, ideally with different software to get **genetic diversity** and at **different locations**

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

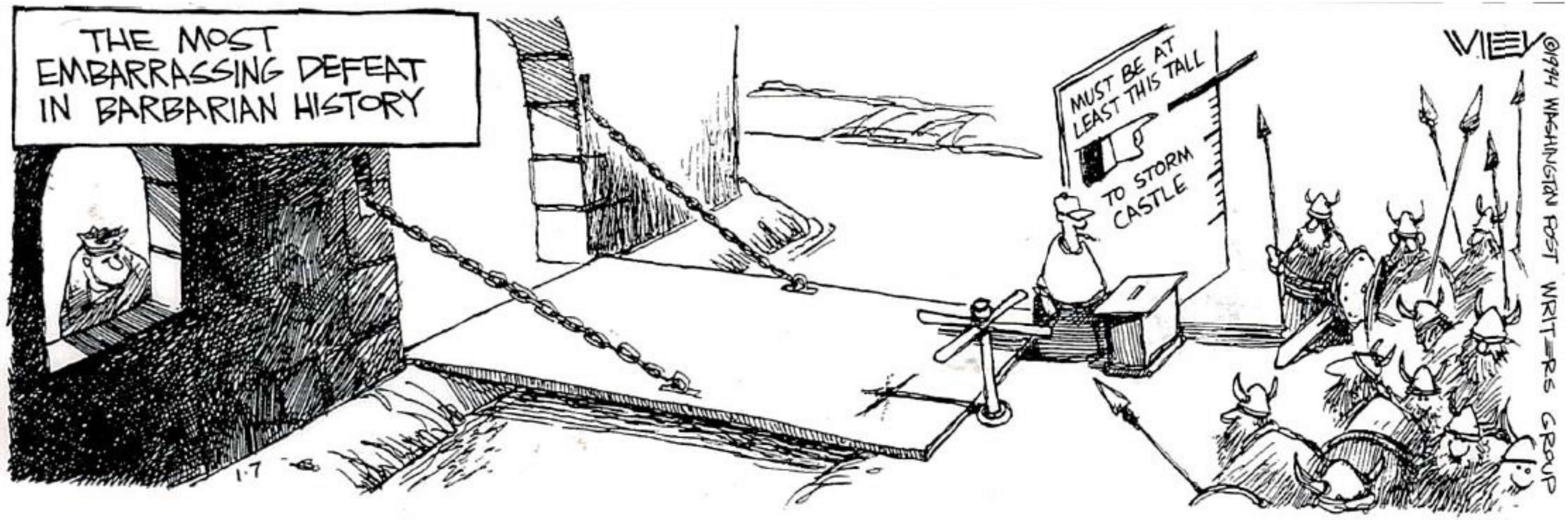
Access Controls Lists

- Routers convert IP address to MAC addresses locally
  - 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
  - Doesn’t help much in DDoS.
  - If traffic is cut off, we can’t analyze it.
- Need something more sophisticated.

Firewalls


regional ISP
mail server
mail server
Source IP Address
IP
Destination IP Address
Alice
Bob

Firewalls


THE MOST EMBARRASSING DEFEAT IN BARBARIAN HISTORY
MUST BE AT LEAST THIS TALL TO STORM CASTLE
1-7
VIEY
©1994 WASHINGTON POST WRITERS GROUP

Firewalls

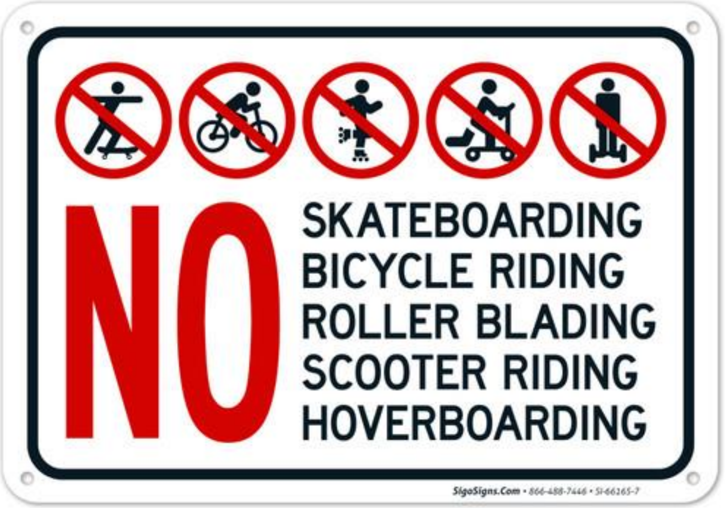
- Firewalls are the castles of the Internet age

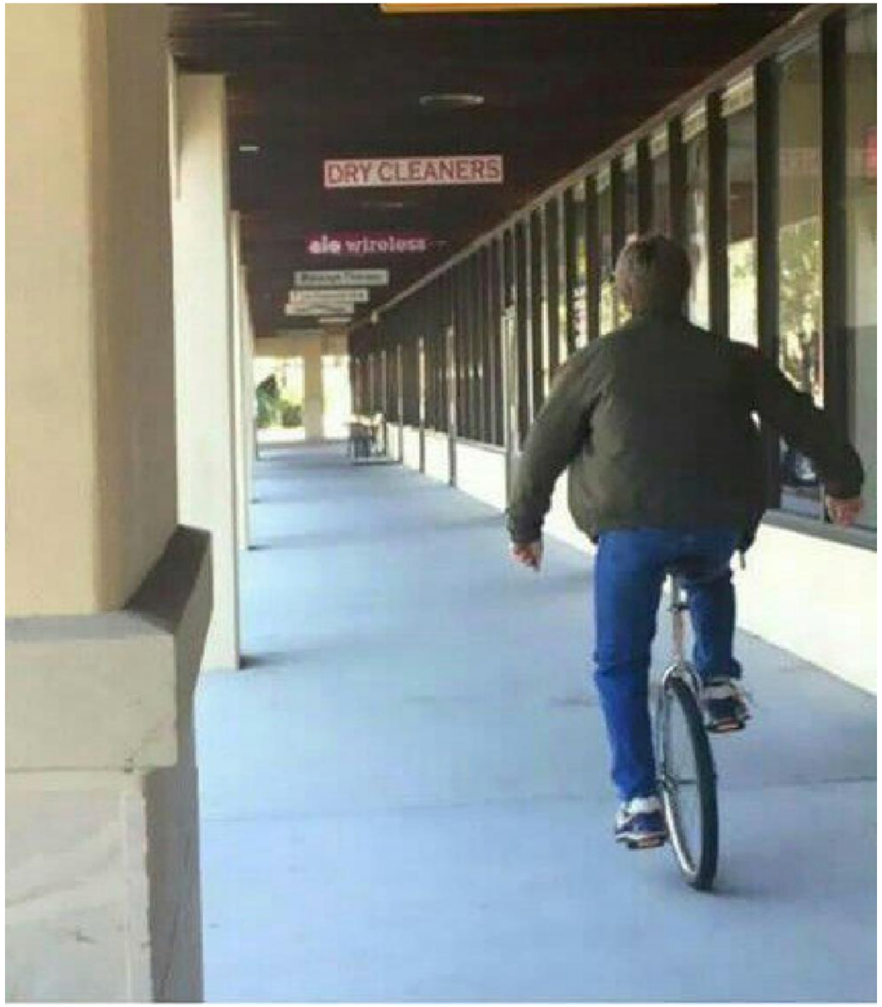
- All traffic into/out of a company has to go through a small number of gates (choke points)
  - Wireless access point should be outside of firewall

- 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"→ Preferred, Bcs we have implicit denial of access.
    → We can't anticipate all attacks.

Blocklist vs allowlist

Blocklist vs allowlist





Allow-list:

- ✓ Has less chances of being circumvented.
- ✓ Can have a greater effect on useability
- ✓ More popular these days

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
  - Configured from internal network, or dedicated machine not connected to the internet

Packet filtering gateways

- Simplest type – very fast and transparent

- Make decision based on network header of a packet

- 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

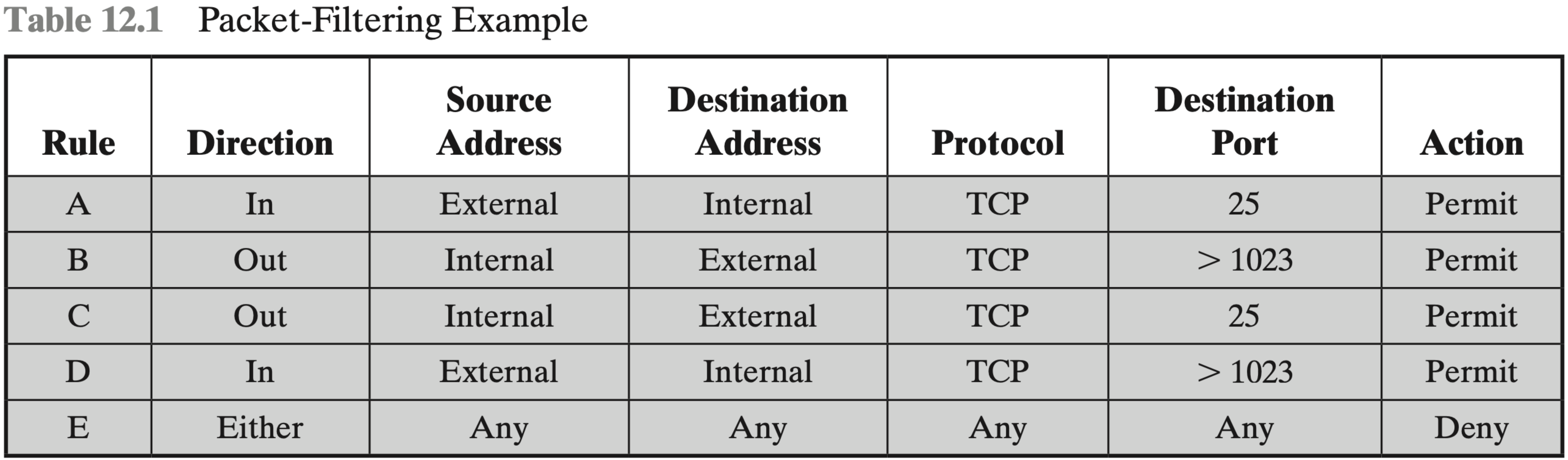


Table 12.1 Packet-Filtering Example

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

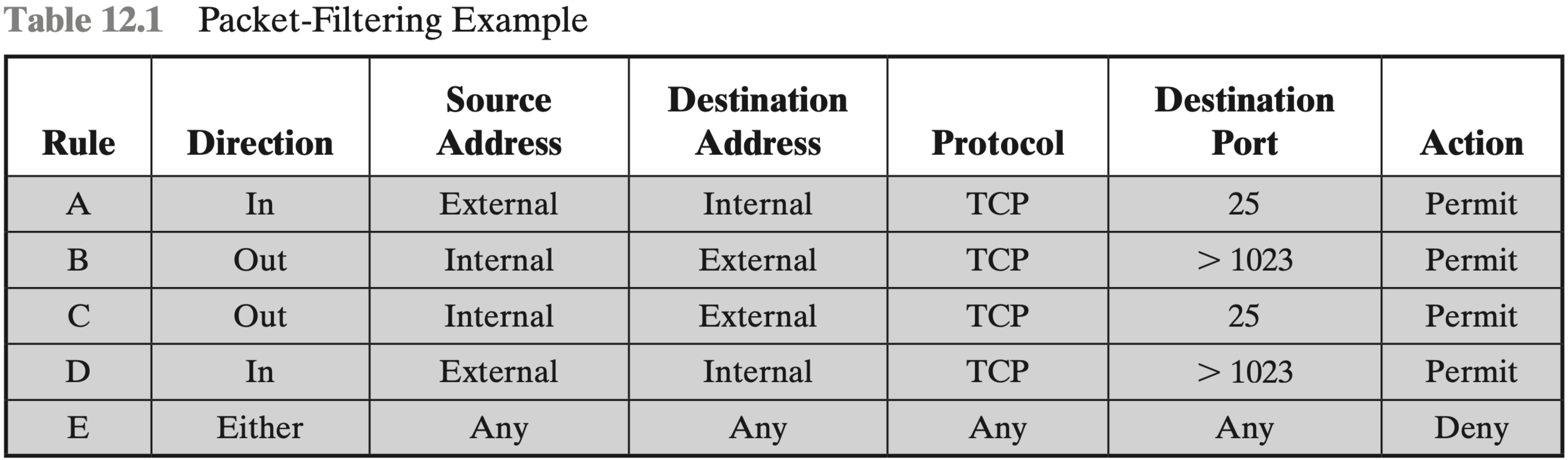


Table 12.1 Packet-Filtering Example

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

Inbound mail from an external source is allowed (port 25 is for SMTP incoming).

This is an explicit statement of the default policy.
All rulesets include this rule implicitly as the last rule.

Example

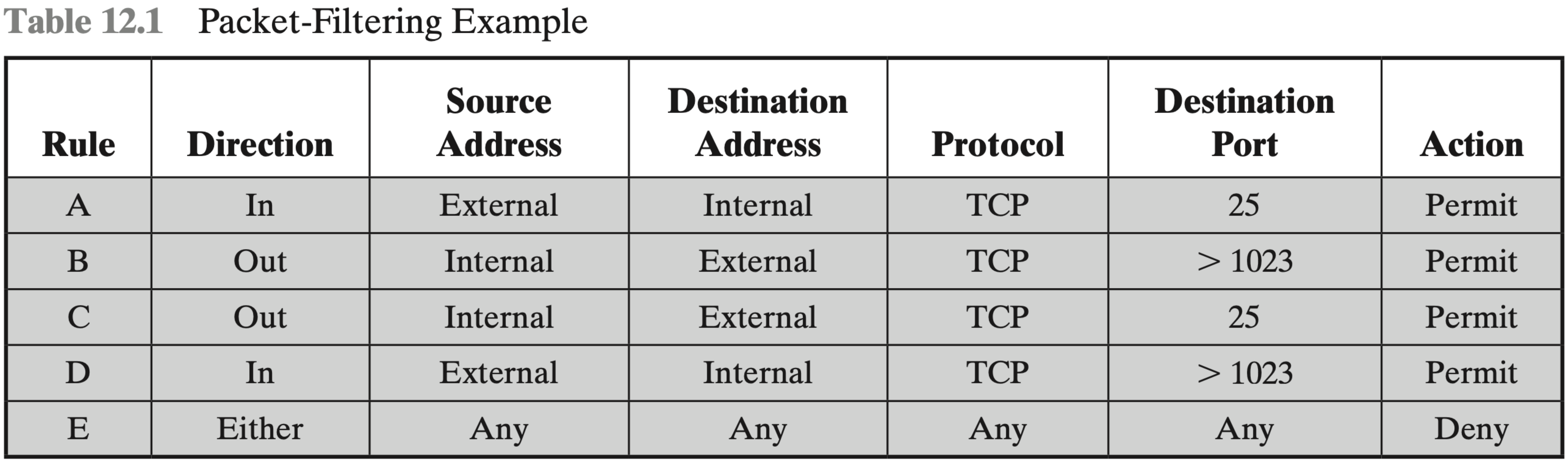
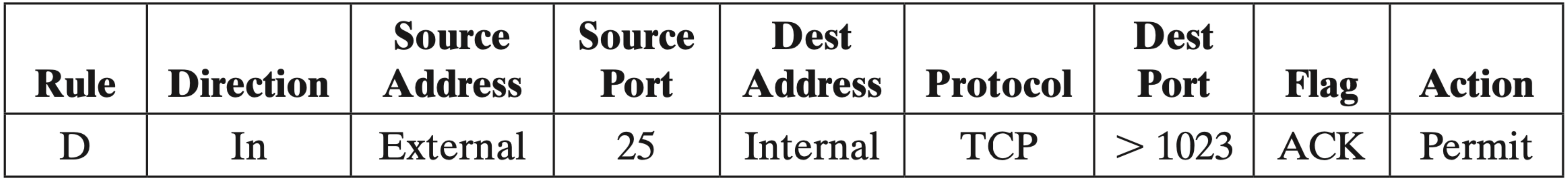


Table 12.1 Packet-Filtering Example

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

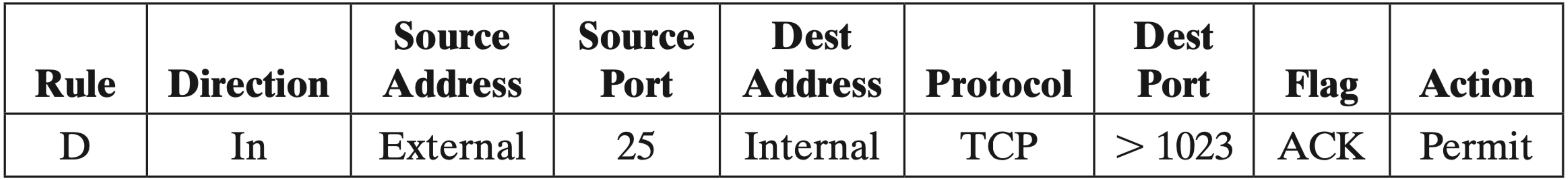


Rule	Direction	Source Address	Source Port	Dest Address	Protocol	Dest Port	Flag	Action
D	In	External	25	Internal	TCP	> 1023	ACK	Permit

A: Rule D allows external traffic to any destination port above 1023.

A: An attacker could gain access to internal machines by sending packets with a TCP source port number of 25.

Example



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

A: Rule D allows external traffic to any destination port above 1023.

To counter this attack, the firewall ruleset can be configured with a source port field for each row

A: An attacker could gain access to internal machines by sending packets with a TCP source port number of 25.

To counter this attack, we can add an ACK flag field to each row.

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?

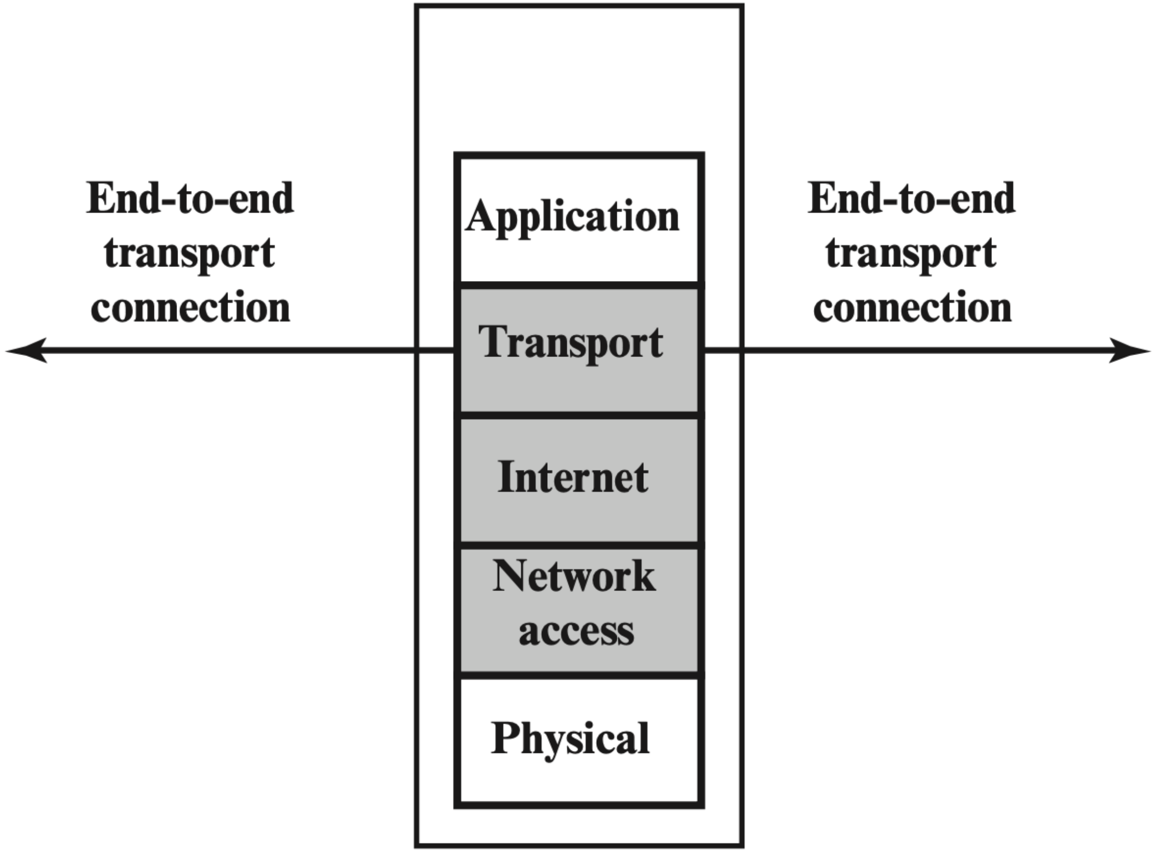
What about spoofing

- 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

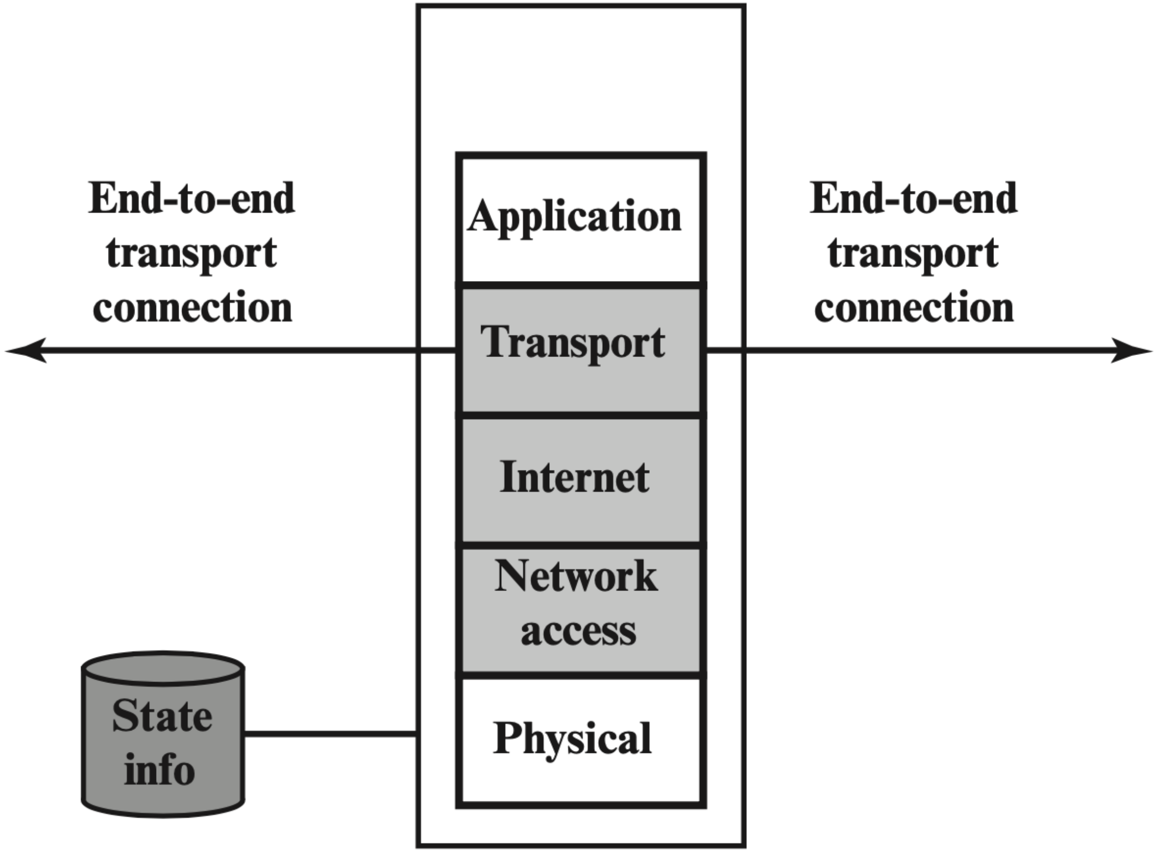
Q: Does this eliminate spoofed traffic completely?

A: No. Internal spoofing may still occur.

→ Packet filtering does not use state.


End-to-end transport connection
Application
Transport
Internet
Network access
Physical
End-to-end transport connection

(b) Packet filtering firewall

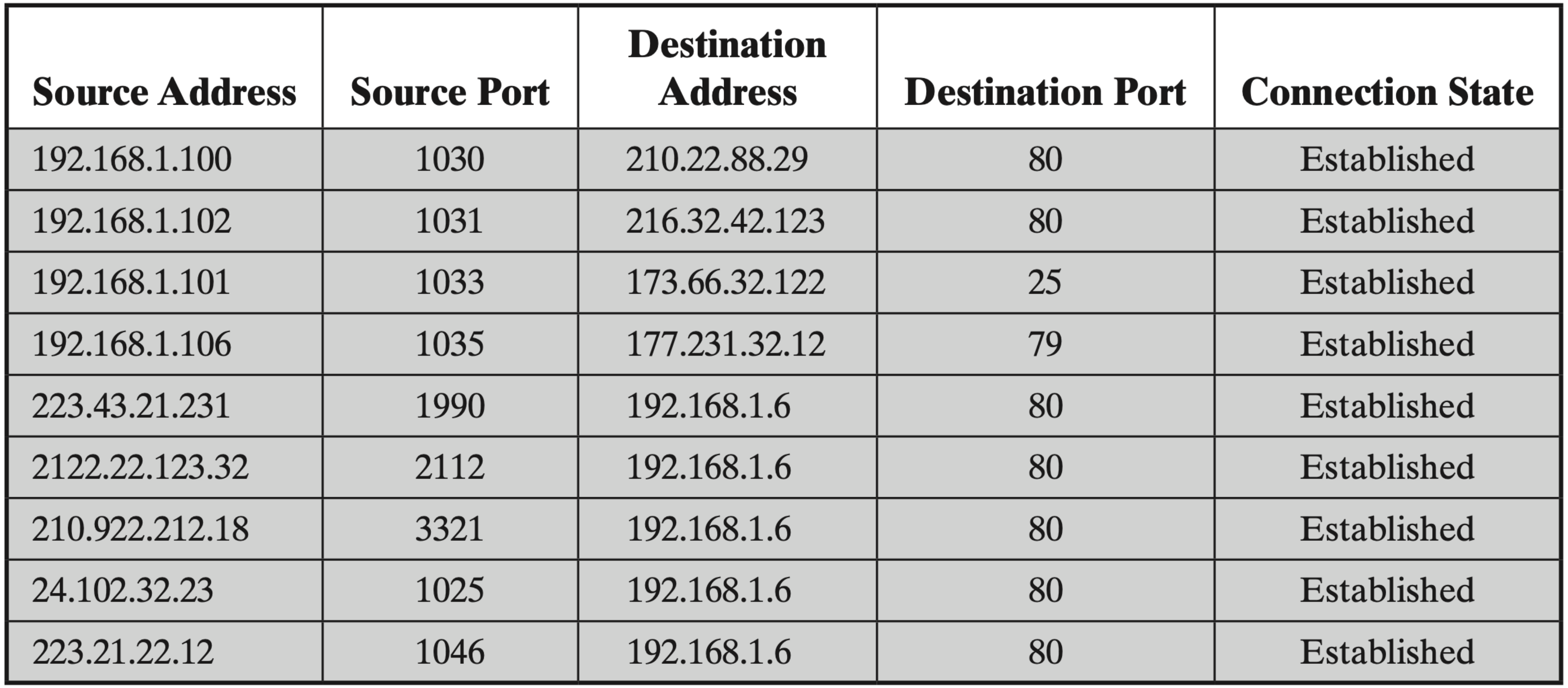

End-to-end transport connection
Application
Transport
Internet
Network access
Physical
State info
End-to-end transport connection

(c) Stateful inspection firewall

Stateful inspection firewalls

- More expensive than packet filtering
- Keep state 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)



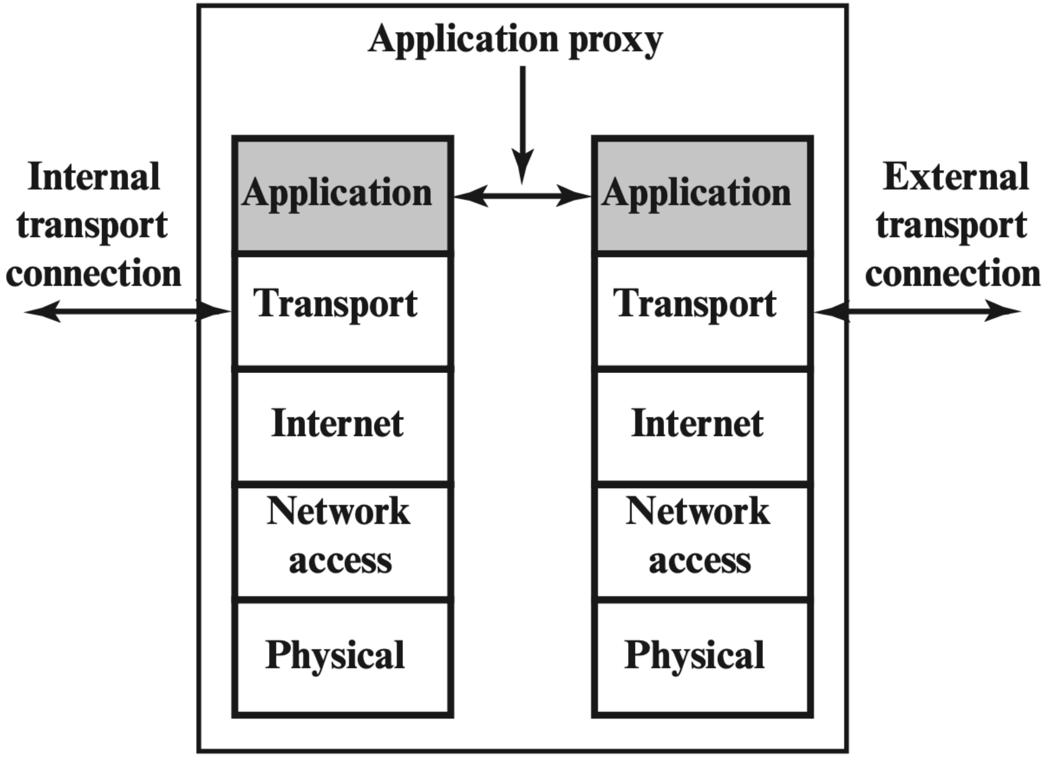
Source Address	Source Port	Destination Address	Destination Port	Connection State
192.168.1.100	1030	210.22.88.29	80	Established
192.168.1.102	1031	216.32.42.123	80	Established
192.168.1.101	1033	173.66.32.122	25	Established
192.168.1.106	1035	177.231.32.12	79	Established
223.43.21.231	1990	192.168.1.6	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

Stateful inspection firewalls

- Can stop attacks that packet filtering will not stop
- Have to store information about packets instead of doing it at the hardware level.
- This does not prevent insider attacks, e.g., using tunnel to get through the firewall(more details later).

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
  - Simulates the app to catch malicious behavior
  - Intercepting proxy requires no explicit configuration by client (or knowledge of this filtering by client)
  - All other traffic is blocked



(d) Application proxy firewall

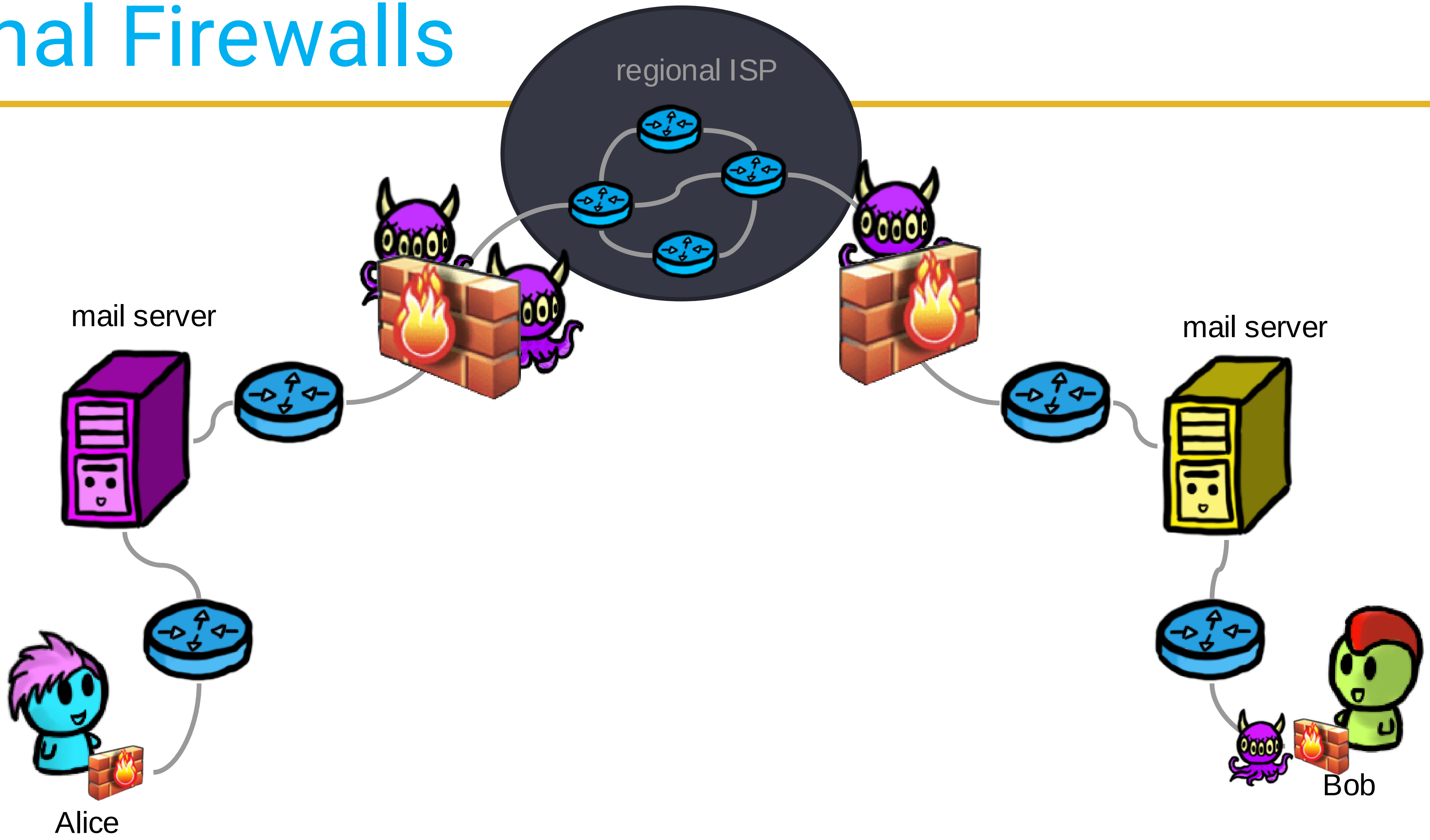
Application Proxy

- 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



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
  - Implements user-defined policy

- 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?
      → Possible leaks of private data (e.g., passwords). Malware uploading data.
      → Prevent malware from making outbound connections, keystroke.

Segmentation and separation

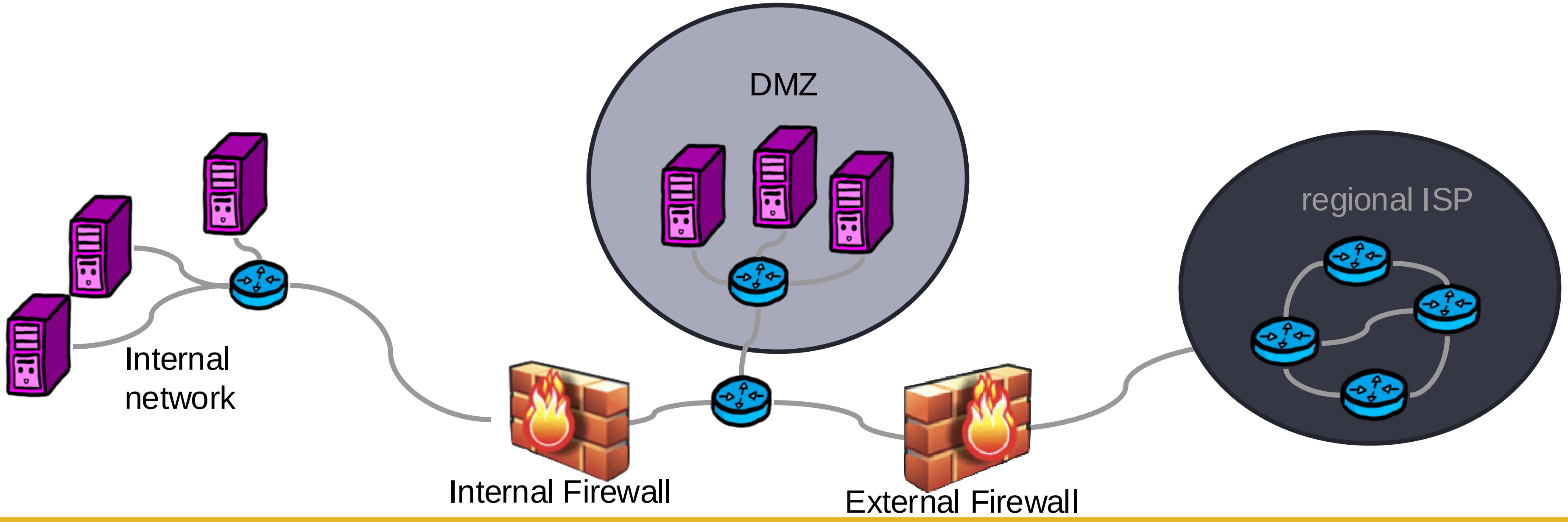
- Don’t put all a company’s servers on a single machine
  - Deploy them on **multiple** machines, depending on their functional and access requirements

- 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

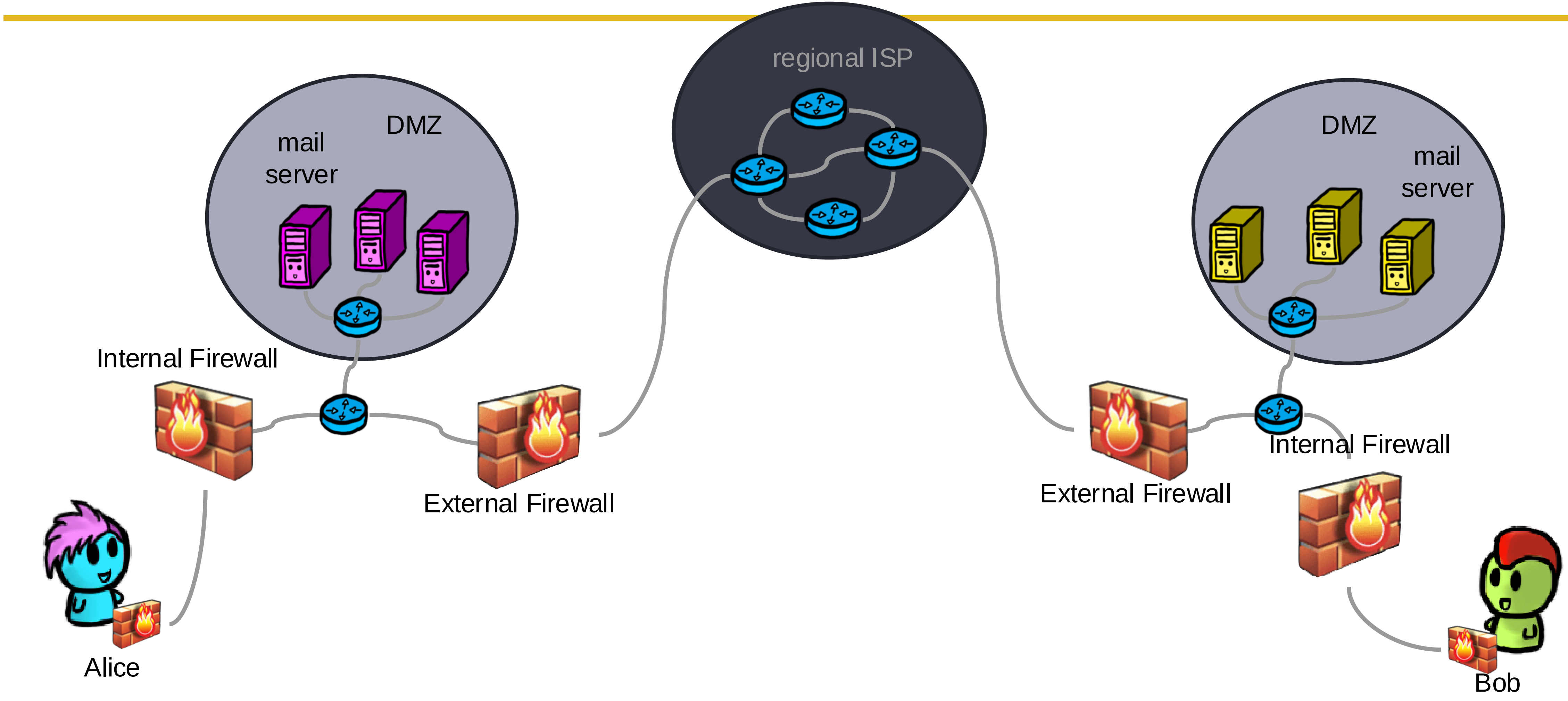
- Therefore, it shouldn’t be trusted by other servers of the company, and it should be deployed outside the company firewall

Demilitarized Zone (perimeter network)

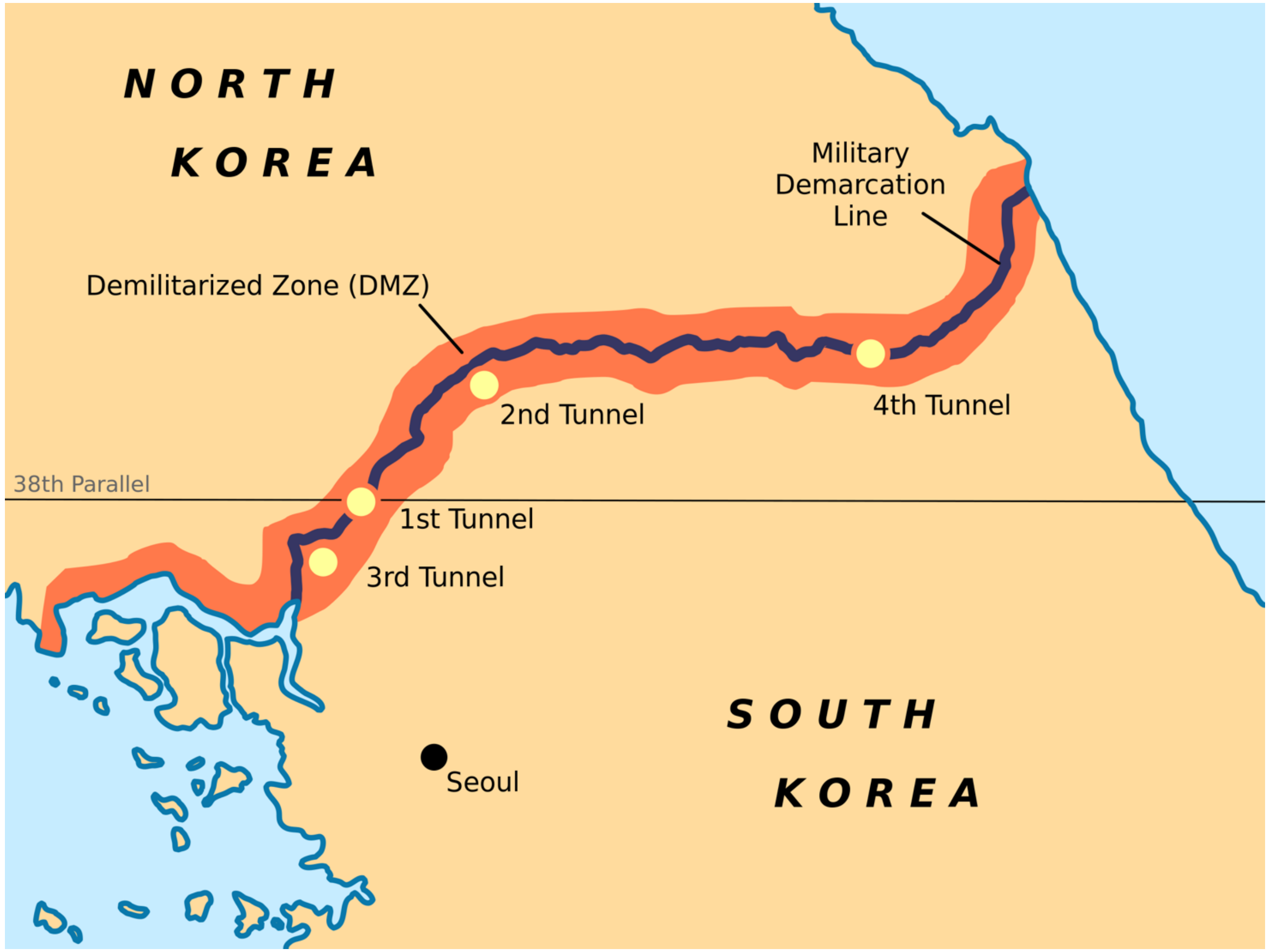
- Subnetwork that contains an organization’s external services, accessible to the Internet
  - E.g. Webserver, email sever, DNS server
- Deploy external and internal firewall
  - External firewall protects DMZ
  - Internal firewall protects internal network from attacks lodged in DMZ



DMZ 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?

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 defense

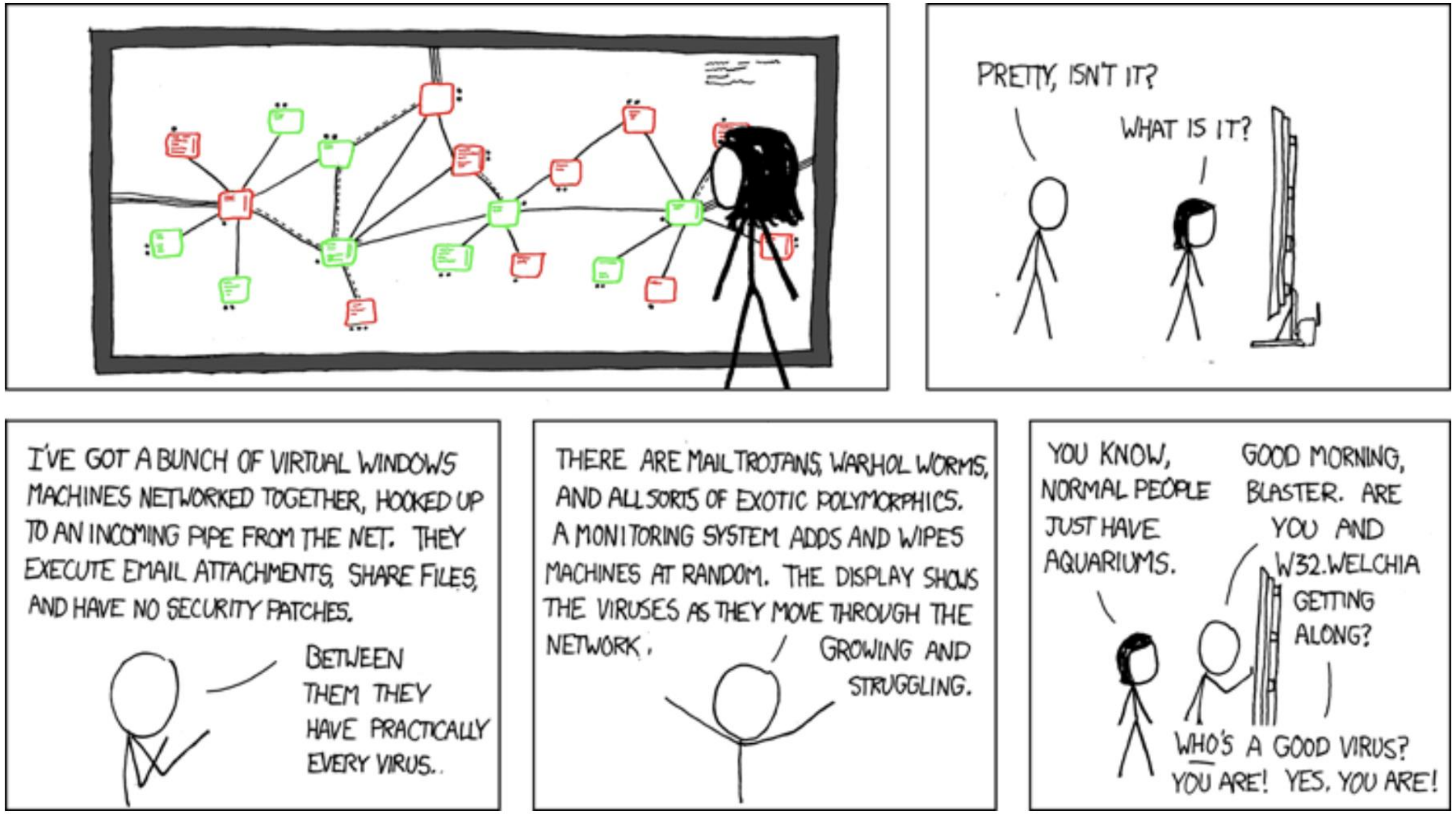
Intrusion Detection Advantages

- 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

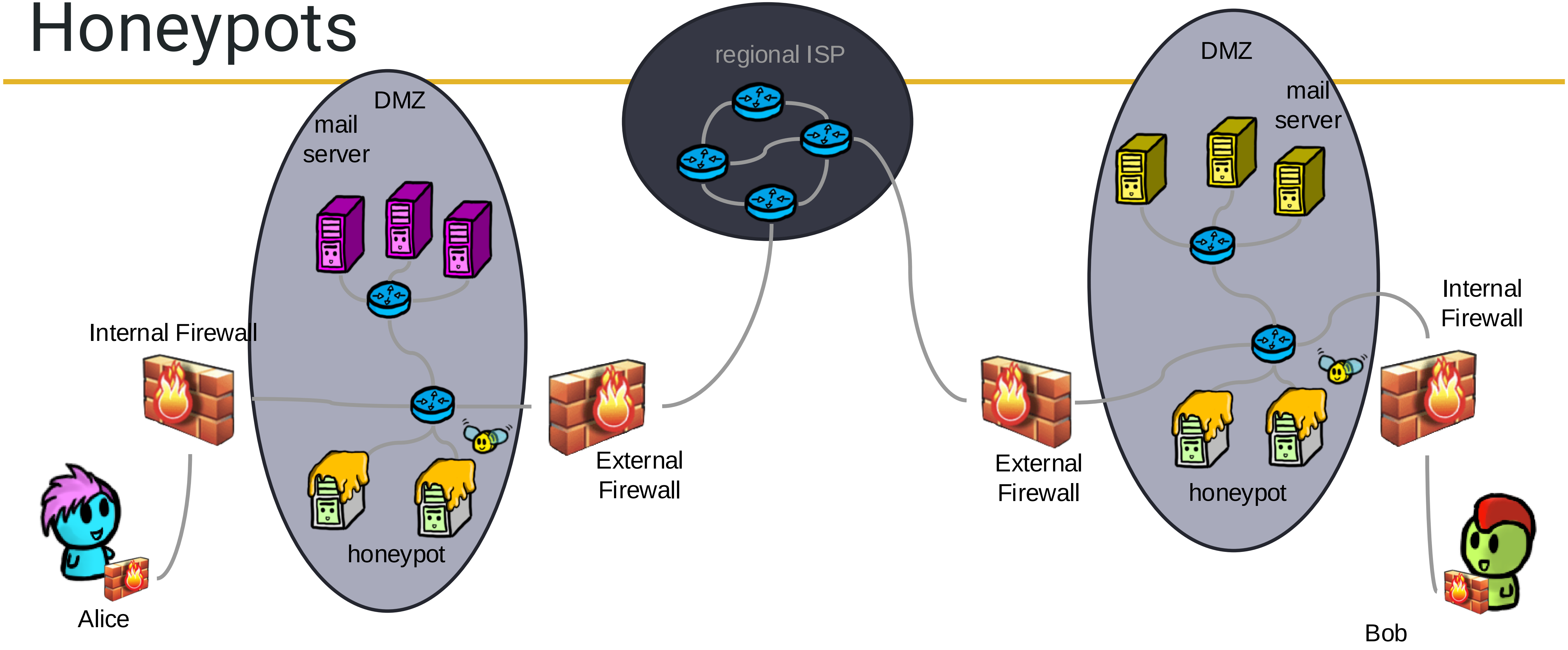
- Set up an (unprotected diversion) computer or an entire network as a trap for an attacker
- System has no production value, so any activity is suspicious
    - Any received email is considered spam
- Observe attacker to learn about new attacks, to identify and stop attacker, or to divert attacker from 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


PRETTY, ISN'T IT?
WHAT IS IT?
I'VE GOT A BUNCH OF VIRTUAL WINDOWS MACHINES NETWORKED TOGETHER, HOOKED UP TO AN INCOMING PIPE FROM THE NET. THEY EXECUTE EMAIL ATTACHMENTS, SHARE FILES, AND HAVE NO SECURITY PATCHES.
BETWEEN THEM THEY HAVE PRACTICALLY EVERY VIRUS.
THERE ARE MAIL TROJANS, WARHOL WORMS, AND ALL SORTS OF EXOTIC POLYMORPHICS. A MONITORING SYSTEM ADDS AND WIPES MACHINES AT RANDOM. THE DISPLAY SHOWS THE VIRUSES AS THEY MOVE THROUGH THE NETWORK,
GROWING AND STRUGGLING.
YOU KNOW, NORMAL PEOPLE JUST HAVE AQUARIUMS.
GOOD MORNING, BLASTER. ARE YOU AND W32.WELCHIA GETTING ALONG?
WHO'S A GOOD VIRUS? YOU ARE! YES, YOU ARE!

https://xkcd.com/350/

Honeypots


regional ISP
DMZ
mail server
Internal Firewall
External Firewall
honeypot
Alice
DMZ
mail server
Internal Firewall
External Firewall
honeypot
Bob

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 behavior by attacker

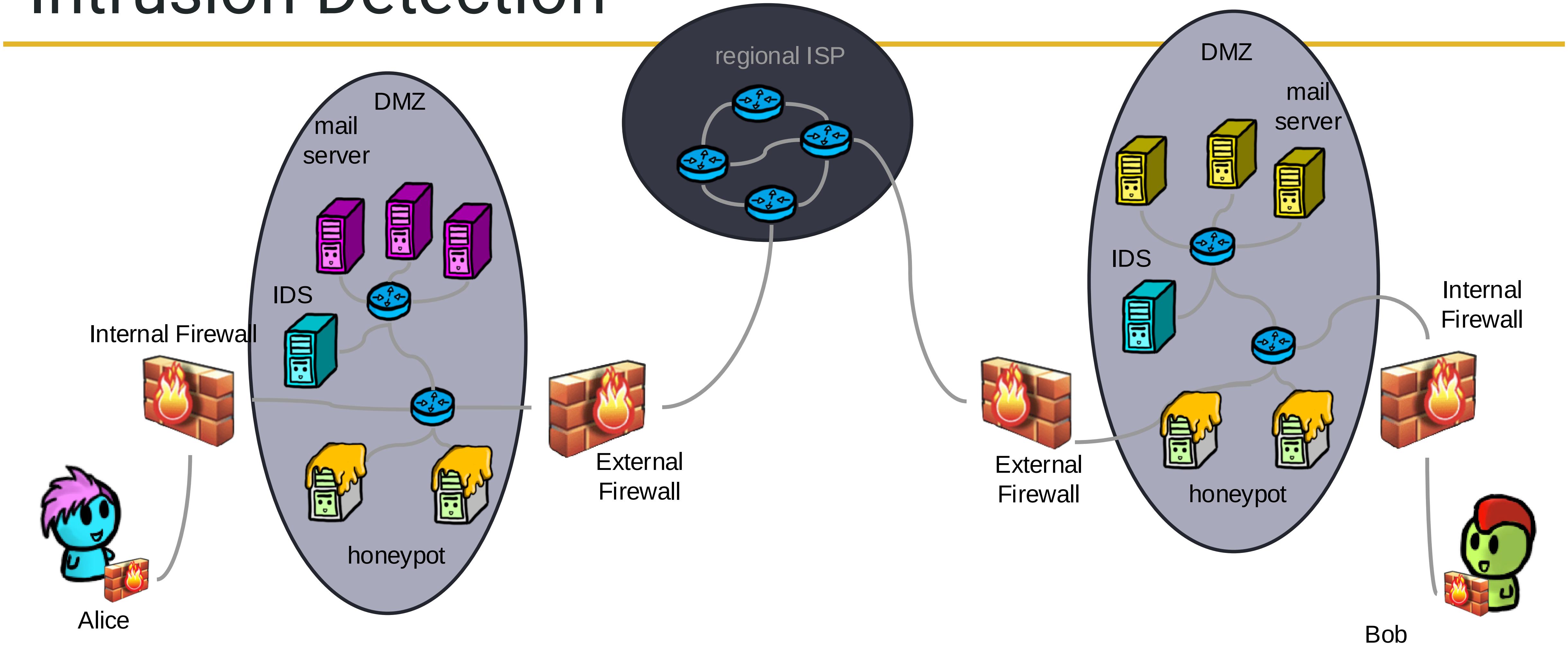
Types of Honeypots

- 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



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

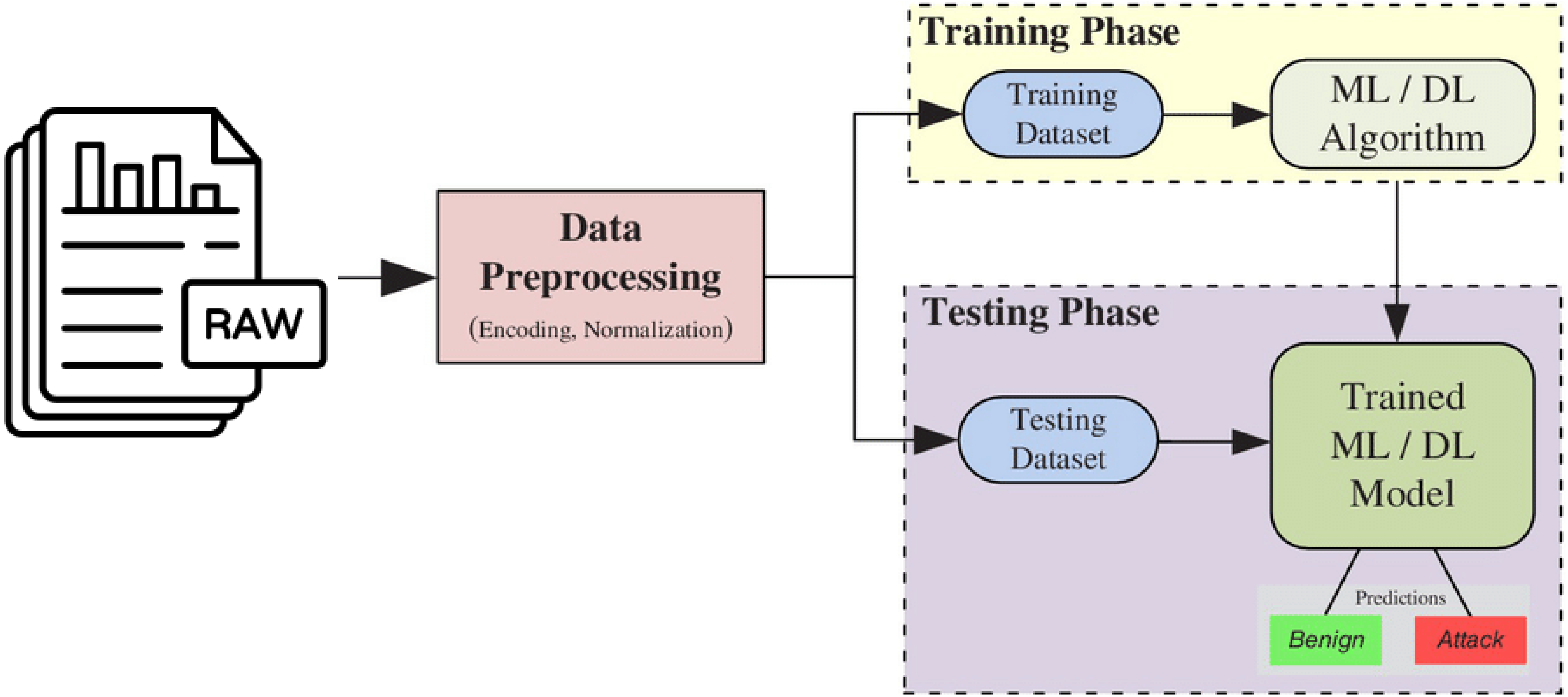
Signature-based IDSs

- 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 behavior that is out of the ordinary
- By modelling good behavior and raising alert when system activity no longer resembles this model
- Or by modelling bad behavior 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

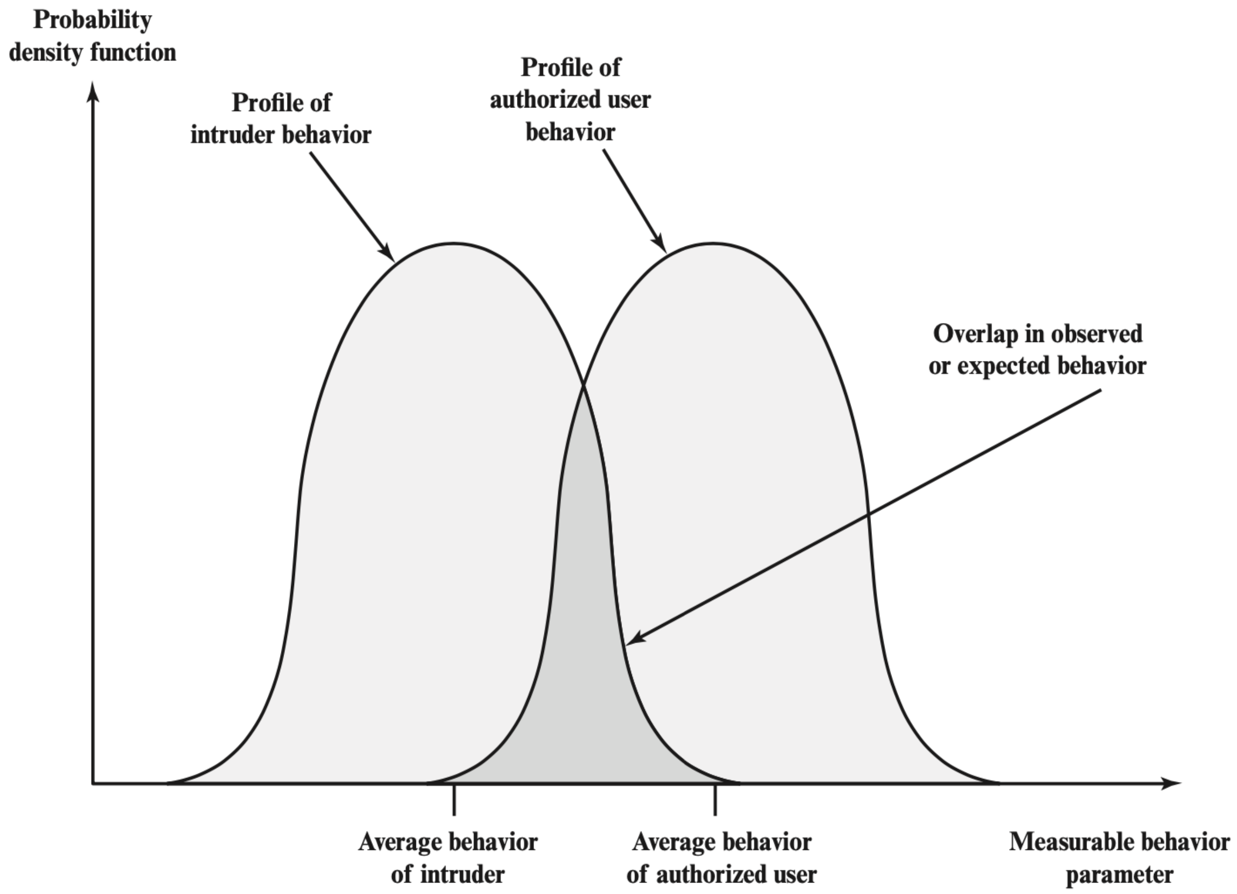
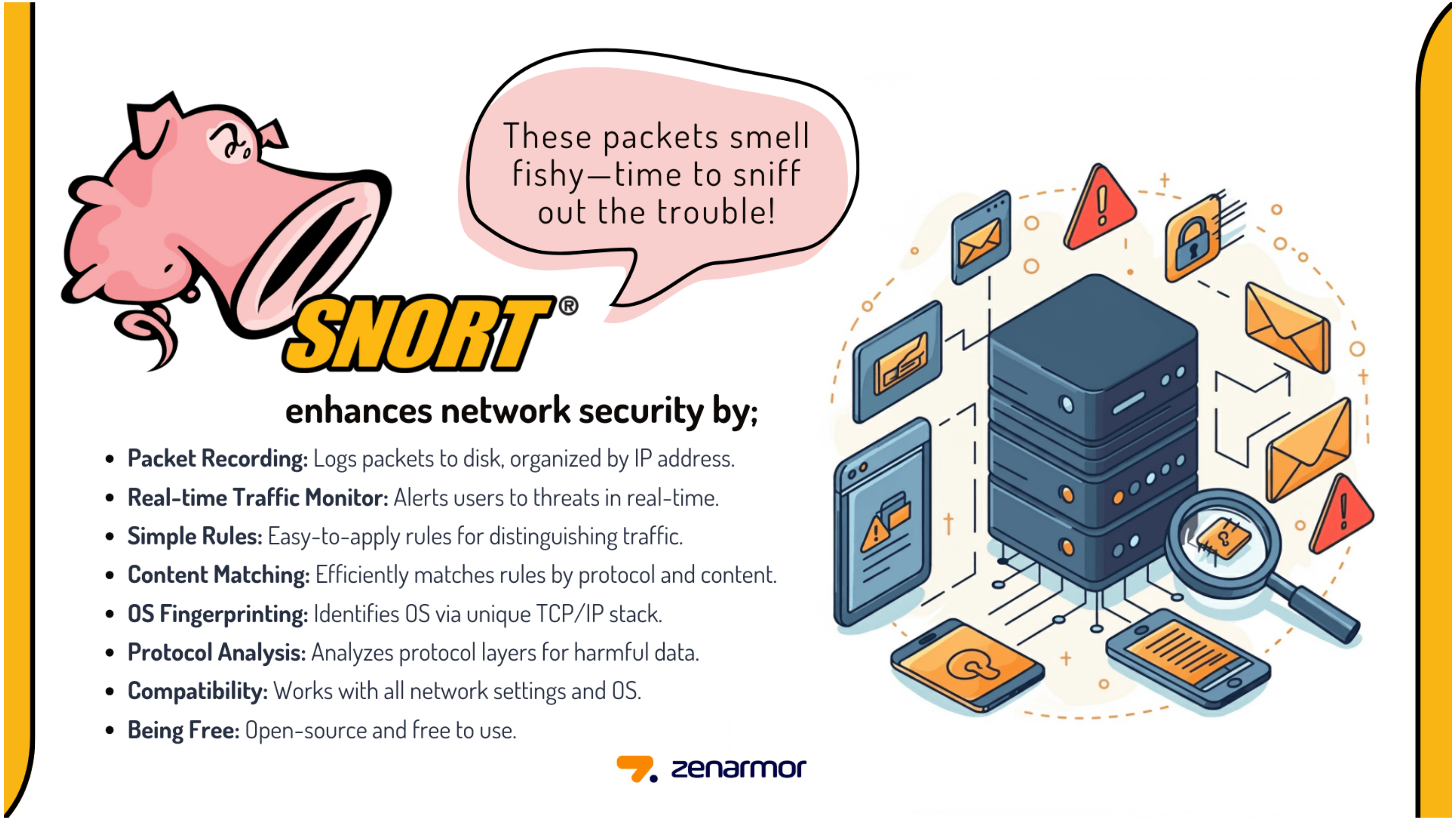


Figure 11.1 Profiles of Behavior of Intruders and Authorized Users

Example: Snort

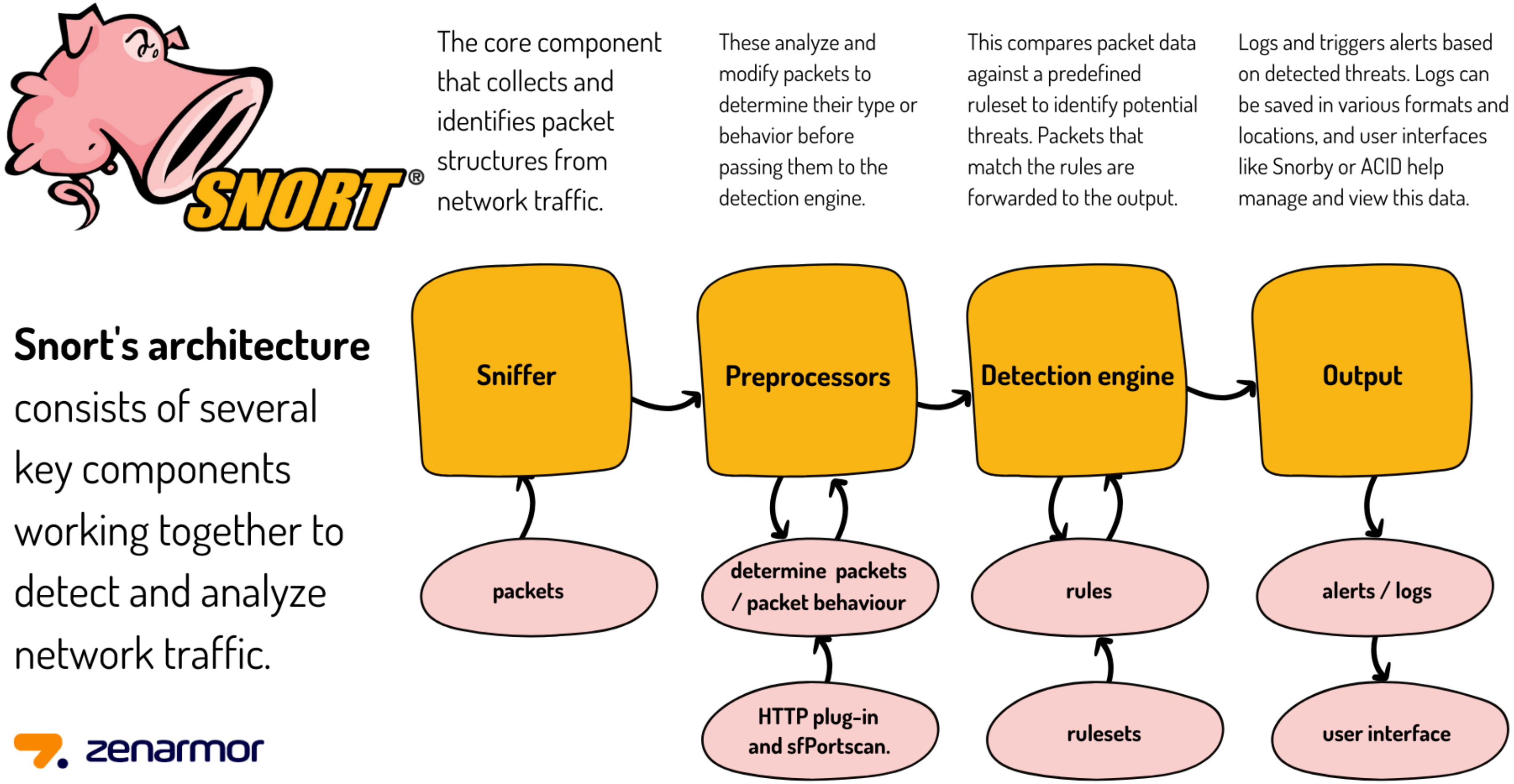


enhances network security by;

- **Packet Recording:** Logs packets to disk, organized by IP address.
- **Real-time Traffic Monitor:** Alerts users to threats in real-time.
- **Simple Rules:** Easy-to-apply rules for distinguishing traffic.
- **Content Matching:** Efficiently matches rules by protocol and content.
- **OS Fingerprinting:** Identifies OS via unique TCP/IP stack.
- **Protocol Analysis:** Analyzes protocol layers for harmful data.
- **Compatibility:** Works with all network settings and OS.
- **Being Free:** Open-source and free to use.

zenarmor

Example: Snort



Snort's architecture consists of several key components working together to detect and analyze network traffic.

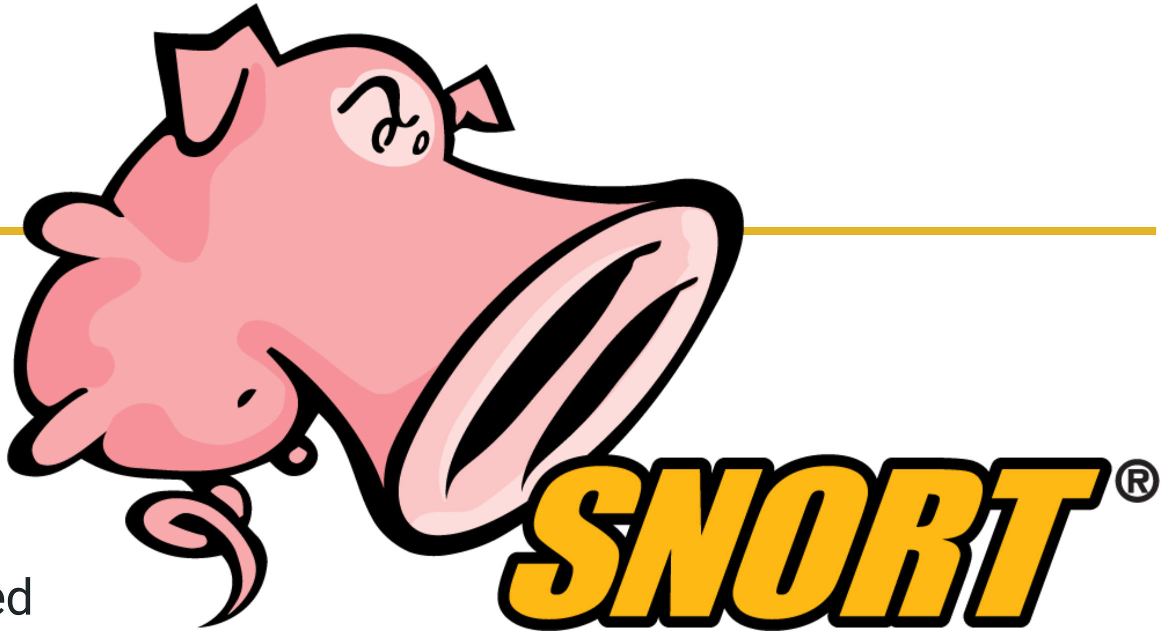
The core component that collects and identifies packet structures from network traffic.

These analyze and modify packets to determine their type or behavior before passing them to the detection engine.

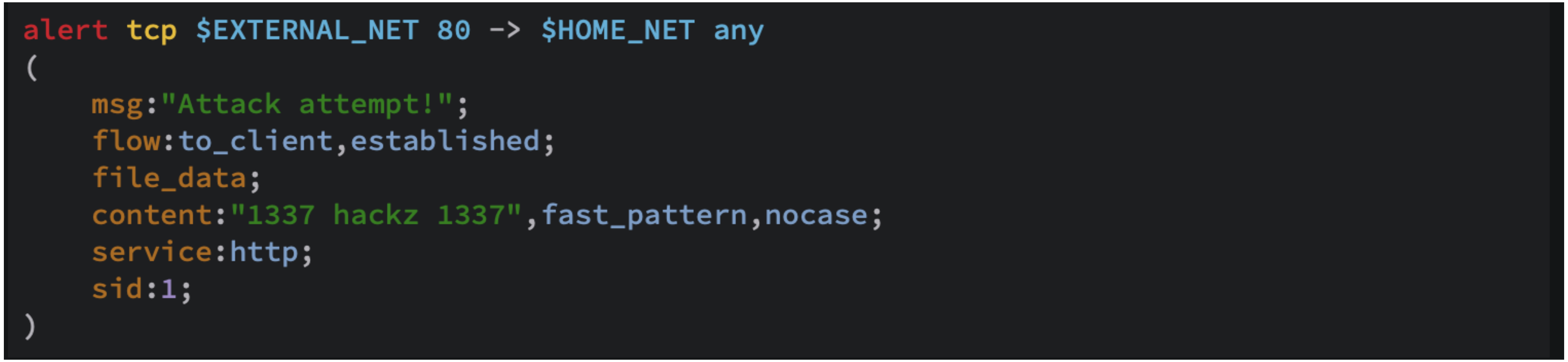
This compares packet data against a predefined ruleset to identify potential threats. Packets that match the rules are forwarded to the output.

Logs and triggers alerts based on detected threats. Logs can be saved in various formats and locations, and user interfaces like Snorby or ACID help manage and view this data.

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?