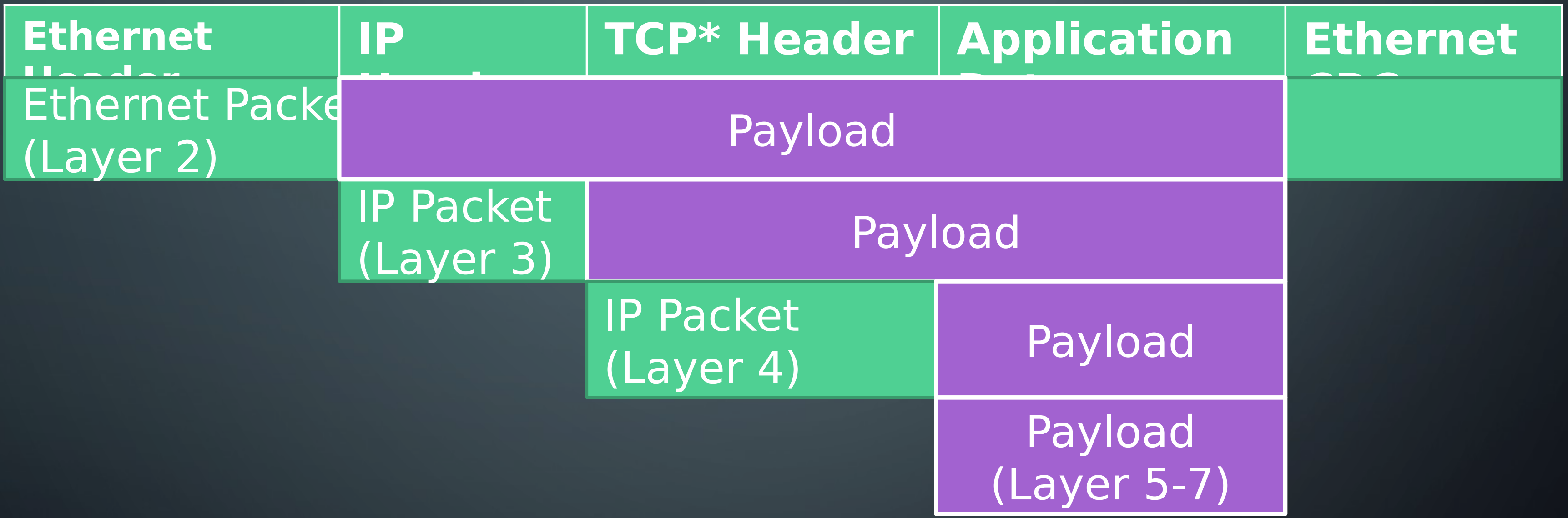
LINUX NETWORKING

PREVIOUSLY COVERED

- What is an IP?
- What is a subnet?
- CIDR notation

- Focus on IPv4

A TYPICAL PACKET



Ethernet Header

IP Header

TCP* Header

Application Data

Ethernet CRC

Ethernet Packe

(Layer 2)

Payload

IP Packet

(Layer 3)

Payload

IP Packet

(Layer 4)

Payload

Payload

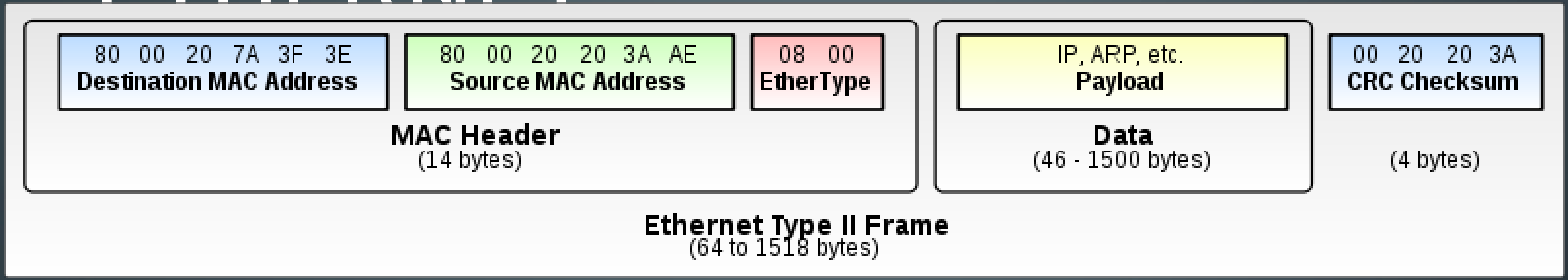
(Layer 5-7)

* Could be TCP, UDP, ICMP, or other protocols that ride on IP.

MTU

- Maximum Transmission Unit
  - Maximum size a layer can pass forward without having to break up the packet (fragmentation)
    - Ethernet is 1500bytes
    - 802.11 is 2272bytes
    - Jumbo Frames is 1500-9000bytes
- Ethernet Efficiency
  - Efficiency = $\frac{Payload\_Size}{Frame\_Size}$ $\frac{1500}{1538} = 97.53\%$ or 97.5Mbps on a 100Mbps connection

DATA LINK - LAYER 2 - ETHERNET



80 00 20 7A 3F 3E
Destination MAC Address

80 00 20 20 3A AE
Source MAC Address

08 00
EtherType

MAC Header
(14 bytes)

IP, ARP, etc.
Payload

Data
(46 - 1500 bytes)

00 20 20 3A
CRC Checksum

(4 bytes)

Ethernet Type II Frame
(64 to 1518 bytes)

- Layer 2
- Typically 14 byte header
  - 6 byte destination address
  - 6 byte source address
  - 2 bytes for type
    - IP, IPv6, ARP, etc.
- Addresses must be unique
  - First 3 bytes represent manufacture
  - Burnt in during manufacturing – can be overridden (or spoofed)
- Special Addresses
  - Broadcast Address – FF:FF:FF:FF:FF:FF

DATA LINK HARDWARE – HUBS AND SWITCHES

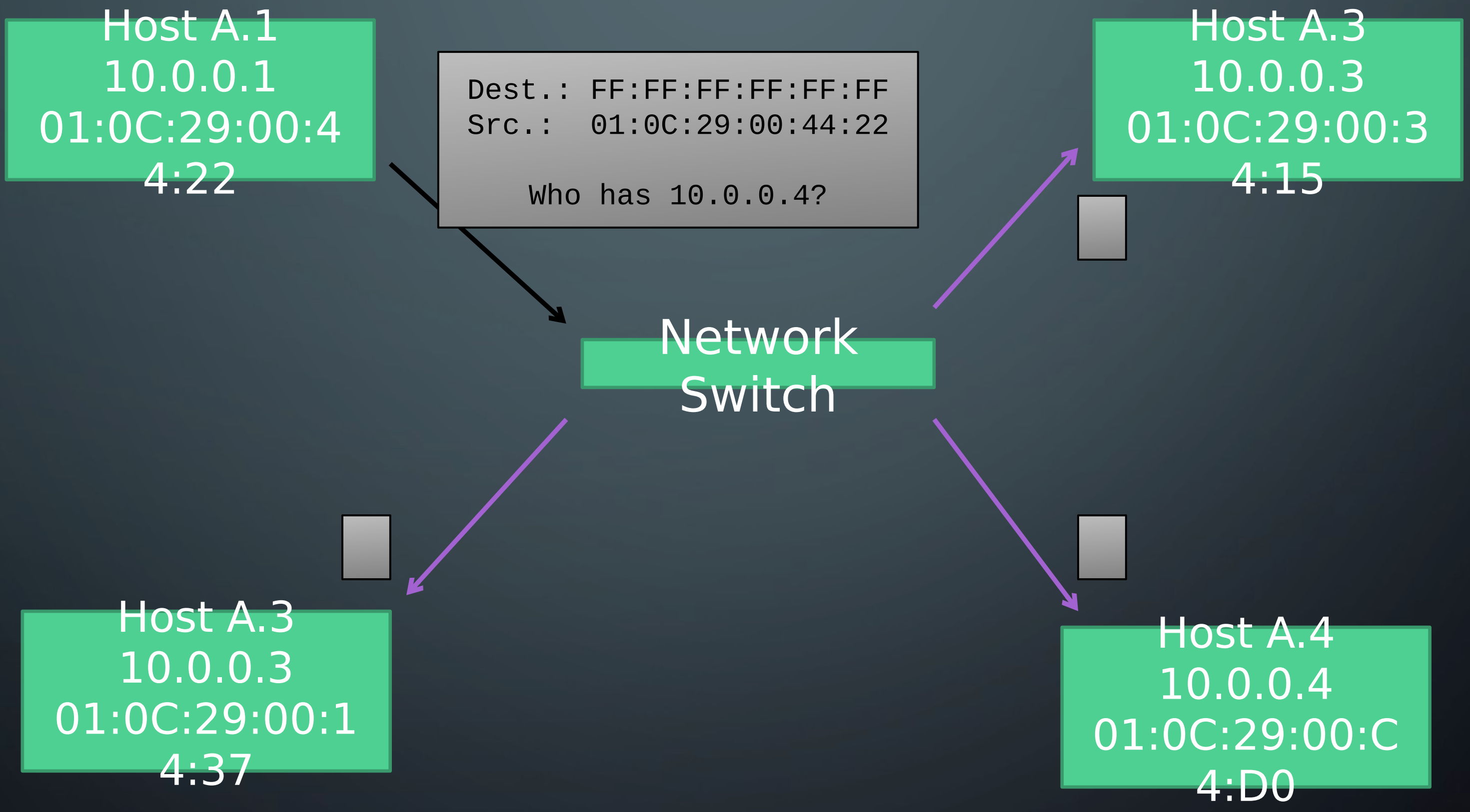
- Hubs
  - Send all packets to everybody
    - Not very secure
  - Shared Bandwidth
- Switches are smart hubs
  - Maintain MAC address list for each port
  - Dedicated port bandwidth

ARP

- Address Resolution Protocol
  - Resolve IP addresses to MAC addresses
    - Broadcasts who has IP to network
    - IP holder responds via senders MAC address

- Hubs and switches can only route MAC addresses
  - No knowledge of IP

ARP EXAMPLE - REQUEST



Host A.1
10.0.0.1
01:0C:29:00:44:22

```
Dest.: FF:FF:FF:FF:FF:FF
Src.:  01:0C:29:00:44:22

   Who has 10.0.0.4?
```

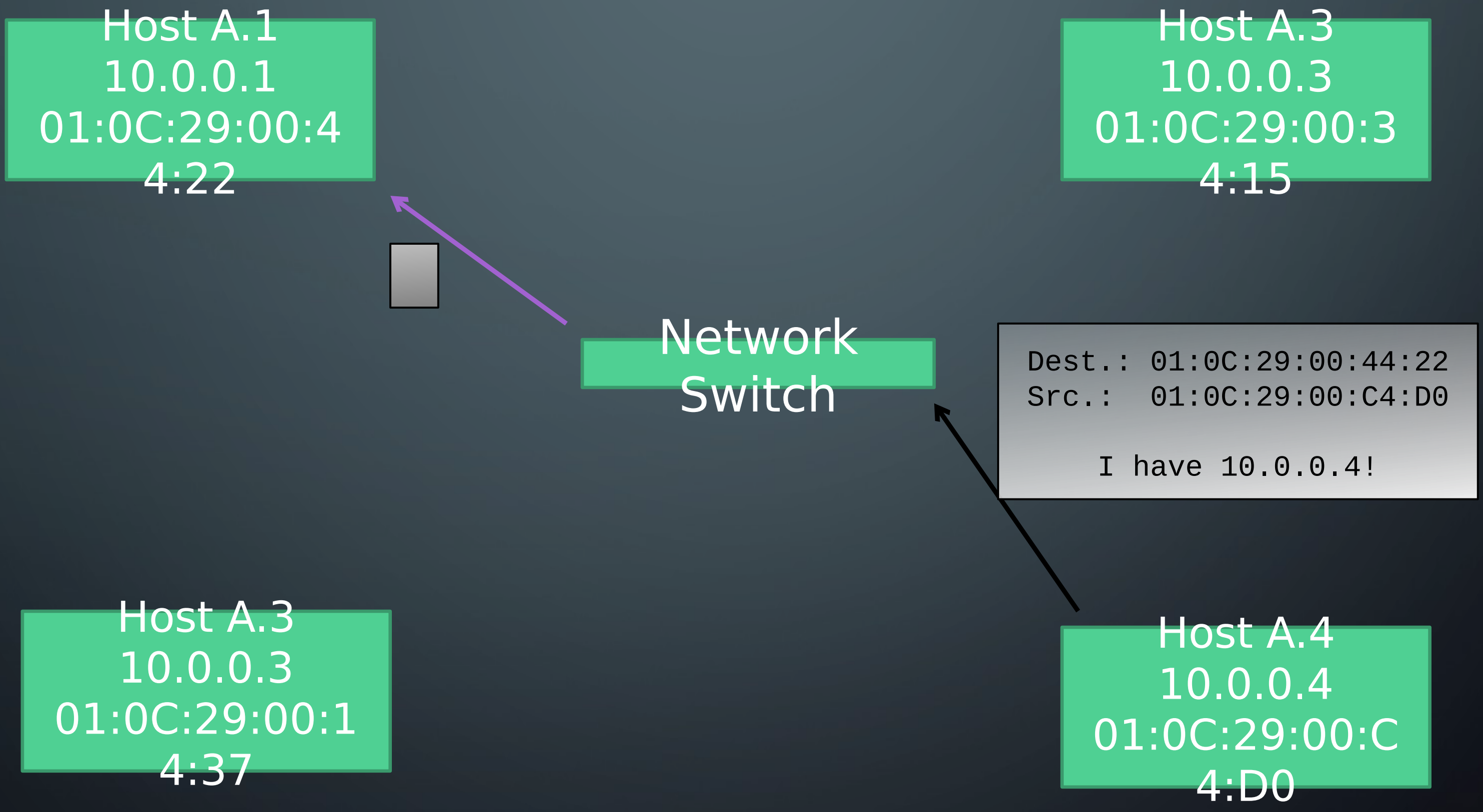
Network Switch

Host A.3
10.0.0.3
01:0C:29:00:34:15

Host A.3
10.0.0.3
01:0C:29:00:14:37

Host A.4
10.0.0.4
01:0C:29:00:C4:D0

ARP EXAMPLE - RESPONSE



Host A.1
10.0.0.1
01:0C:29:00:44:22

Host A.3
10.0.0.3
01:0C:29:00:34:15

Network Switch

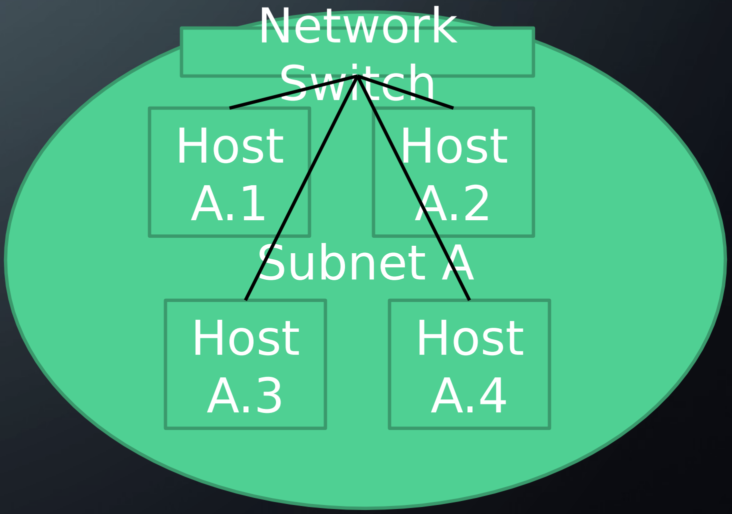
```
Dest.: 01:0C:29:00:44:22
Src.:  01:0C:29:00:C4:D0

    I have 10.0.0.4!
```

Host A.3
10.0.0.3
01:0C:29:00:14:37

Host A.4
10.0.0.4
01:0C:29:00:C4:D0

ARP/MAC

- Traditional Networks (ARP) Address Resolution Protocol
  - A.1 wants to talk to A.2
  - A.1 asks all hosts/everyone (**broadcasts**) what MAC is A.2?
  - A.2 **Broadcasts** back answer
  - A.1 sends packet to A.2 with A.2's MAC address (otherwise switch wouldn't know which network port to send it to)



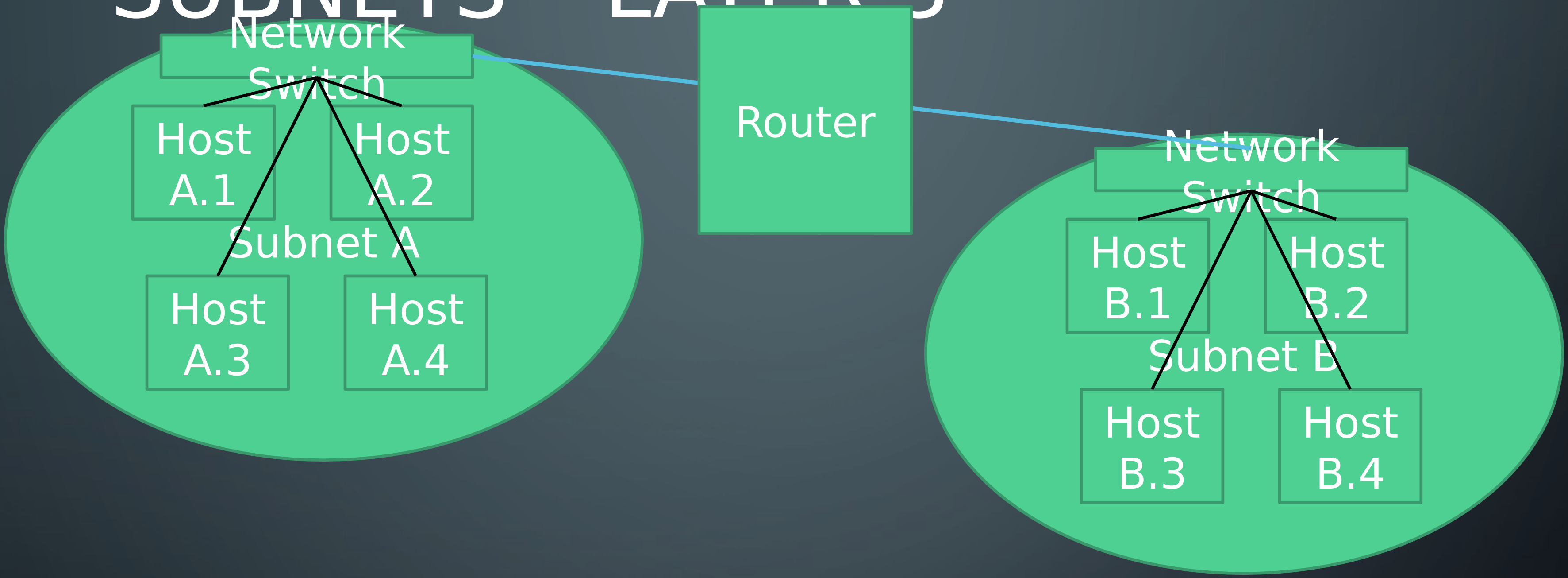
ARP RESPONSE - SECURITY

- Nothing to prevent other hosts from answering
  - First to respond wins
  - Can create Man in the Middle
- Switches can only remember finite number of MAC addresses (4k?)
  - If too many, switch can failsafe revert to hubs
  - MAC flood to create this situation
- Advanced switches can prevent this
  - $50 12 port switch vs. a $5k one.

ARP

- Works good, but what about large networks?
- Each host receives broadcasts
  - Must check if message is meant for host
- More hosts means more broadcast
  - Eventually run out of host system resources
- Need a way to segment networks

SUBNETS – LAYER 3



- A.1 wants to talk to B.1
  - A.1 sees B.1’s IP is not on local subnet
  - A.1 sends data packet to default Router to route it
  - Router received packet and ARP process begins on B subnet

** Some additional ARPing may initially occur between A.1 and router (assuming cached)

SUBNETTING

- A subnet is a sub-network
  - A range of IP addresses
  - Defined by a subnet

- 10.0.20.0/24 – Subnet is 256 hosts
- 10.0.20.0/23 – Subnet is 512 hosts

WHAT IS A ROUTER?

- Router – Forwards data between compute networks beyond directly connected devices.
  - Connects multiple subnets together
  - (Slide 15)

- Devices are directly connected when data is forwarded using network switches.

ROUTER [GATEWAY IS A ROUTER]

- Routes traffic
  - Can be static routes (this is what we’ll use)
  - Can dynamically build routes
    - Self healing, load balancing, scalable, etc.

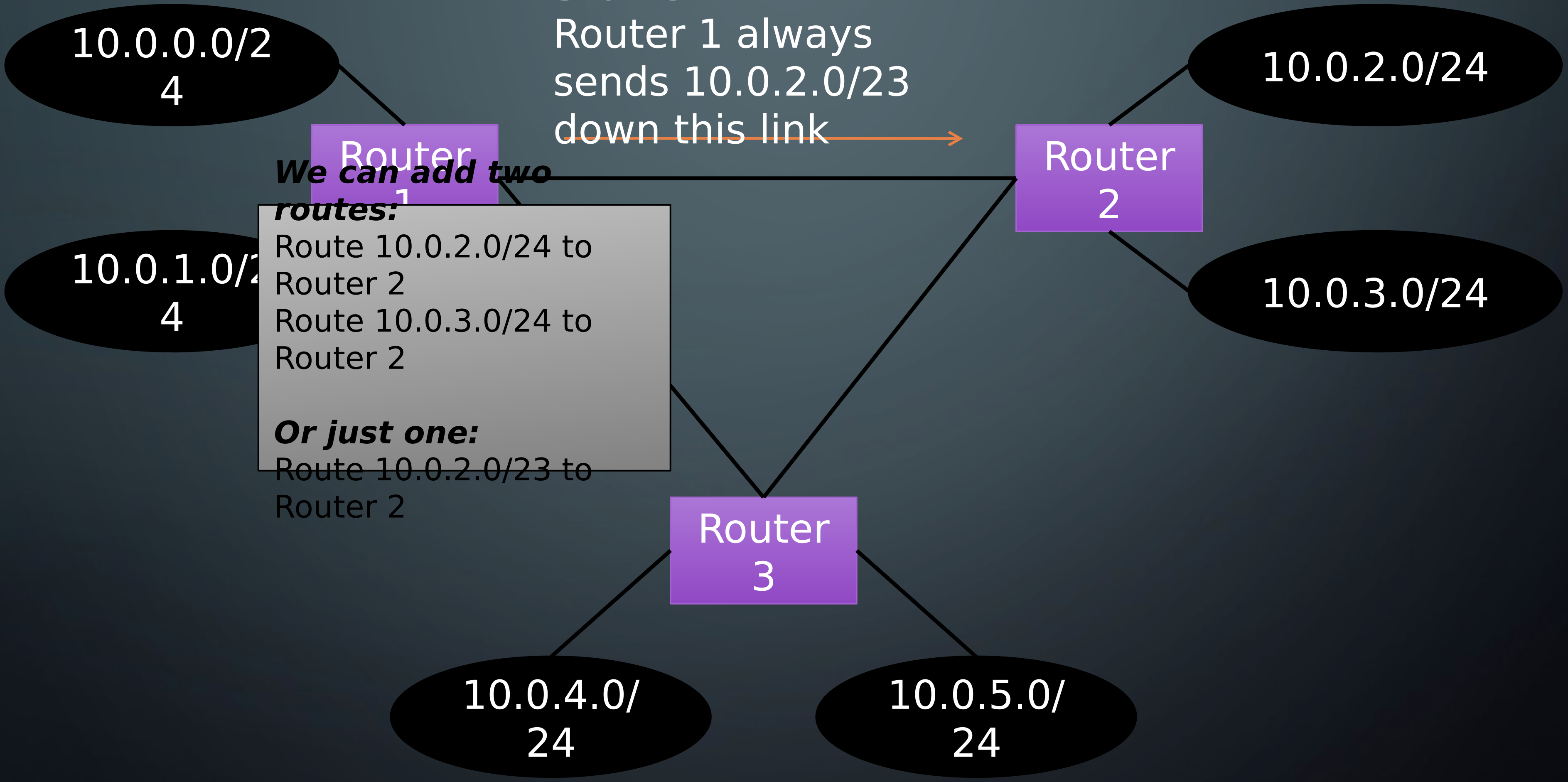
- If you want to send to an IP address not on your subnet (defined by subnet mask) you will need a router to send it for you
  - Can have a default router (only one)
  - Can have static routes to override
  - netstat -rn shows default routing table
    - Or: ip route show

ROUTING TABLE: NETSTAT -RN

```
user@router:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
128.198.50.16   0.0.0.0         255.255.255.248 U         0 0          0 eth0
10.0.5.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.7.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.0.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.3.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.9.0        0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.11.0       0.0.0.0         255.255.255.0   U         0 0          0 eth1
10.0.12.0       10.0.0.106      255.255.254.0   UG        0 0          0 eth1
0.0.0.0         128.198.50.17   0.0.0.0         UG        0 0          0 eth0
```

- Routes are processed in order – default route last
- Mask 0.0.0.0 routes everything, but it is the last to be checked
- Almost all hosts have at least one route
  - Usually just a default route

STATIC VS. DYNAMIC



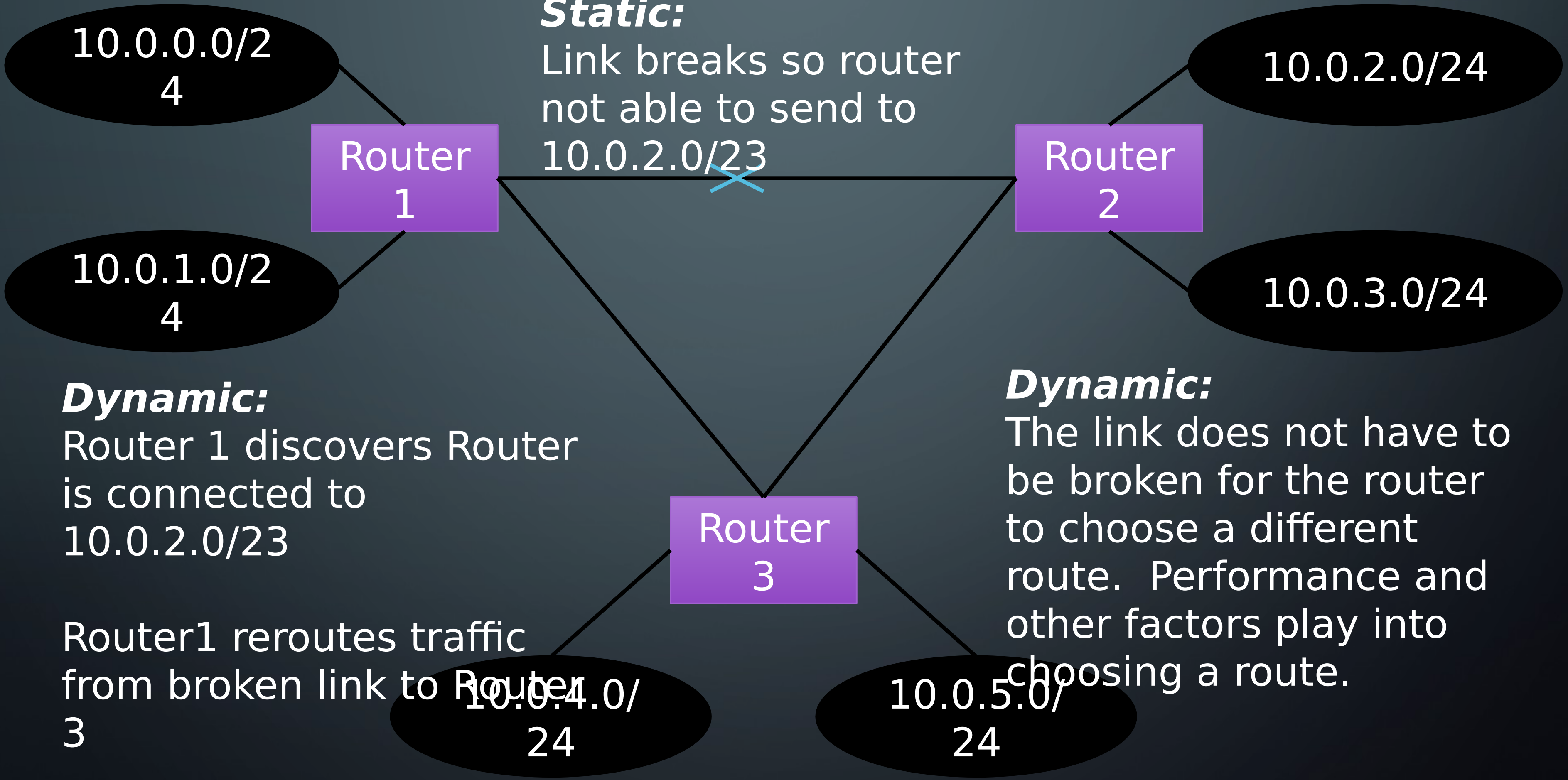
Static:
Router 1 always sends 10.0.2.0/23 down this link

We can add two routes:
Route 10.0.2.0/24 to Router 2
Route 10.0.3.0/24 to Router 2

Or just one:
Route 10.0.2.0/23 to Router 2

STATIC VS. DYNAMIC

Static:
Link breaks so router not able to send to 10.0.2.0/23



Dynamic:
Router 1 discovers Router is connected to 10.0.2.0/23

Router1 reroutes traffic from broken link to Router 3

Dynamic:
The link does not have to be broken for the router to choose a different route. Performance and other factors play into choosing a route.

RIP (Routing Information Protocol) is a way for the Routers to dynamically exchange route information.

STATIC VS. DYNAMIC

- So why choose Static?

STATIC VS. DYNAMIC

- So why choose Static?
  - It’s quick/easier.
  - It’s constant.

CREATING STATIC ROUTES

- Any traffic sent to us for a given subnet, we forward to a given IP address
- 1st: Need a subnet to route
- 2nd: Need a destination to route it to
- Examples:
  - ISP gave us 128.198.0.0/22
  - We have 6 routers and have to use one for the incoming connection.

MANUALLY ADDING ROUTES

- Default routes (gateways)

  -
    ```
    route add default gw 10.0.0.1
    ```
  - Statically routes subnet mask 0.0.0.0 or /0 to 10.0.0.1

- Static routes

  -
    ```
    route add -net 10.0.12.0 netmask 255.255.254.0 gw 10.0.0.106 dev eth0
    ```

PERSISTING STATIC ROUTES

- Edit vi /etc/network/interface and add:
  - `up route add –net 10.0.13.0/24 gw 10.0.12.137`

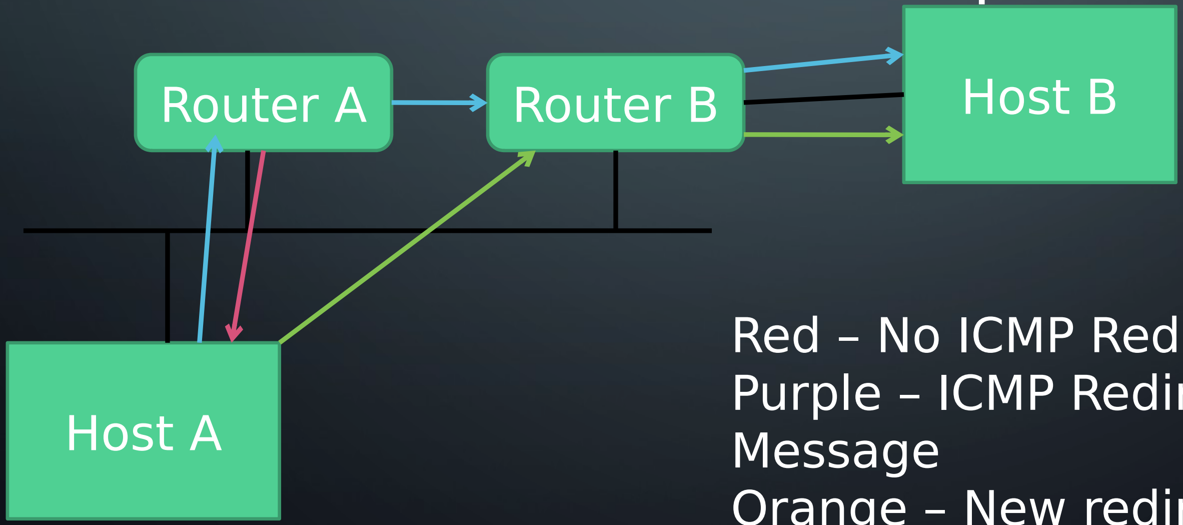
- This will apply a static route and route all 10.0.13.0 traffic the router sees to 10.0.12.137

ICMP

- Internet Control Messaging Protocol
  - Intended to complement IP
  - Not used to send data but rather host status and error messages
- Ping
  - ICMP command that queries if a host is online
  - If hosts receives a ping ‘echo request’, the host, if online, should respond with a ‘echo reply’
  - Useful for determining what hosts are online

ICMP REDIRECTS

- If a router received a packet and determines the host can route it more efficiently it sends an ICMP redirect
- Prevents excess router hops



Red – No ICMP Redirect
Purple – ICMP Redirect Message
Orange – New redirected route

HOW TO CHECK ROUTES

- tracert
  - Is able to determine the routers between it and a given destination.

- To install:
  - apt-get install traceroute
    - apt-get will be covered in later slides – this is just a reference.

HOW TO SETUP NETWORKING ON UBUNTU

- Must have a network interface

- Use lsmod to list modules inserted into the kernel
  - /etc/modules – file containing modules at boot time
  - /etc/modprobe.d – config files for modules

HOW TO SETUP NETWORKING ON UBUNTU

- Where is the network interface?
  - Can use `dmesg` to help determine interface names and link availability
  - `ifconfig -a`
  - Looking in /dev for stuff that looks right

HOW TO ADD A MACHINE TO A NETWORK?

- Assign a unique IP
- Configure host to boot up with ip address
- Add default routes
  - Allows it access to the internet
- Add a DNS server’s IP to the host
  - vi edit the /etc/resolv.conf
    - nameserver 10.0.0.1

MANUALLY ASSIGNING IP ADDRESS

- Quickly configuring IP
  - `ifconfig eth0 10.0.0.2 netmask 255.255.255.0 up`
  - `route add default gw 10.0.0.1`
- Route command adds a default static route
  - Routes subnet mask 0.0.0.0 to 10.0.0.1

PERMANENTLY ADDING IP ADDRESS

- Edit /etc/network/interface and add:

```
auto eth0
iface eth0 inet static # can be static or dhcp
address 10.0.0.2
netmask 255.255.255.0  # this is a /24
gateway 10.0.0.1       # default gateway (optional)
```

- gateway is a static route for 0.0.0.0
- gateway must exist on your local subnet

BRING INTERFACE ONLINE

- /etc/init.d/networking restart
  - Restarting networking can cause all adapters to restart
    - Consider using nohup is connected remotely
    - sudo nohup /etc/init.d/networking restart
- ifup eth0

HOW DO WE CREATE A ROUTER IN UBUNTU?

- Easy!

- Add two network interfaces
- Configure them
- Enable Routing

CREATING UBUNTU ROUTER

- Edit /etc/network/interface and add:

```
auto eth0
iface eth0 inet static
address 10.0.0.106
netmask 255.255.255.0
gateway 10.0.0.1

auto eth1
iface eth1 inet static
address 10.0.12.1
netmask 255.255.255.0
```

- Edit /etc/sysctl.conf and uncomment line to:
  - `net.ipv4.ip_forward=1`