Networking Basics

05 - Unicast, Broadcast, Multicast, and Anycast

Wolfgang Tremmel
academy@de-cix.net

Networking Basics

DE-CIX Academy

01 - Networks, Packets, and Protocols

02 - Ethernet, 02a - VLANs, 02b - QinQ

03 - IP, 03a - Routing, 03b - Global routing

04a - UDP, 04b - TCP

04c - ICMP, 04d - Traceroute

➡ **05 - Unicast, Broadast, Multicast, and Anycast**

06a - Domain Name System (DNS)

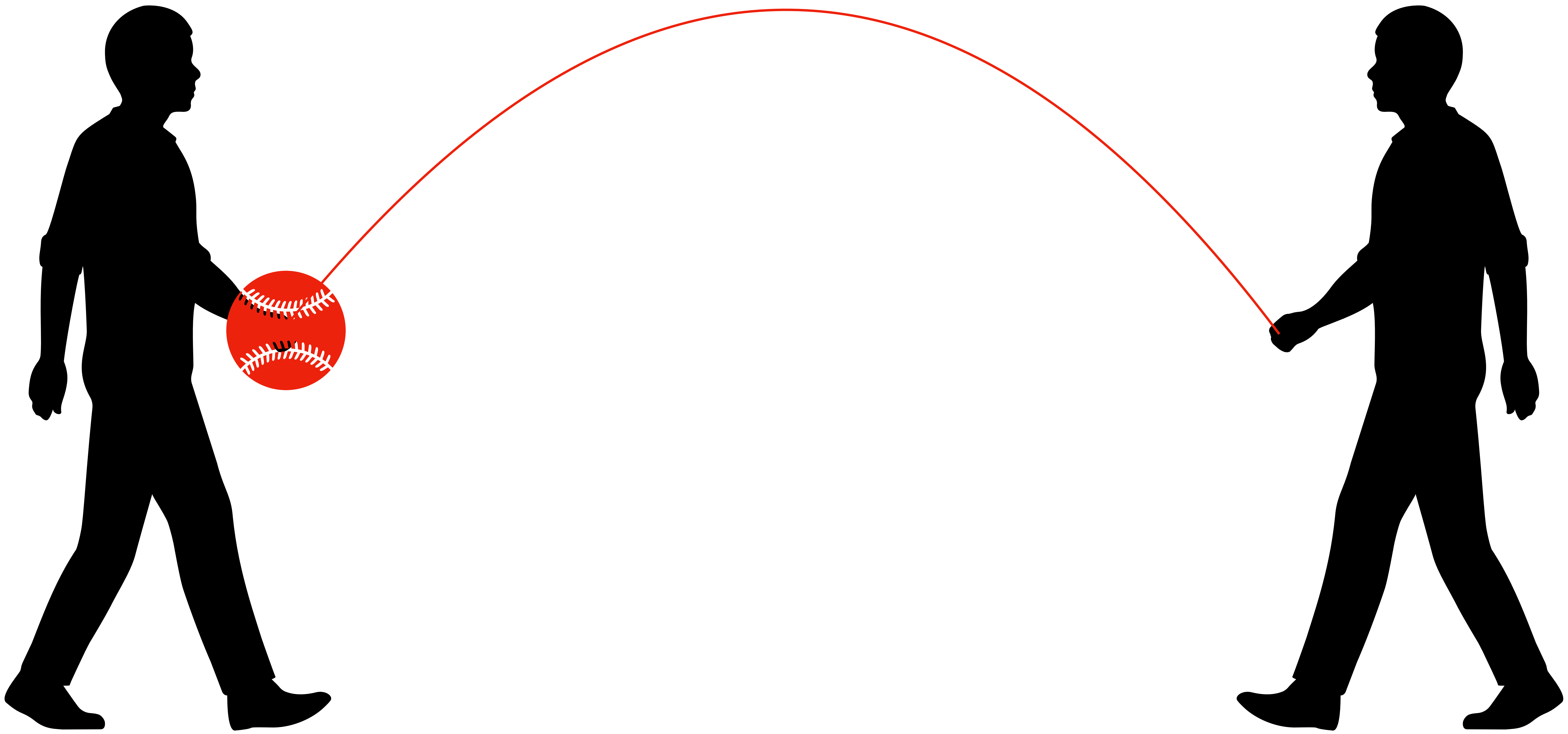
07a - SMTP, 07b - HTTP

Types of communication

DE CIX

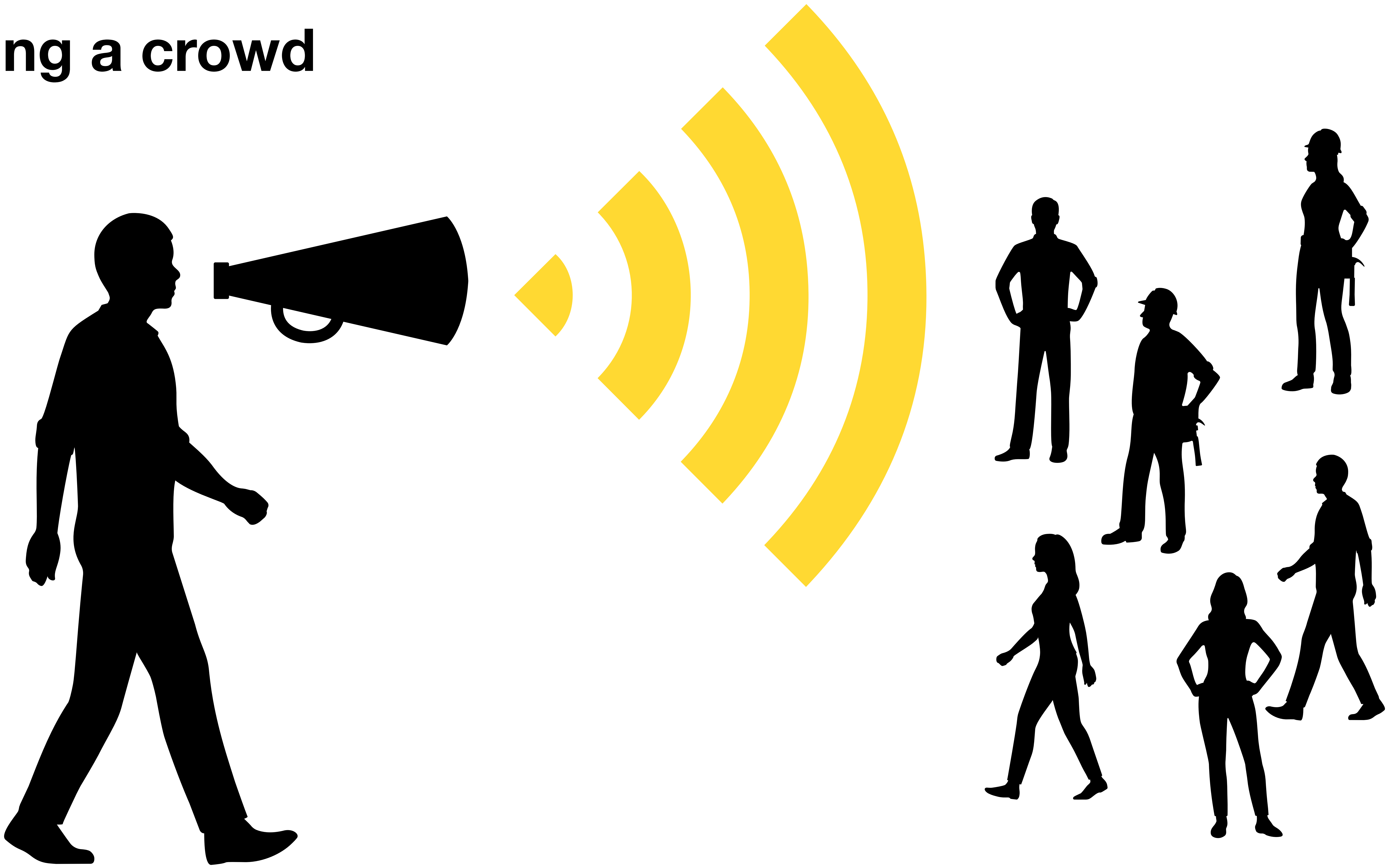
Unicast

One to one communication



Broadcast

Addressing a crowd

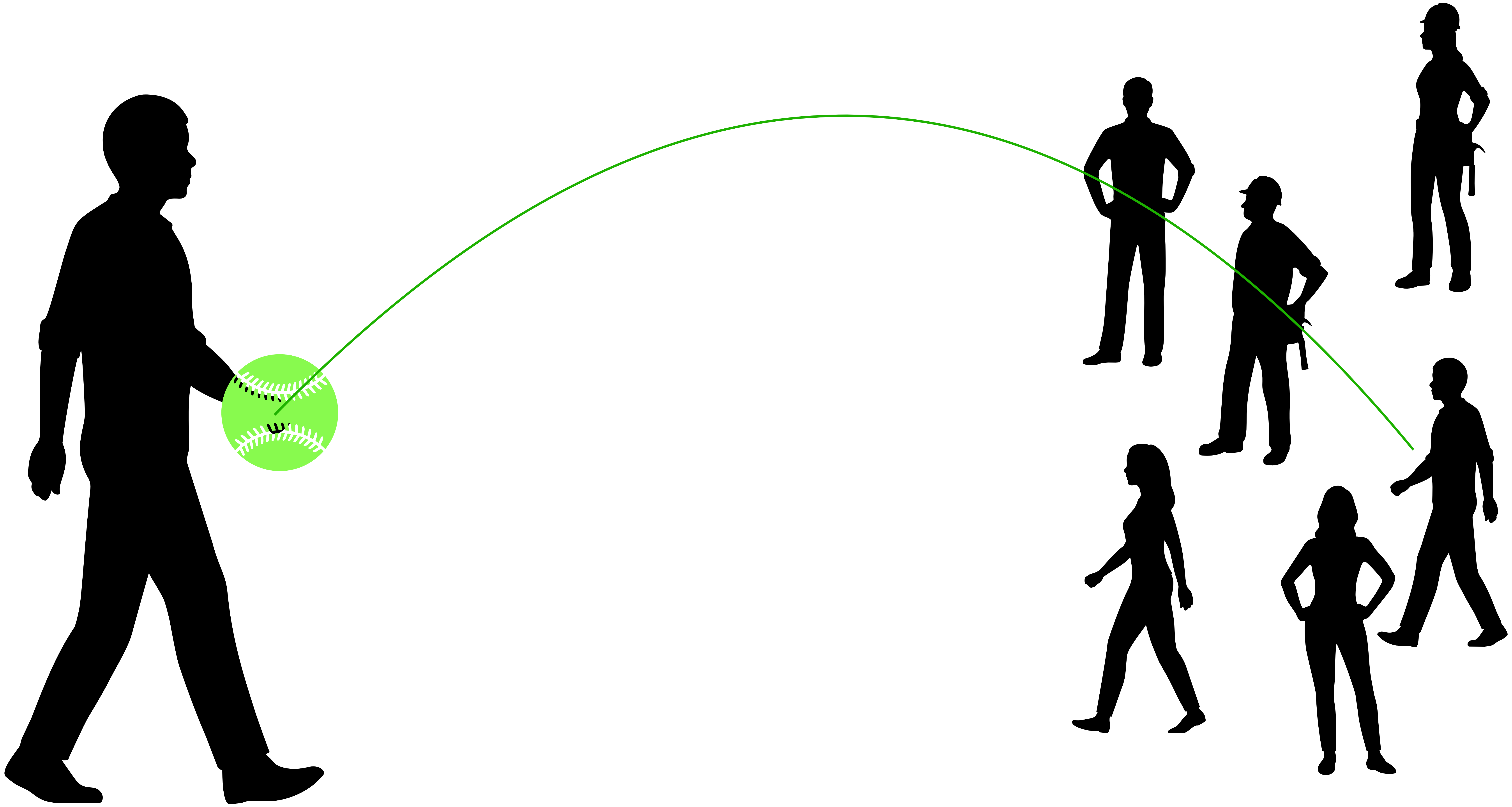


Multicast

Addressing a group

Anycast

Addressing someone - don't care who



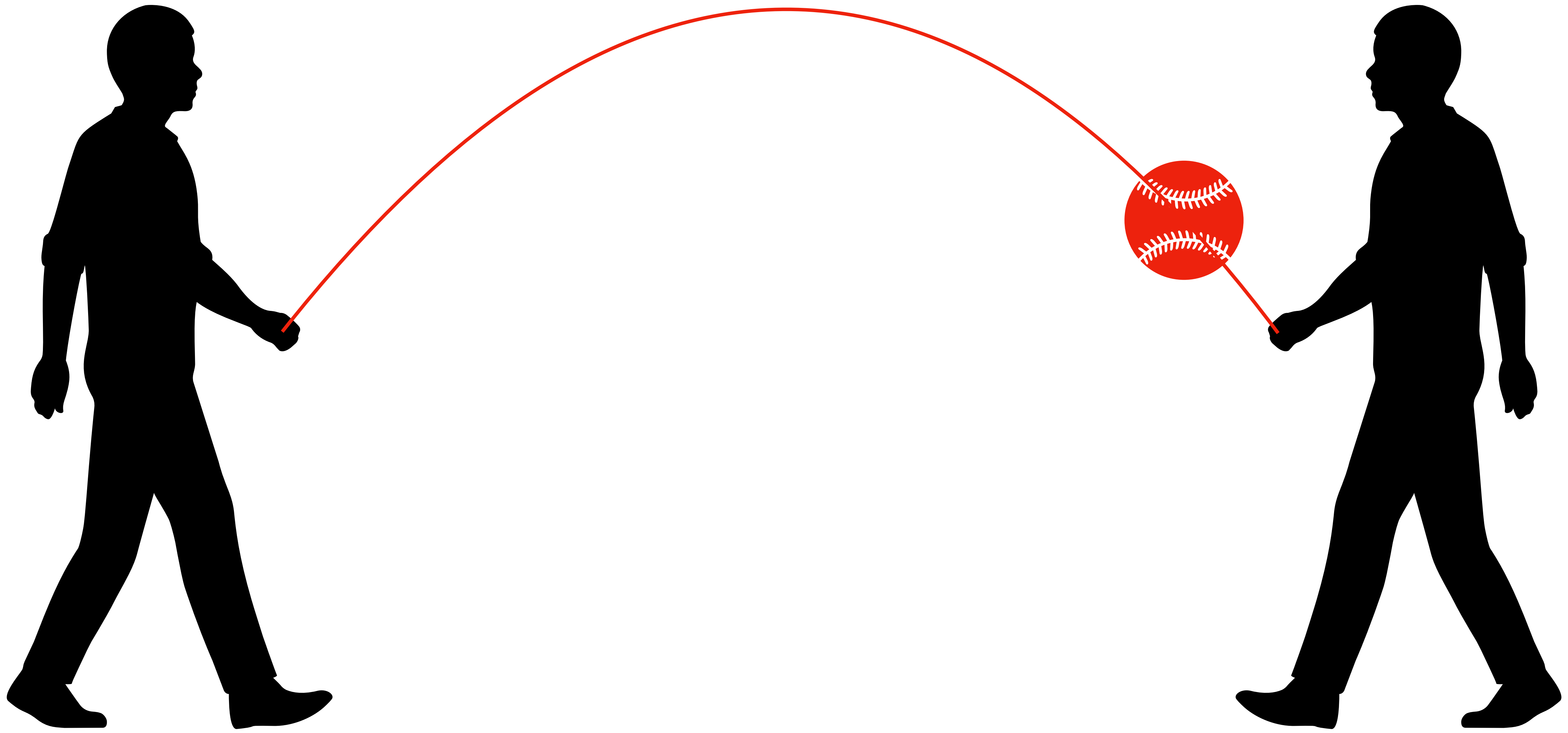
Internet Model

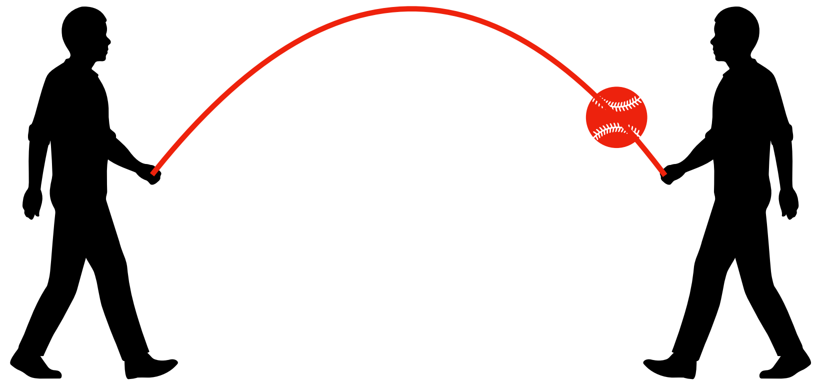
Layers working together

- This happens on (nearly) all layers
- Today we will look on the following layers
  - Link layer - Ethernet
  - Internet layer - IPv4 and IPv6
  - Transport layer
    - UDP
    - TCP

Layer	Name
5	Application
4	Transport
3	Internet
2	Link
1	Physical

Unicast





Unicast

IP Layer / Link Layer

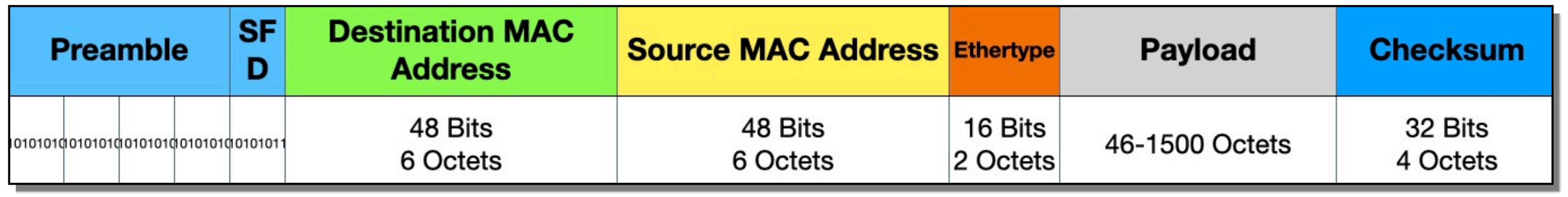
- One sender, one receiver

192.0.2.14 → 198.51.100.23

- Same for IPv4 and IPv6:
  - Destination IP address is receiver
  - Source IP address is sender

2001:db8:272b:6082:b431:3297:d13d:9c4d →
2001:db8:7978:c:226:b0ff:fed8:3d8a

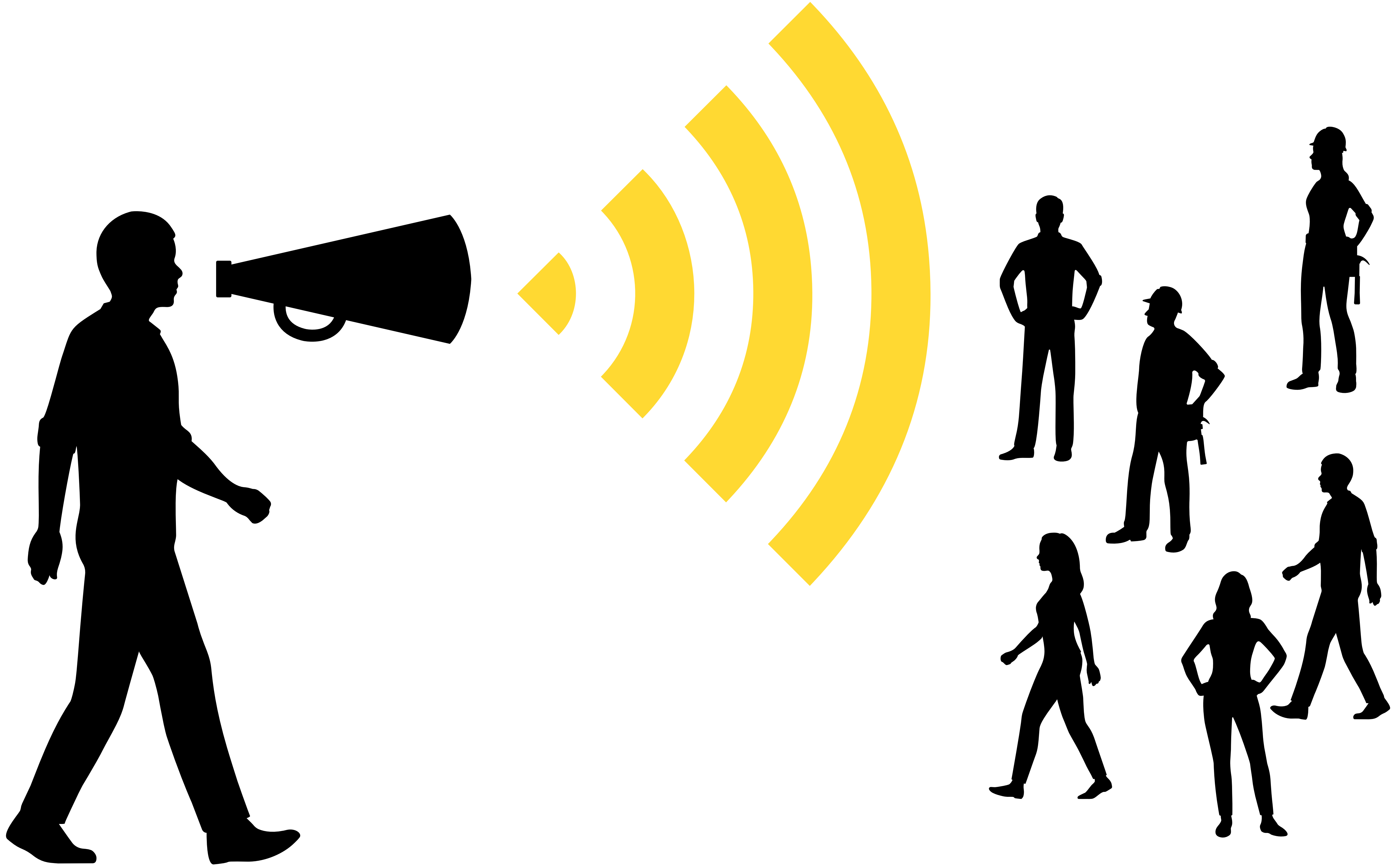
- Ethernet
  - Destination MAC address is device address of receiver
  - Source MAC address is device address of sender



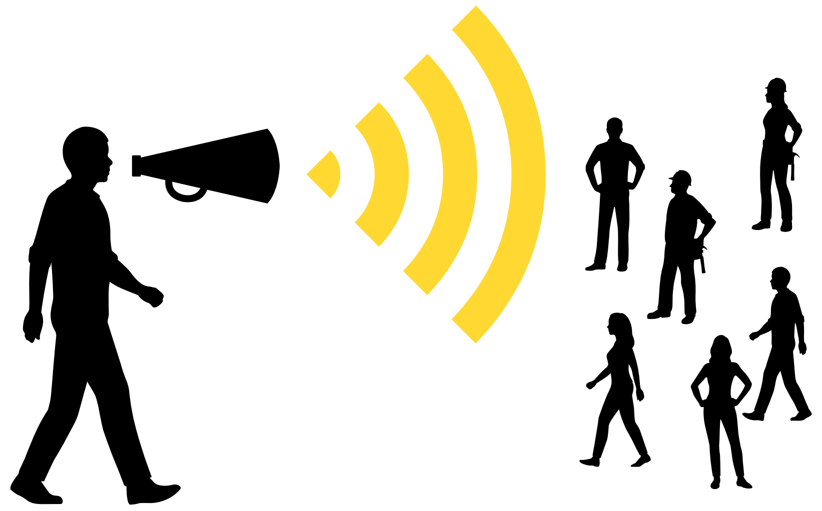
Preamble	SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010 10101010 10101010 10101010	10101011	48 Bits 6 Octets	48 Bits 6 Octets	16 Bits 2 Octets	46-1500 Octets	32 Bits 4 Octets

00:26:b0:d8:3d:8a → e0:63:da:51:ee:c8

Broadcast



DE CIX



Broadcast

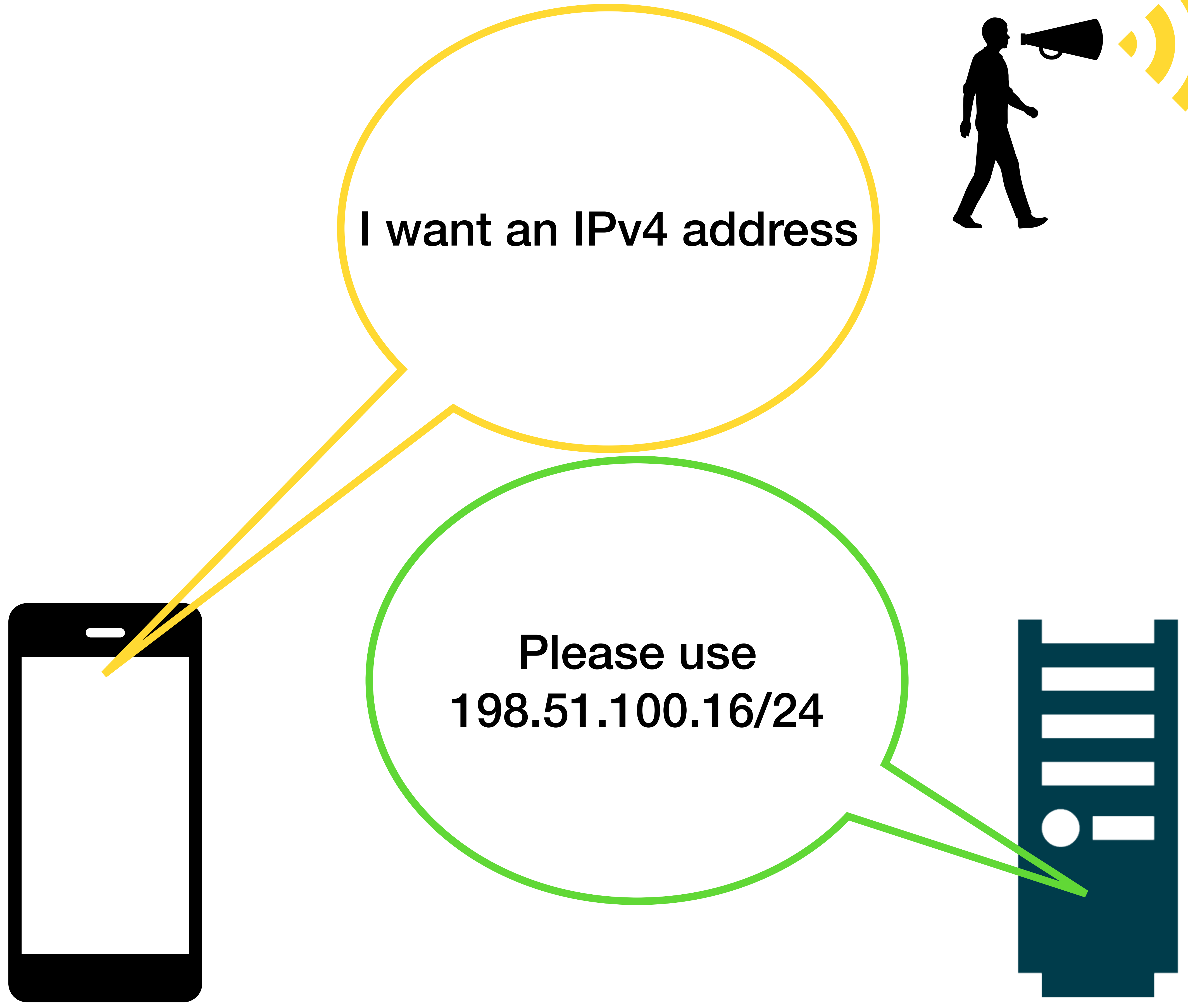
Applications of Broadcast

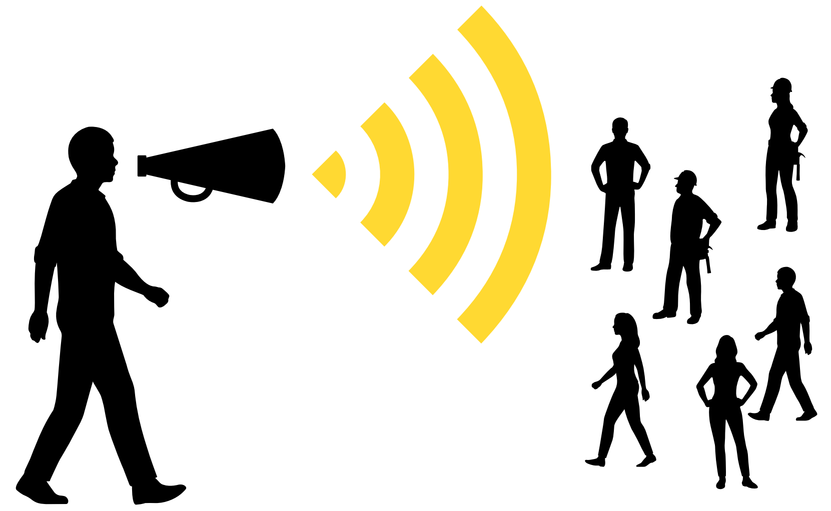
- Broadcast is often used for discovery
  - of services
  - of systems providing these services
- Example: DHCP
  - "give me an IPv4 address"

Broadcast

Example: DHCP

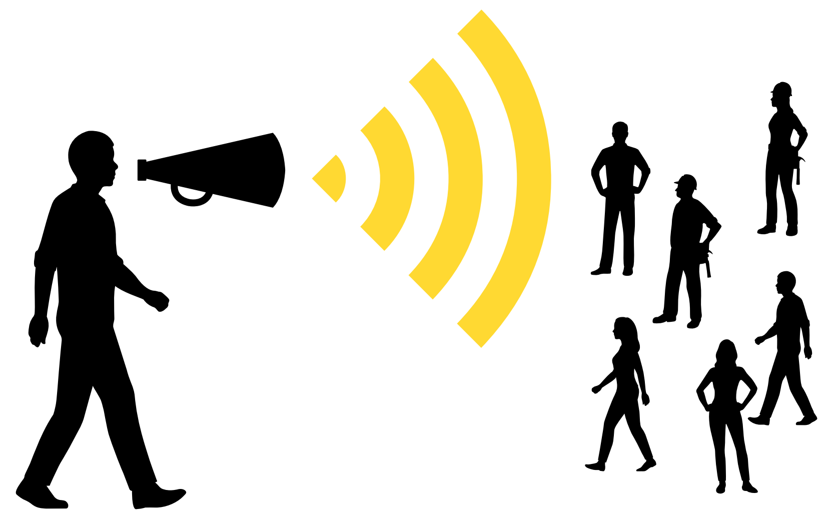
- DHCP - Dynamic Host Configuration Protocol
- Used in IPv4 to get an IPv4 address when joining a network
- Based on UDP
- Today: Just the Broadcast part





Broadcast

Example: DHCP



- Remember the IPv4 header?
  - And the UDP header?
- For this example we focus on:
  - Source- / Destination IPv4
  - Source- / Destination Port
- We skip the DHCP part of the packet

	Byte	0	1	2	3
IPv4	0	Version | Header Length always 4 | 5..15	DSCP / ECN	Total Length 20..65535	
	4	Identification		Flags / Fragment Offset	
	8	Time To Live	Protocol	Header Checksum	
	12	Source IPv4 Address			
	16	Destination IPv4 Address			

Broadcast

Example: DHCP

- Source port for DHCP is 68
- Destination port is 67
- We do not have an IPv4 address yet
  - So we use "0.0.0.0"
- We want to send a broadcast
  - So as destination IPv4 we set the ***Broadcast Address***
  - This is **255.255.255.255**

	Byte	0	1	2	3
IPv4	0	Version | Header Length always 4 | 5..15	DSCP / ECN	Total Length 20..65535	
	4	Identification		Flags / Fragment Offset	
	8	Time To Live	Protocol	Header Checksum	
	12	Source IPv4 Address			
	16	Destination IPv4 Address			
UDP	20	Source Port		Destination Port	
	24	Length		Checksum	
DHCP		DHCP Data			

Broadcast Address

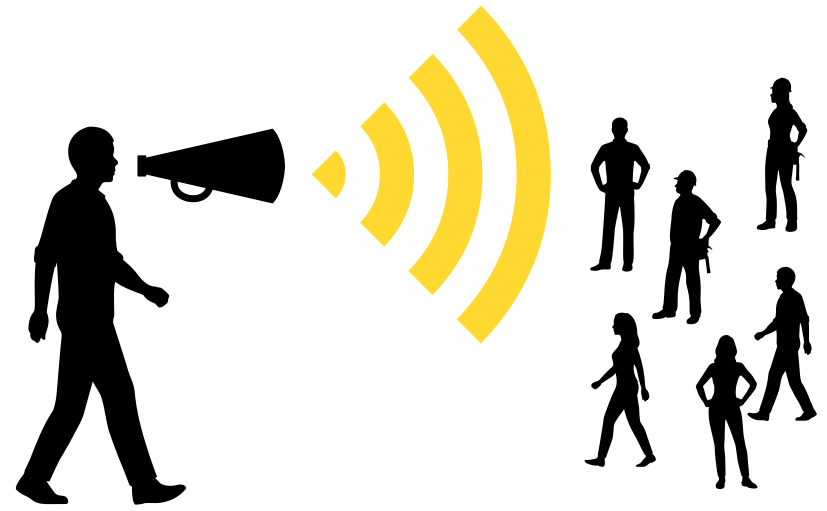
IPv4 (IPv6 does not have a concept of broadcast)

- The highest address in IPv4 is the broadcast address
- So for network 198.51.100.0/24 the broadcast address is...
    - 198.51.100.**255**
- Easy isn't it?
    - Simply set the host-part to **all-1s** (in binary) *)
- **255.255.255.255** is the broadcast address if the destination network is unknown (yet)

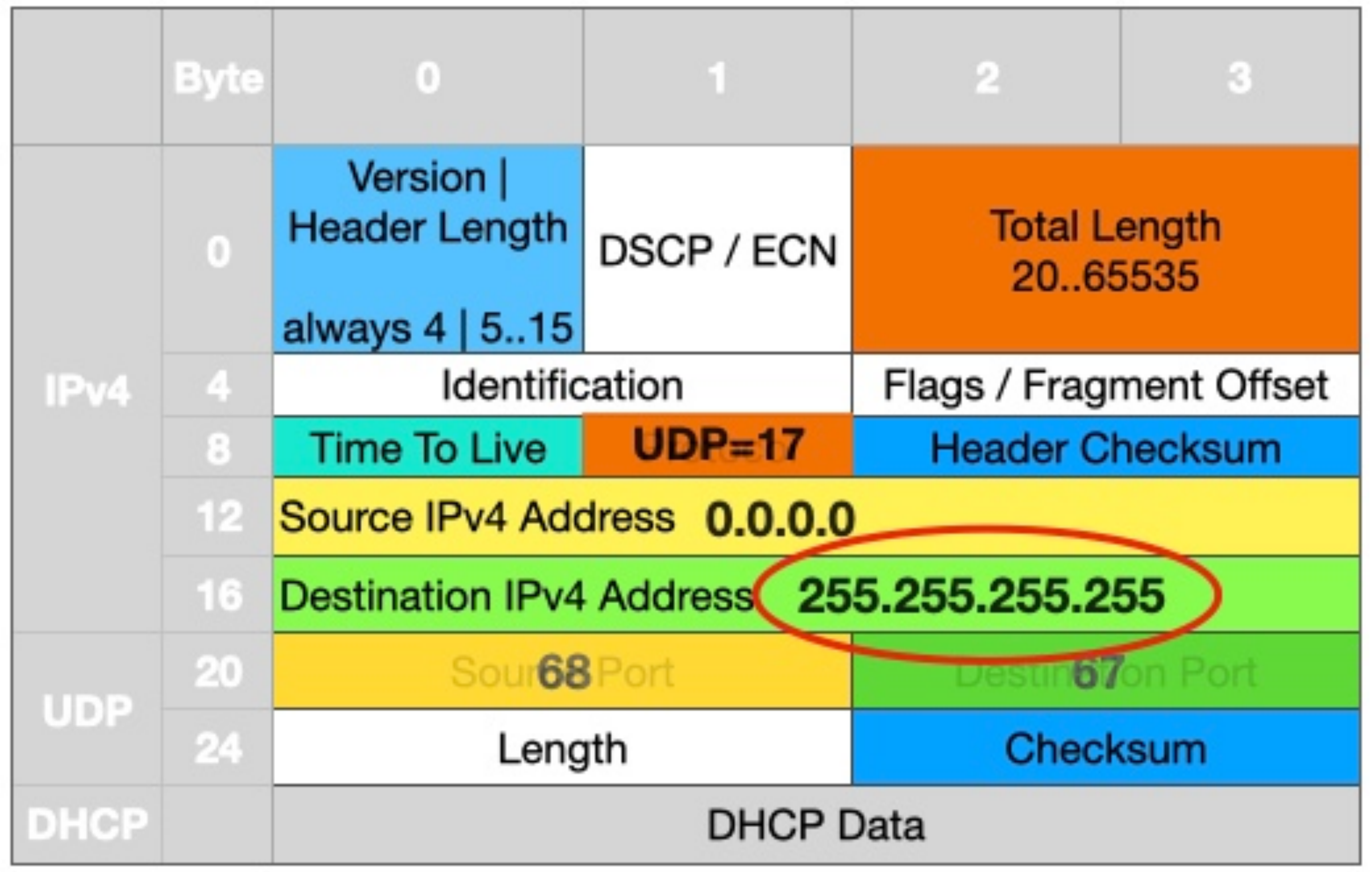
*) RFC3021 allows /31 point-to-point IPv4 networks which do not have a broadcast address

Broadcasting

Taking care of the Link Layer



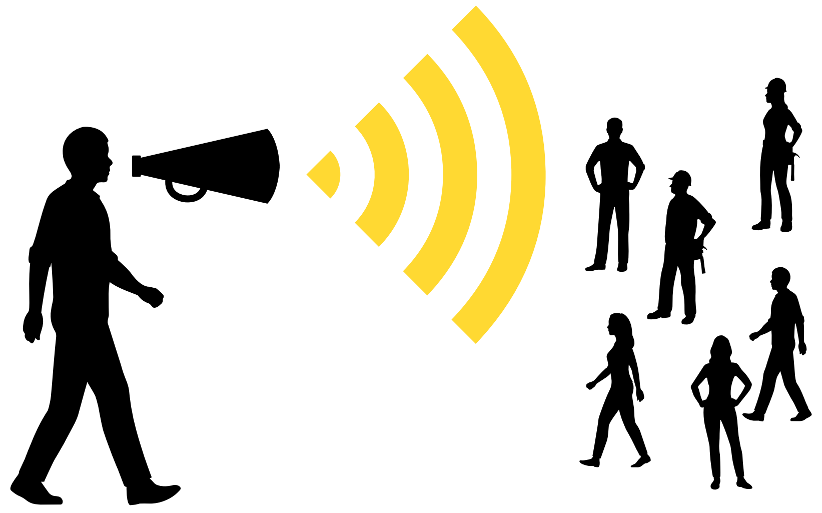
- So we now have an IPv4 UDP packet with a broadcast address as destination
- We now need to send it
- Lets put it into an Ethernet Frame



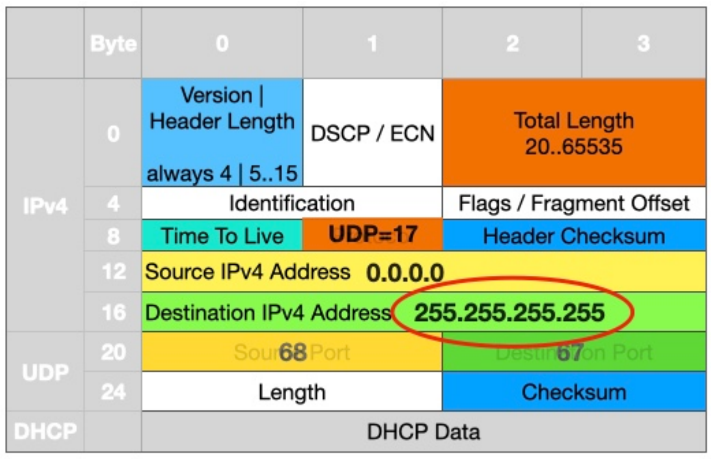
	Byte	0	1	2	3
IPv4	0	Version | Header Length always 4 | 5..15	DSCP / ECN	Total Length 20..65535	
	4	Identification		Flags / Fragment Offset	
	8	Time To Live	UDP=17	Header Checksum	
	12	Source IPv4 Address 0.0.0.0			
	16	Destination IPv4 Address 255.255.255.255			
UDP	20	Sour68Port		Destin67on Port	
	24	Length		Checksum	
DHCP		DHCP Data			

Broadcasting

Taking care of the Link Layer



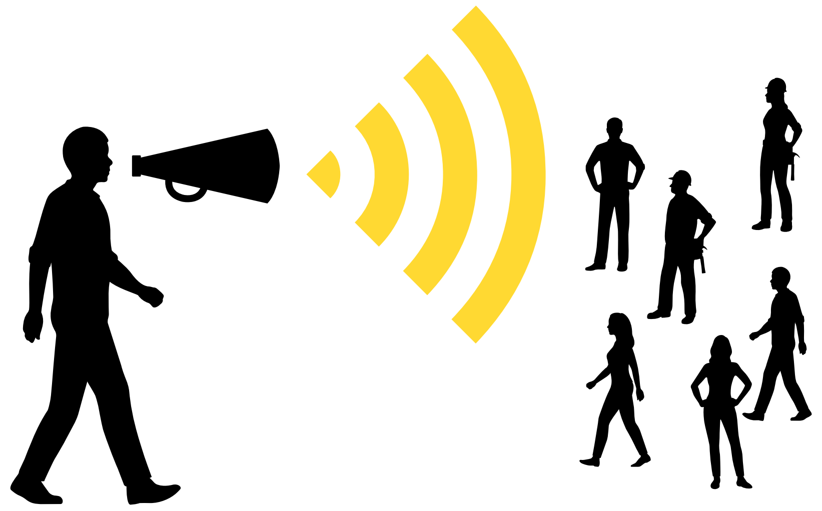
- So we now have an IPv4 UDP packet with a broadcast address as destination
- We now need to send it
- Lets put it into an Ethernet Frame
- Ethertype is 0x800 (IPv4)

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	48 Bits 6 Octets	48 Bits 6 Octets	16 Bits 2 Octets		32 Bits 4 Octets
10101010	10101010	10101010	10101010	10101011			0x800		

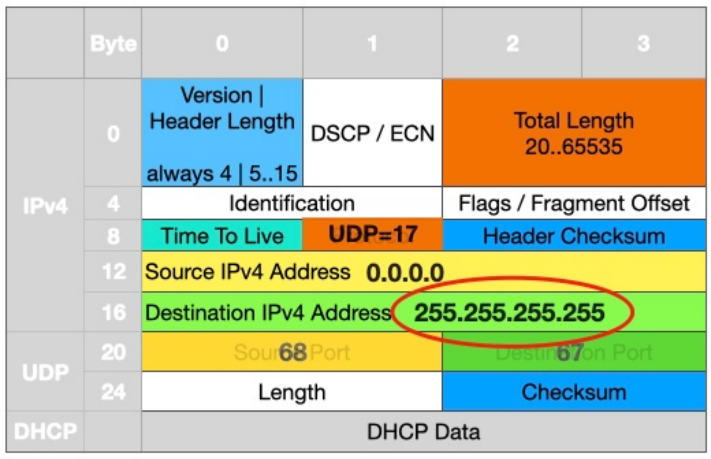
	Byte	0	1	2	3
IPv4	0	Version | Header Length always 4 | 5..15	DSCP / ECN	Total Length 20..65535	
	4	Identification		Flags / Fragment Offset	
	8	Time To Live	UDP=17	Header Checksum	
	12	Source IPv4 Address 0.0.0.0			
	16	Destination IPv4 Address 255.255.255.255			
UDP	20	Sour 68 Port		Destin 67 on Port	
	24	Length		Checksum	
DHCP		DHCP Data			

Broadcasting

Taking care of the Link Layer



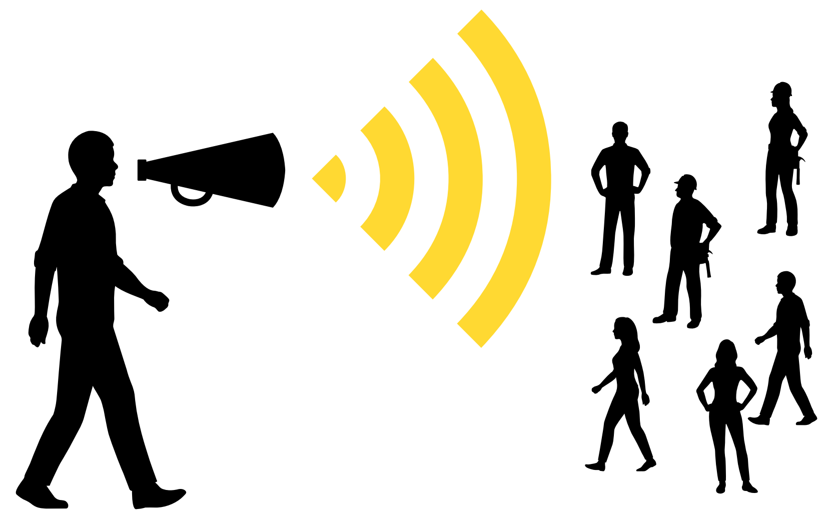
- Ethertype is 0x800 (IPv4)
- Source MAC address is our MAC address (hardware)
- Destination MAC address is the Ethernet Broadcast Address
  - **ff:ff:ff:ff:ff:ff** - all "1"s

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	48 Bits 6 Octets	48 Bits 6 Octets	16 Bits 2 Octets		32 Bits 4 Octets
10101010	10101010	10101010	10101010	10101011	**ff:ff:ff:ff:ff:ff**	00:26:b0:d8:3d:8a	0x800		

	Byte	0	1	2	3
IPv4	0	Version | Header Length always 4 | 5..15	DSCP / ECN	Total Length 20..65535	
	4	Identification		Flags / Fragment Offset	
	8	Time To Live	**UDP=17**	Header Checksum	
	12	Source IPv4 Address **0.0.0.0**			
	16	Destination IPv4 Address		**255.255.255.255**	
UDP	20	Sour **68** Port		Destin **67** on Port	
	24	Length		Checksum	
DHCP		DHCP Data			

Broadcasting

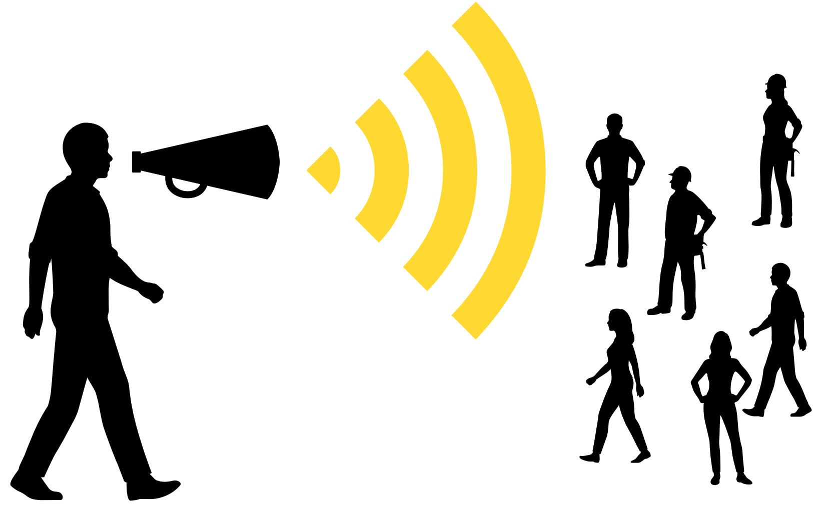
Processing and disadvantages



- Ethernet frames with Broadcast destination MACs are forwarded to the CPU / Operating System for processing on all stations
- For DHCP: If the receiving system has a UDP service on port 67
  - the DHCP request is processed and maybe answered
  - if not, the OS discards the packet (not the network hardware!)
- this consumes CPU cycles on every connected device
- at all stations on the Ethernet - this can be a problem on large LANs
- the DE-CIX Frankfurt peering LAN has more than 1500 stations connected and had more than 1.5Mbit/s broadcast traffic

Broadcasting

Conclusion



- Broadcast means "send to every device on the shared network"
- On IPv4 the broadcast address is the highest address on the network
  - with the host-part of the address all 1s
- On Ethernet the broadcast address is ff:ff:ff:ff:ff:ff
- IPv6 has no concept of broadcast
- Broadcast should be avoided when ever possible as it consumes unnecessary resources

Multicast (in a LAN)

Multicast in a LAN

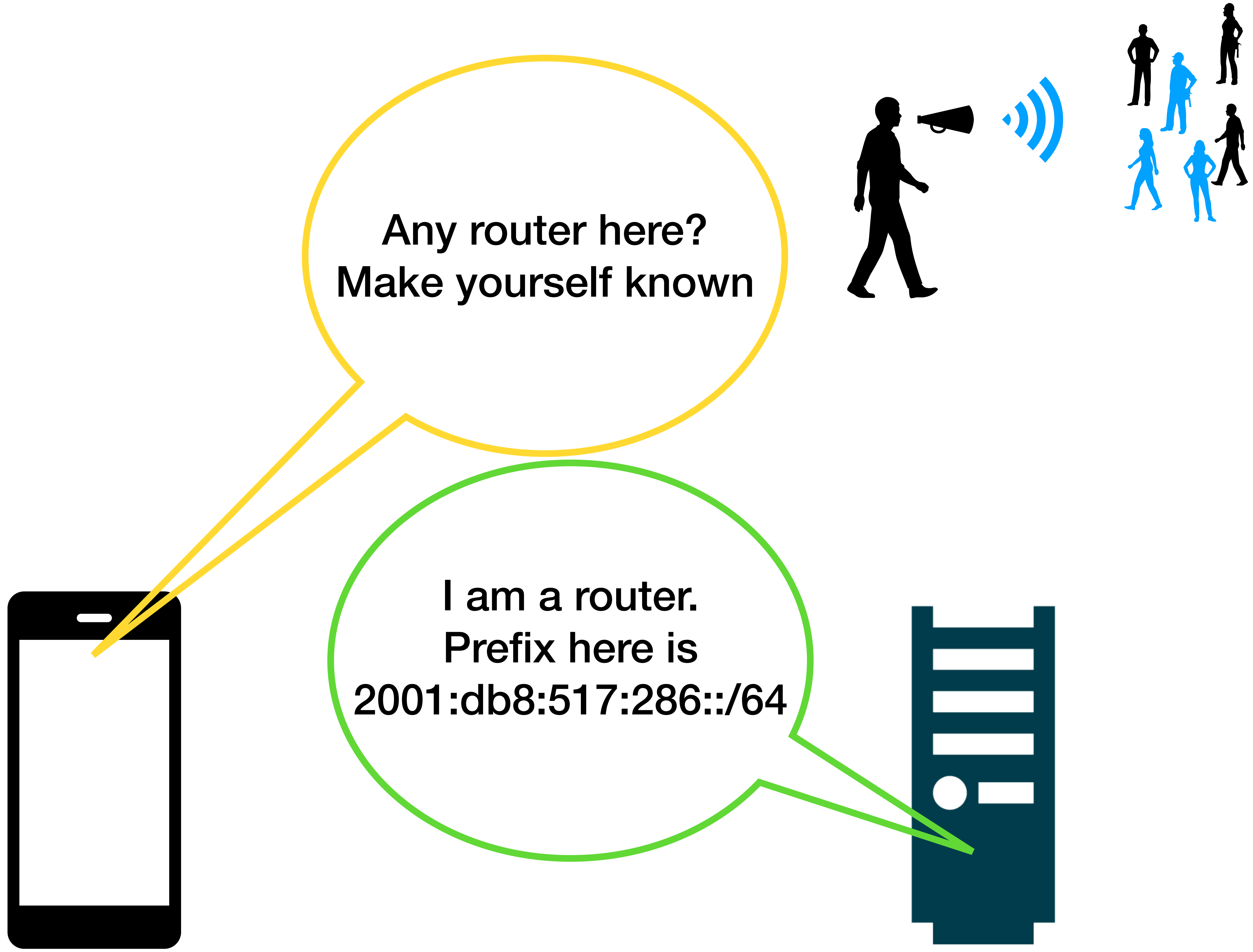
IPv6 has no concept of Broadcast

- When specifying IPv6 it was decided Broadcast was not really needed
  - As broadcast can be seen as just a special case of Multicast
- It is more effective to address a "group of hosts" than "all hosts"
  - a group like "all routers"
    - or even more specialised "all routers running RIP"
  - or all "network time protocol servers"
- hosts join these groups when they run the specified protocol

Multicast in IPv6

Example: Neighbor Discovery

- Neighbor discovery is used to discover stations on the local network
- Like asking for any router
- We cover just the question part via multicast here
- Address assignment is a topic for another time



Multicast in IPv6

Example: Neighbor Discovery

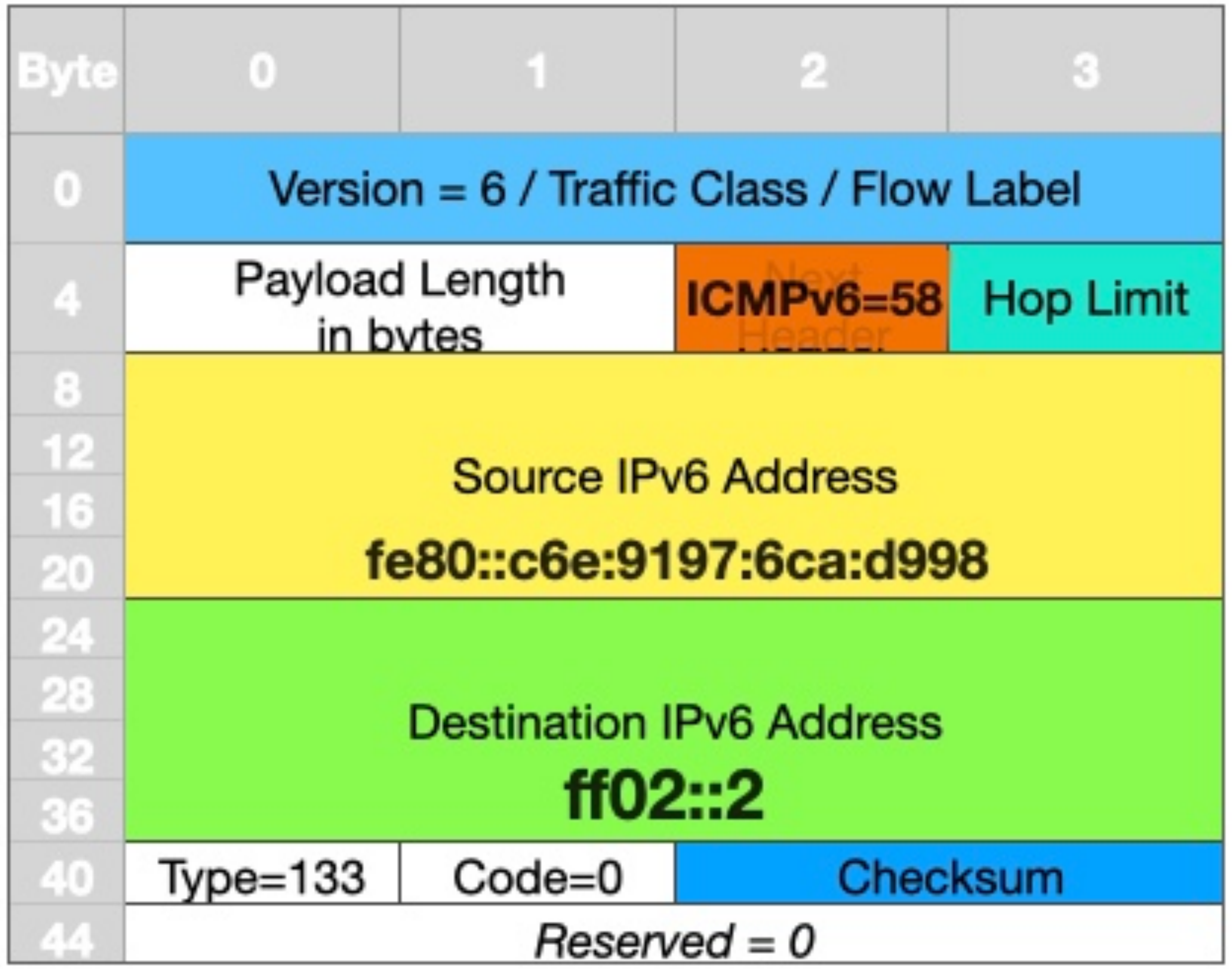
- Neighbor Discovery uses ICMP, but we focus on the IPv6 multicast part
- Destination is "all routers" address: **ff02::2**
- Source address is the "link local" address of the sending interface

Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length in bytes		Next Header	Hop Limit
8	Source IPv6 Address			
12				
16				
20				
24	Destination IPv6 Address			
28				
32				
36				

Multicast in IPv6

Transport via Ethernet

- We now have an IPv6 packet with a Mulitcast destination address
- Again we must put this into an Ethernet Frame
- RFC2464 tells us how

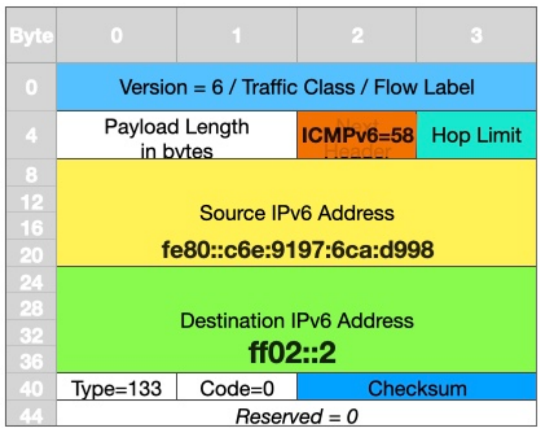


Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length in bytes		ICMPv6=58 Next Header	Hop Limit
8 12 16 20	Source IPv6 Address **fe80::c6e:9197:6ca:d998**			
24 28 32 36	Destination IPv6 Address **ff02::2**			
40	Type=133	Code=0	Checksum	
44	*Reserved = 0*			

Multicast

Taking care of the Link Layer

- We now have an IPv6 packet with a Mulitcast destination address
- Again we must put this into an Ethernet Frame
- RFC2464 tells us how
- Ethertype is 0x86dd (IPv6)

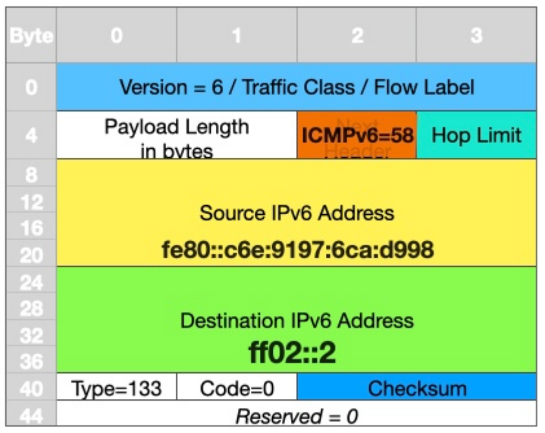
Preamble				SF D	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	48 Bits 6 Octets	48 Bits 6 Octets	16 Bits 2 Octets		32 Bits 4 Octets
10101010	10101010	10101010	10101010	10101011			0x86dd		

Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length in bytes		ICMPv6=58	Hop Limit
8 12 16 20	Source IPv6 Address fe80::c6e:9197:6ca:d998			
24 28 32 36	Destination IPv6 Address ff02::2			
40	Type=133	Code=0	Checksum	
44	*Reserved = 0*			

Multicast

Taking care of the Link Layer

- Ethertype is 0x86dd (IPv6)
- Source MAC address is the senders hardware MAC address
- Destination is built from the destination IPv6 multicast address:
  - 33:33 as first two octets
  - Last four octets of destination IPv6 as last four octets of destination MAC

Preamble				SFD	Destination MAC Address	Source MAC Address	Ethertype	Payload	Checksum
10101010	10101010	10101010	10101010	10101011	48 Bits 6 Octets	48 Bits 6 Octets	16 Bits 2 Octets		32 Bits 4 Octets
10101010	10101010	10101010	10101010	10101011	**33:33:00:00:00:02**	00:26:b0:d8:3d:8a	0x86dd		

Byte	0	1	2	3
0	Version = 6 / Traffic Class / Flow Label			
4	Payload Length in bytes		ICMPv6=58	Hop Limit
8	Source IPv6 Address **fe80::c6e:9197:6ca:d998**			
12				
16				
20				
24	Destination IPv6 Address **ff02::2**			
28				
32				
36				
40	Type=133	Code=0	Checksum	
44	*Reserved = 0*			

Multicast

Want to know which multicast groups your host has joined?

On Linux:

```
ip maddr show
```

On Mac OS:

```
netstat -g
```

On Windows:

```
netsh interface ip show joins
netsh interface ipv6 show joins
```

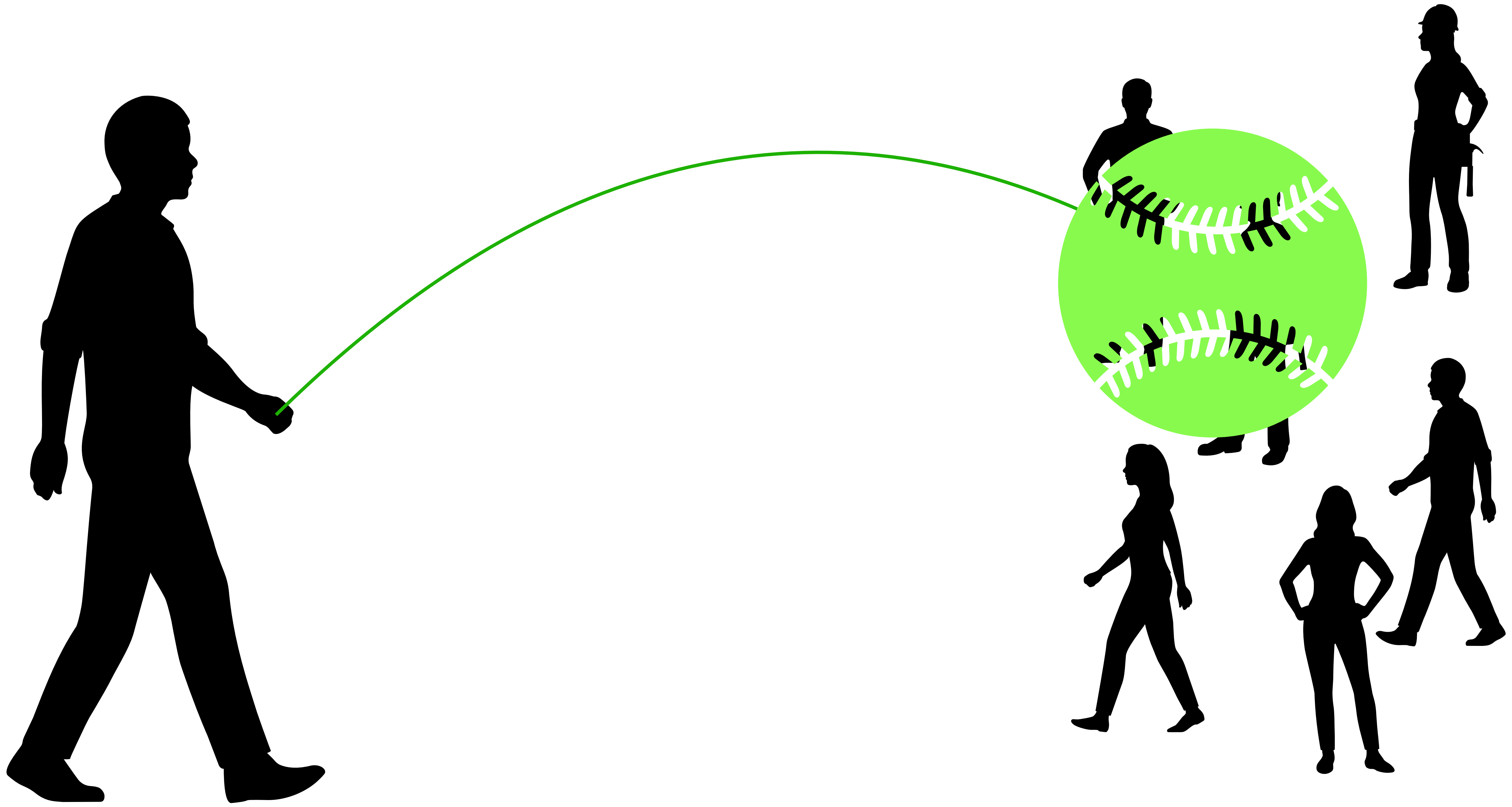
1010101

1010101

um

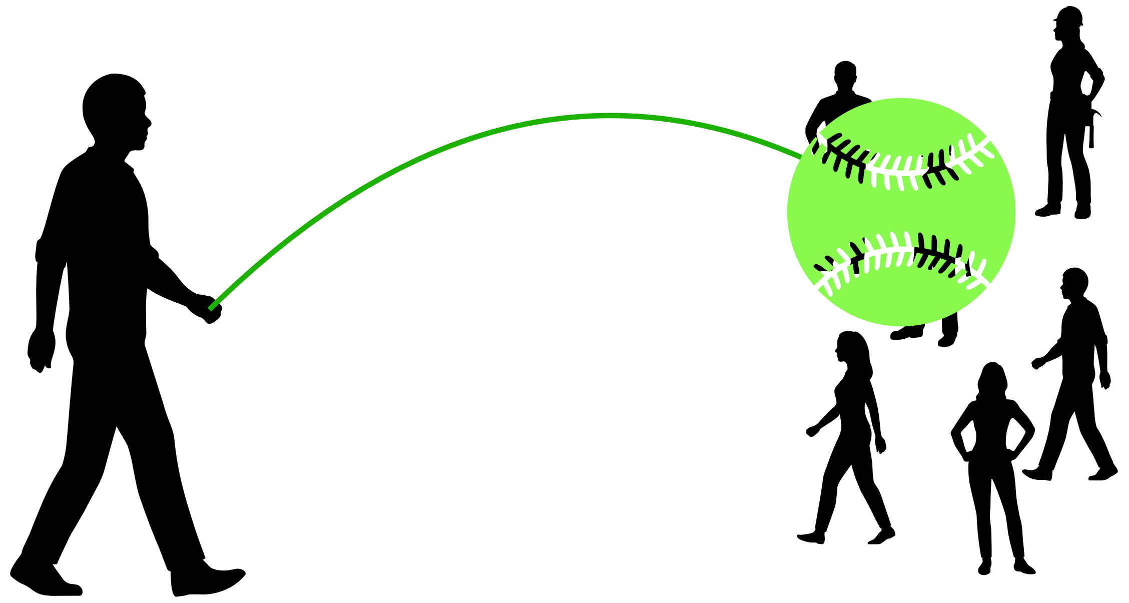
s

Anycast



Anycast

Ask one in a group

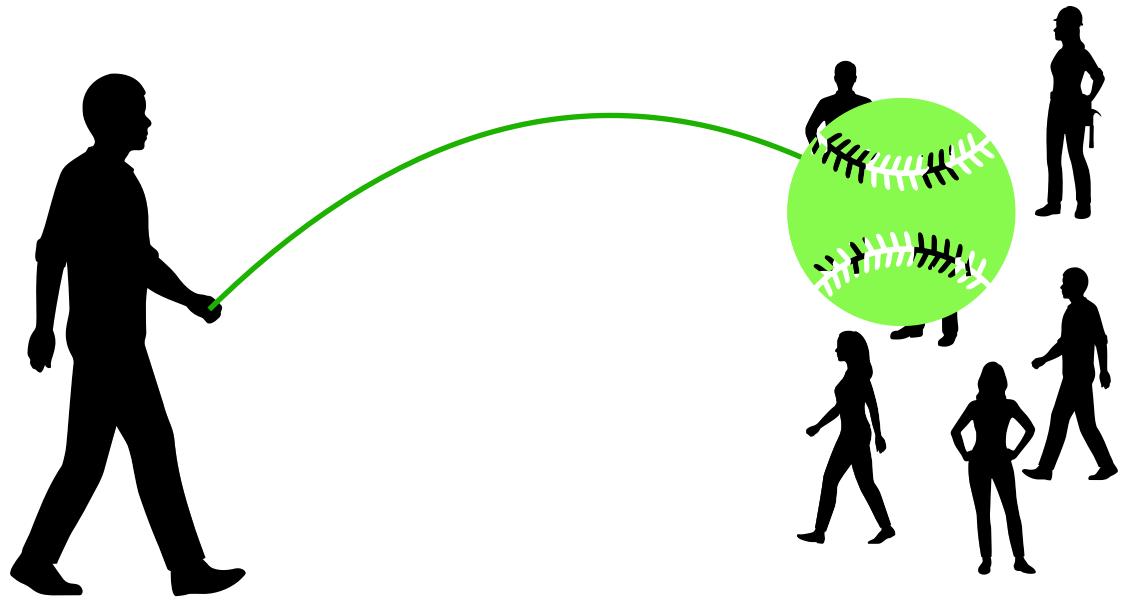


- For the sender the same as Unicast, but you address a group
- And you do not care who answers, because everyone in that group will give you the same answer
- Ideas how to use this?
  - Name to IP address resolution - DNS!
  - Send one packet, get one packet back
  - Via UDP - it is stateless (no "follow up conversation")

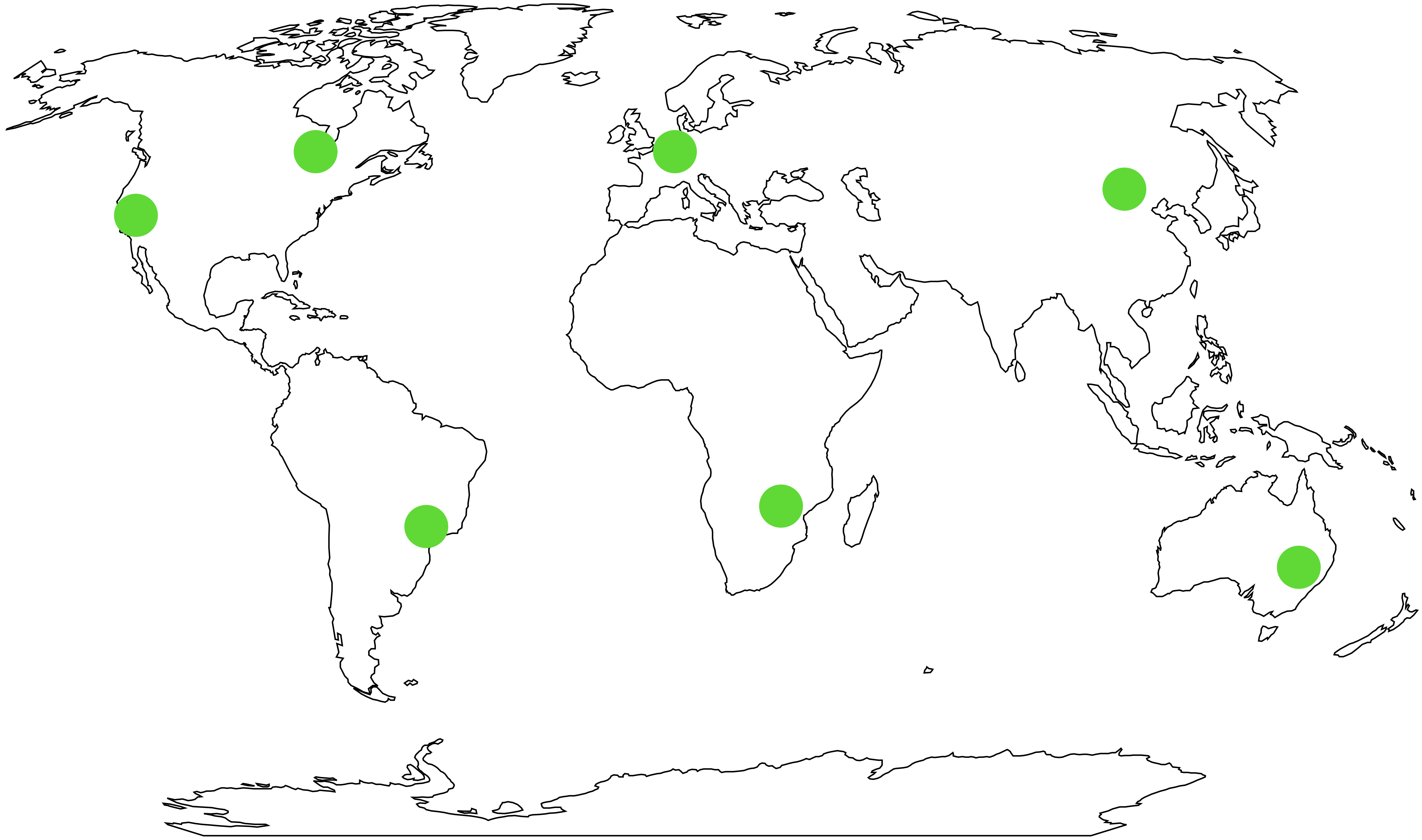
DE CIX

Anycast

How does it work?



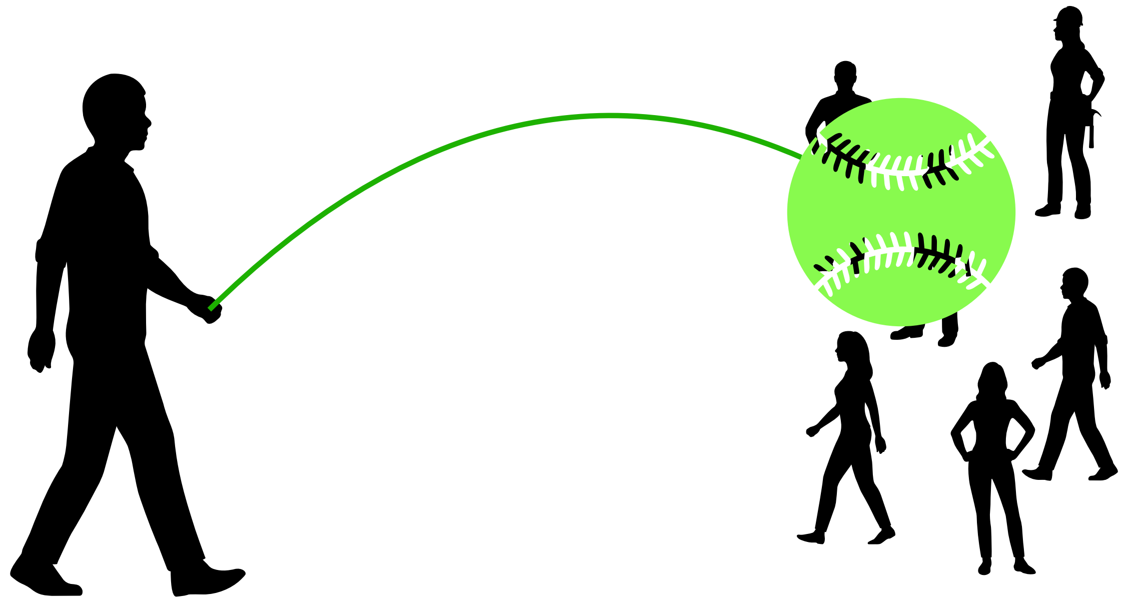
- Use the same IP address at many places (even more as shown)
- With the same Autonomous System number (don't worry if you do not know what that is)
- Routing makes sure the nearest system is reached
- Which answers the query



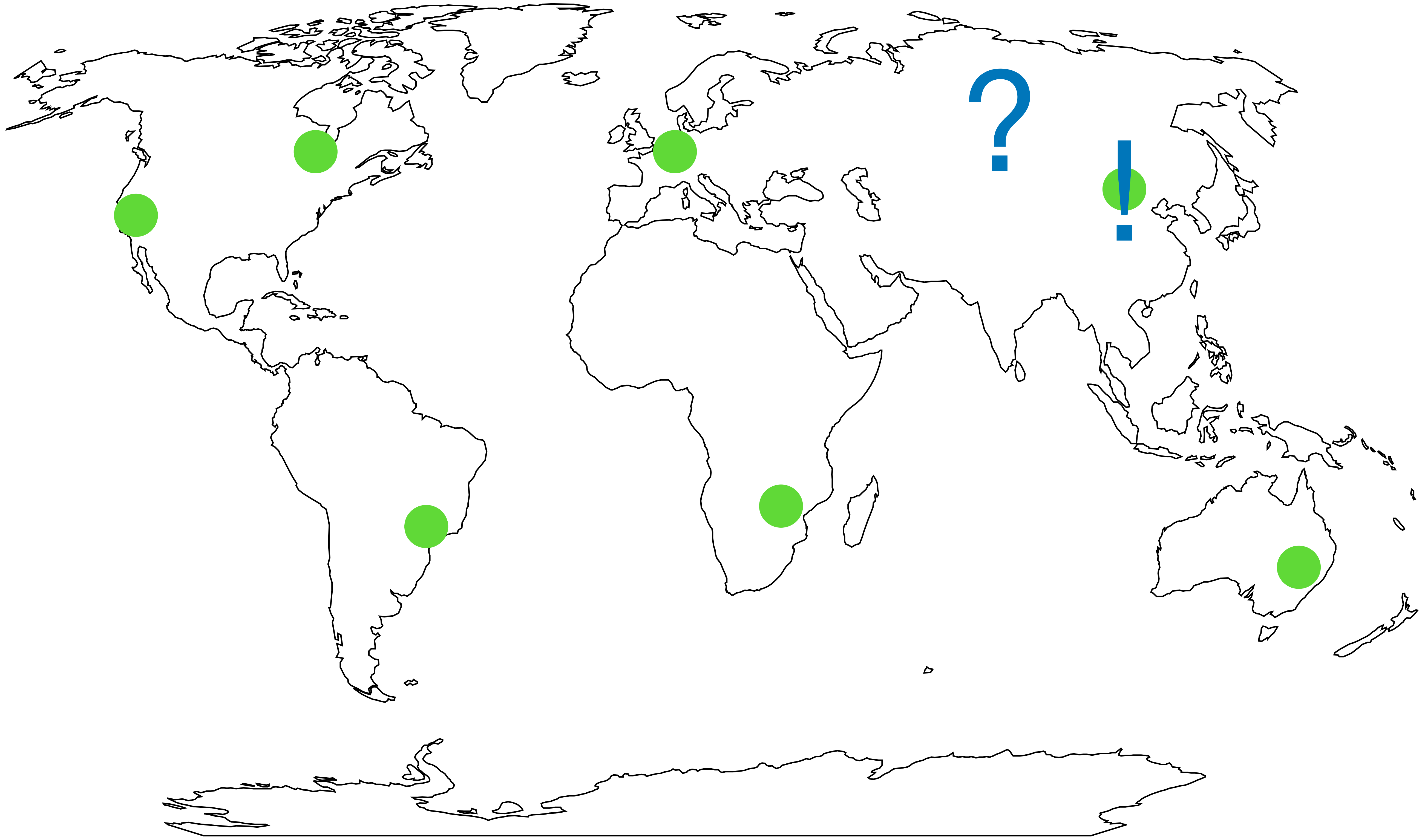
DE CIX

Anycast

How does it work?

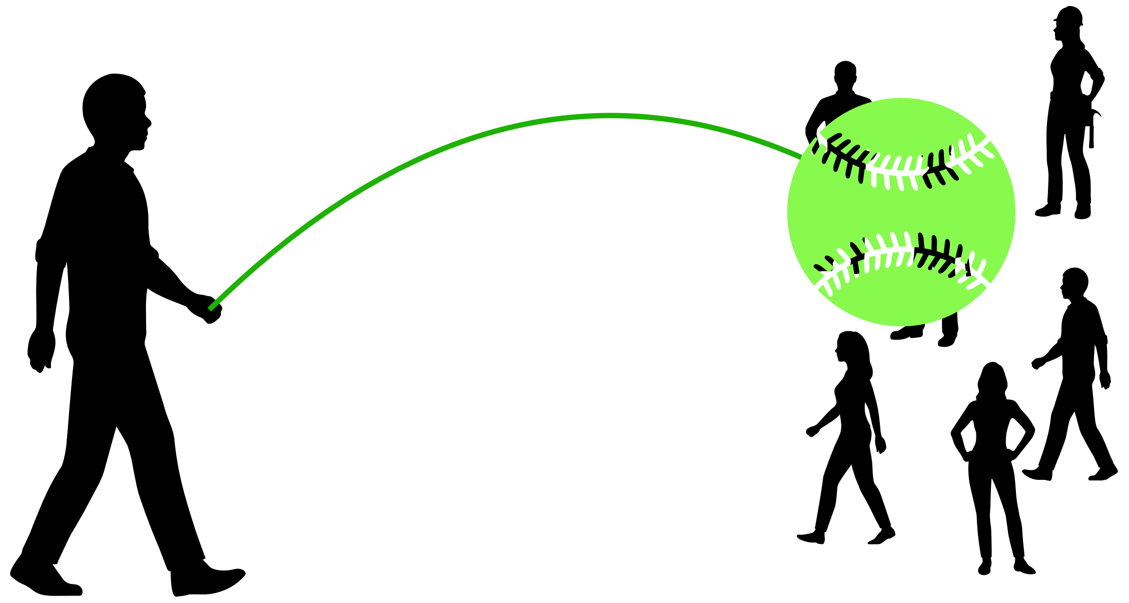


- Use the same IP address at many places (even more as shown)
- With the same Autonomous System number (don't worry if you do not know what that is)
- Routing makes sure the nearest system is reached
- Which answers the query

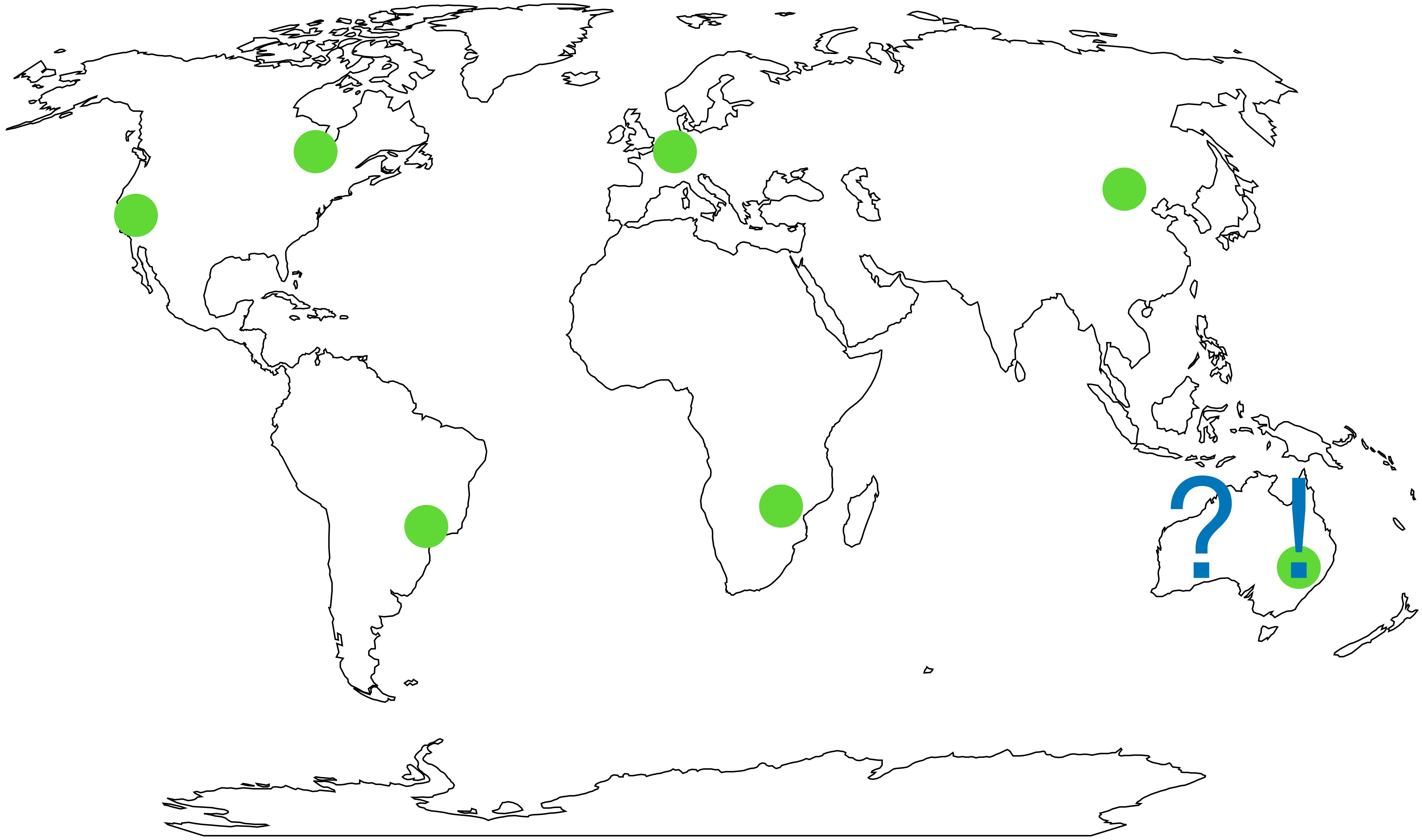


Anycast

How does it work?

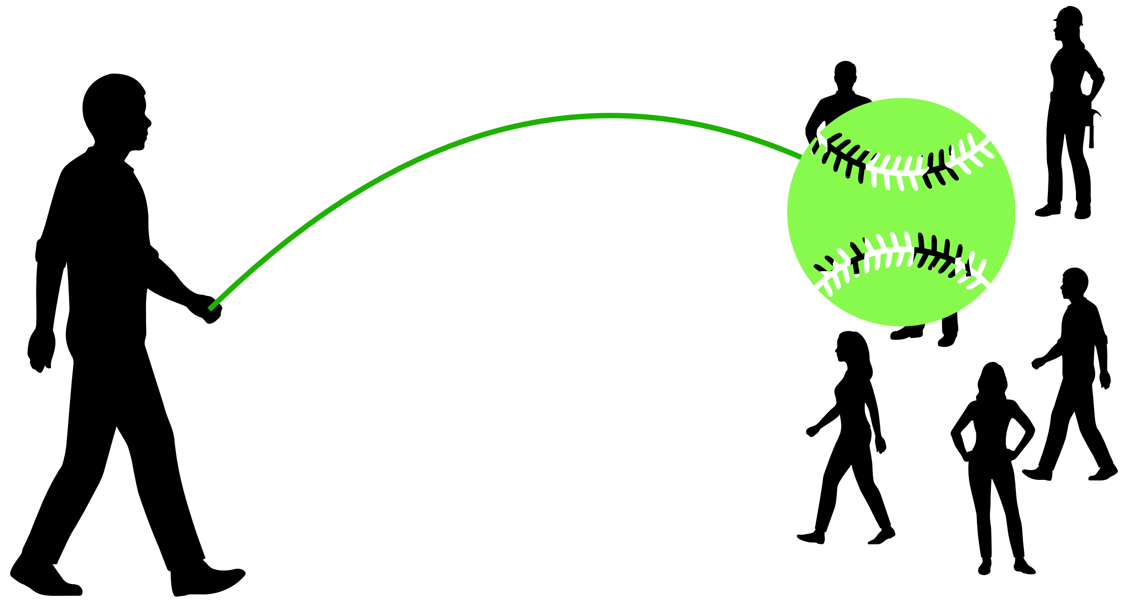


- Use the same IP address at many places (even more as shown)
- With the same Autonomous System number (don't worry if you do not know what that is)
- Routing makes sure the nearest system is reached
- Which answers the query



Anycast

And TCP?



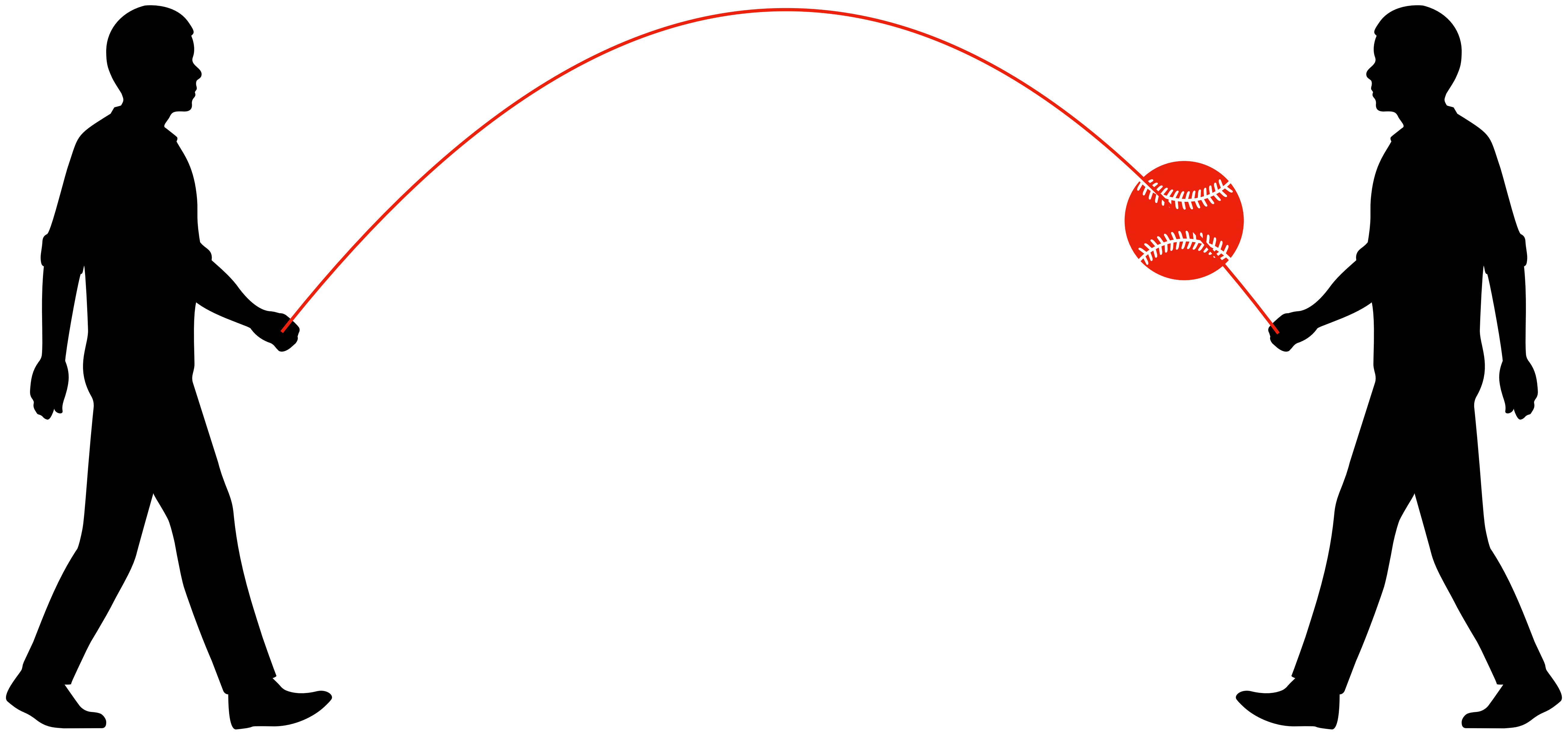
- Using Anycast with UDP is easy.
- But it also works with TCP
    - "Nearest" server takes the connection
    - If during the connection another server is "nearer" and takes over
        - TCP session is reset and needs to be re-established
        - The speed gain due to lower latency outweighs the occasional hickup

Conclusion

Conclusion

Modes of communication

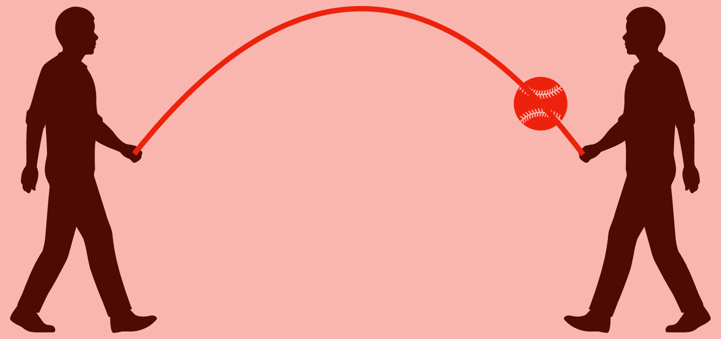
- Unicast
  - 1:1 communication
  - Standard in the Internet



Conclusion

Modes of communication

- Unicast
  - 1:1 communication
  - Standard in the Internet

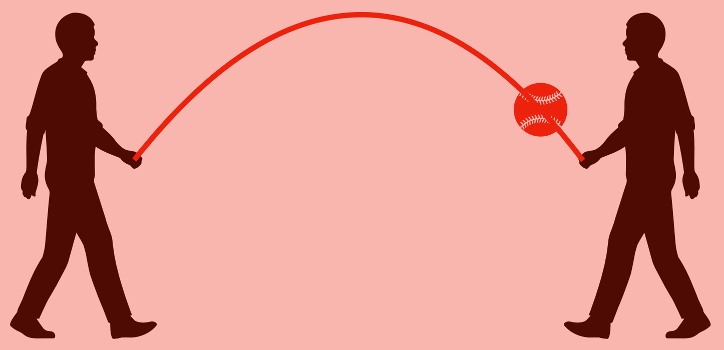


- Broadcast
  - 1:all
  - Discovery

Conclusion

Modes of communication

- Unicast
  - 1:1 communication
  - Standard in the Internet



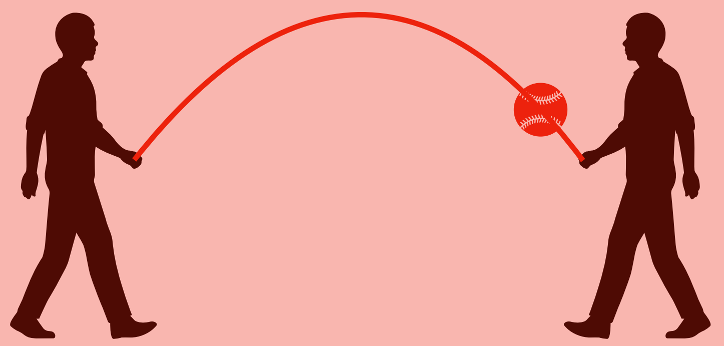
- Broadcast
  - 1:all
  - Discovery

- Multicast
  - 1:group
  - address a group with something in common

Conclusion

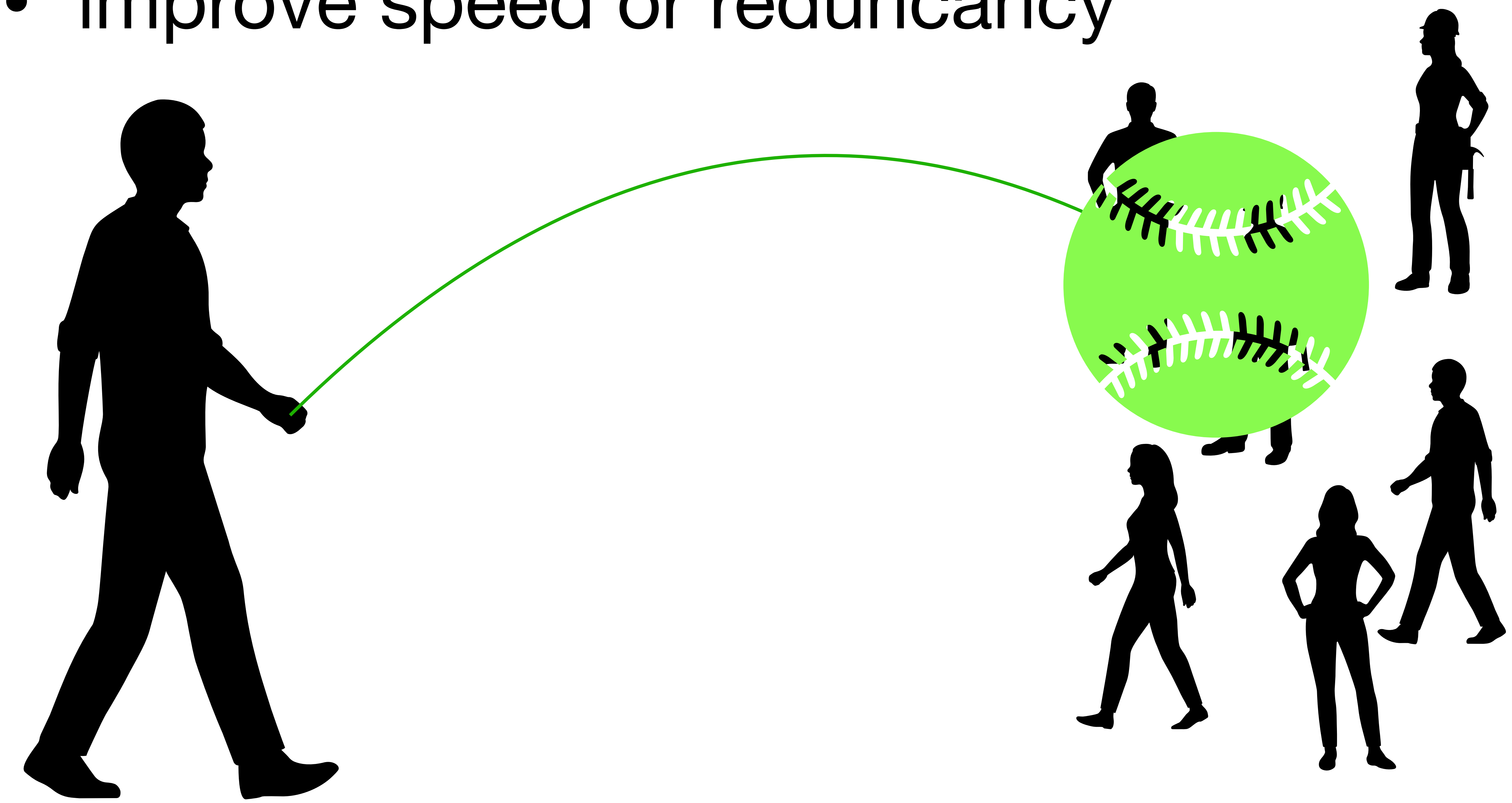
Modes of communication

- Unicast
  - 1:1 communication
  - Standard in the Internet



- Broadcast
  - 1:all
  - Discovery

- Anycast
  - 1:1 but don't care which one
  - improve speed or reduncancy

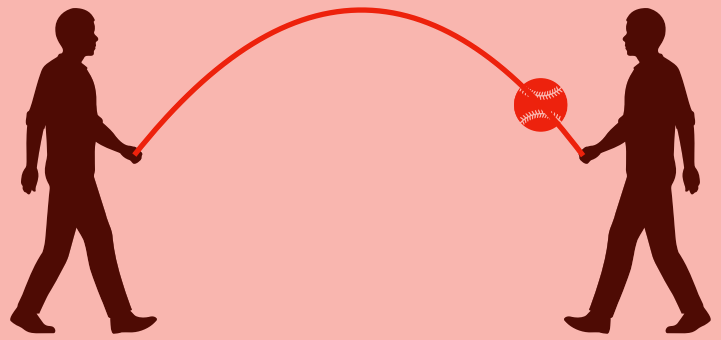


- Multicast
  - 1:group
  - address a group with something in common

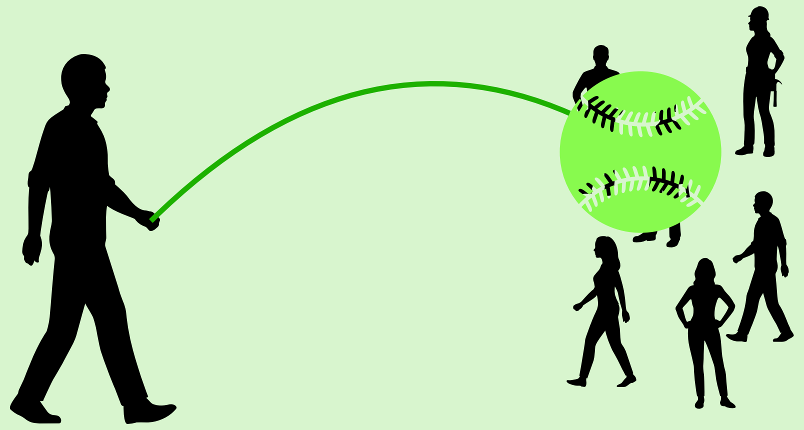
Conclusion

Modes of communication

- Unicast
  - 1:1 communication
  - Standard in the Internet



- Broadcast
  - 1:all
  - Discovery



- Anycast
  - 1:1 but don't care which one
  - improve speed or reduncancy

- Multicast
  - 1:group
  - address a group with something in common

Thank you!

academy@de-cix.net

Links and further reading

DE CIX

Links and further reading

- Internet protocol - https://en.wikipedia.org/wiki/Internet_Protocol
- Protocol stack - https://en.wikipedia.org/wiki/Protocol_stack
  - Transport Layer: https://en.wikipedia.org/wiki/Transport_layer
  - Datagram: https://en.wikipedia.org/wiki/Datagram
- IP Network Model: https://en.wikipedia.org/wiki/Internet_protocol_suite
- IPv4
  - IPv4 - https://en.wikipedia.org/wiki/IPv4
- IPv6
  - IPv6 itself - https://en.wikipedia.org/wiki/IPv6
  - IPv6 header - https://en.wikipedia.org/wiki/IPv6_packet
  - Transmission of IPv6 over Ethernet: https://tools.ietf.org/html/rfc2464
- History of Internet and IP
  - Internet Hall of Fame - https://internethalloffame.org
  - Defense Advanced Research Projects Agency (DARPA) - https://www.darpa.mil
  - ARPANET - https://www.darpa.mil/about-us/timeline/arpanet
  - The "Protocol Wars" - https://en.wikipedia.org/wiki/Protocol_Wars

Links and further reading

- List of IP protocol numbers
  - https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers
  - https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
- UDP - User Datagram Protocol
  - https://en.wikipedia.org/wiki/User_Datagram_Protocol
- TCP - Transmission Control Protocol
  - https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- ICMP - Internet Control Message Protocol
  - for IPv4: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
  - for IPv6: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_for_IPv6

Links and further reading

Unicast

- Wikipedia articles:
  - https://en.wikipedia.org/wiki/Point-to-point_(telecommunications)

Links and further reading

Broadcast

- Wikipedia articles:
  - https://en.wikipedia.org/wiki/Broadcasting_(networking)
  - https://en.wikipedia.org/wiki/Broadcast_address
- Ethernet related:
  - https://en.wikipedia.org/wiki/Broadcast_storm
- IP related:
  - Dynamic Host Configuration Protocol: RFC2131, Wikipedia
  - Address Resolution Protocol: RFC826, Wikipedia

Links and further reading

Multicast

- Wikipedia articles:
  - https://en.wikipedia.org/wiki/Multicast
  - https://en.wikipedia.org/wiki/Protocol_Independent_Multicast
  - https://en.wikipedia.org/wiki/Multicast_address
- Ethernet related:
- IP related:
  - https://en.wikipedia.org/wiki/IP_multicast
  - Overview of the Internet Multicast Addressing Architecture: RFC6308
  - IANA Guidelines for IPv4 Multicast Address Assignments: RFC5771
  - Unicast-Prefix-based IPv6 Multicast Addresses: RFC3306
  - Allocation Guidelines for IPv6 Multicast Addresses: RFC3307

Links and further reading

Anycast

- Wikipedia entry: https://en.wikipedia.org/wiki/Anycast
- Root name servers: https://en.wikipedia.org/wiki/Root_name_server
- K-Root name server: https://www.ripe.net/analyse/dns/k-root
- RFCs about Anycast:
  - Introduction: RFC1546
  - Operations: RFC4786
  - Architecture considerations: RFC7094
- Maximilian Wilhelm about Anycast (in German): https://blog.sdn.clinic/my-first-network/

Internet RFCs (Standards)

- There are too many RFCs dealing with IPv4 and IPv6 to be listed here
- Just go to https://tools.ietf.org/html/ and use the search field
- How does something become RFC? https://www.rfc-editor.org/pubprocess/
- The IETF - Internet Engineering Task Force