

Internet Communication Example

Suppose Athelred and Bartok are using a chat program to talk. Athelred is on a computer on a home LAN, Bartok is using a mobile device. Athelred's IP address is 12.34.56.78, and Bartok's is 98.76.54.32. Both chat programs use port 443.

Here's Athelred's chat window

ath_the_red: You watch that vulture documentary?

bart_is_ok: not yet still doing hwk

I  homework! LOL!_

Application Layer — Sending

The chat program on Athelred's computer needs to send the words Athelred just typed in a message to the other chat program, but to know what to show, the other program needs extra information, and for everything to be formatted in a way it can use, such as changing symbols into codes.

So let's say the message created by the chat program looks like:

```
FROM: ath_the_red | TO: bart_is_ok | MESSAGE: I  
[heart17] homework! LOL!
```

Transport Layer – Sending

The message now needs to be broken into packets by Athelred's computer so it can be sent to Bartok's computer. Each packet gets part of the data, and has a header with the sending socket address (IP address:port #), the receiving socket address, how many packets there were, and which packet it is.

So these packets might be:

Packet 1

```
packet 1 of 4
from: 12.34.56.78:443
to: 98.76.54.32:443
data: FROM: ath_the_red |
```

Packet 2

```
packet 2 of 4
from: 12.34.56.78:443
to: 98.76.54.32:443
data: TO: bart_is_ok | MES
```

Packet 3

```
packet 3 of 4
from: 12.34.56.78:443
to: 98.76.54.32:443
data: SAGE: I [heart17] ho
```

Packet 4

```
packet 4 of 4
from: 12.34.56.78:443
to: 98.76.54.32:443
data: nework! LOL!
```

Notice that the way the message is broken up doesn't care about the meaning, each packet just gets a certain amount of the data¹.

Internet Layer

Now the packets start from Athelred's computer and are moved across the internet to Bartok's by routers.

Packet 1 is sent from Athelred's computer. Since Athelred is on a LAN, it goes to the router on the LAN, which sends it on to a router on Athelred's ISP Regional Network. Each node looks at the destination IP address in the packet to decide which way the packet is supposed to go.

In those first steps, there was only one choice to send the packet towards its destination in each case. Once the packet is on the ISP Regional Network, however, there is probably enough redundancy that the routers will need to look at current traffic as well as destination IP address. Eventually it will reach routers on the backbone system. Then routers on Bartok's ISP Regional network. Finally it will reach the router Bartok's device is connected to, which has only one choice, and the packet reaches its destination.

Immediately after Packet 1 was sent, each of the others is sent as well. The router on the LAN will make the same choice for these packets as it did for Packet 1, there's no other way to send it but onto the ISP Regional Network.

Beyond that, however, things might be very different. Remember that routers don't try to keep the packets of one message

¹ Real packets are usually not as small as I used for this example, but are generally under 1500 bytes.

together (they don't even know there is even such a thing as a message!).

Since many other packets are moving through the ISP's routers, the traffic situation might change very quickly. So Packet 2 and Packet 3 might be sent in a different direction from Packet 1, and then Packet 4 might be sent in the same direction.

Each router they are sent to next makes a new choice for each one, still sending the packets in the right general direction, but using its current traffic information. So it is possible that in just a few steps, the four packets of this message might all be taking different routes.

Let's say that Packet 2 happens to be traveling over a cable which is cut by a backhoe before Packet 2 can arrive at the next router.

Transport Layer — Receiving Computer

Since each packet took a different route, they might arrive at Bartok's device out of order. Let's say that Packet 3 arrives first. Now Bartok's device knows there should be 4 packets, because that information is the header.

Soon after, Packet 1 and Packet 4 arrive. But after a short time Packet 2 still has not arrived. So Packet 2 is re-sent.

Once it has all the packets, Bartok's device reassembles them into the original message:

```
FROM: ath_the_red | TO: bart_is_ok | MESSAGE: I  
[heart17] homework! LOL!
```

Based on the port number (443) it also knows to give this message to Bartok's chat program.

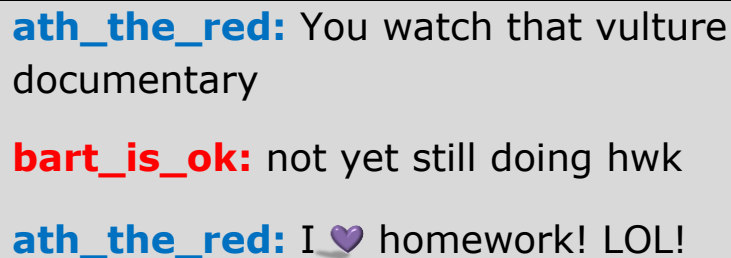
Application Layer – Receiving

Now Bartok's chat program has the message, which it needs to act on. Since it knows the protocol, it knows what the message is telling it.

```
FROM: ath_the_red | TO: bart_is_ok | MESSAGE: I  
[heart17] homework! LOL!
```

So it knows it should show that the words were typed by Athelred, what the words were, and that it should show heart symbol 17 in the middle of the text.

So Bartok's chat program will now look like



The chat window displays three messages in a light gray background. The first message is from 'ath_the_red' in blue text: 'You watch that vulture documentary'. The second message is from 'bart_is_ok' in red text: 'not yet still doing hwk'. The third message is from 'ath_the_red' in blue text: 'I ❤️ homework! LOL!'. Below the messages is a white input field with a small gray cursor line on the left.