

Network Layer: Intro

Network Layer

The purpose of the network layer is to get a segment from one place to another. On the sending side, segments are encapsulated into datagrams, and on the receiving side they are delivered to the transport layer. The endpoints implement 5-layer stacks, but intermediate devices only implement 3 (physical, data link, network).

The network layer is nearly always implemented fully in software.

IP is pretty much the only network layer protocol (even for mobile networks nowadays).

Forwarding vs. Routing

Forwarding is just moving packets from router's input queue to the output queue, whereas routing involves making decision about the route the packets should take.

Forwarding should be very fast – it's usually implemented in hardware, using a forwarding table. Routing can happen on a slower timescale, as routes are fairly static – typically it involves changing/populating the forwarding table.

Routing can be described as part of the control plane, whereas forwarding is part of the data plane. This distinction is useful for software defined networks, where it helps to keep the planes separate.

Service Model

What services should the network layer provide?

Example services for individual datagrams:

- guaranteed delivery
- minimum delay

Example services for a flow of datagrams:

- in-order delivery
- guaranteed minimum bandwidth
- restrictions on jitter (variation in inter-packet spacing)

But would you want to provide them? What's the complexity?

- For minimum delay, we would need admission control – refusing users until the network is less busy.
- For in-order delay we would need sequence numbers.

IP doesn't provide any of these services – the idea is that if you're interested in these services, you should implement them in an end-to-end way.

- The creators didn't want to restrict the networks on which IP can be run.