How Networks Differ

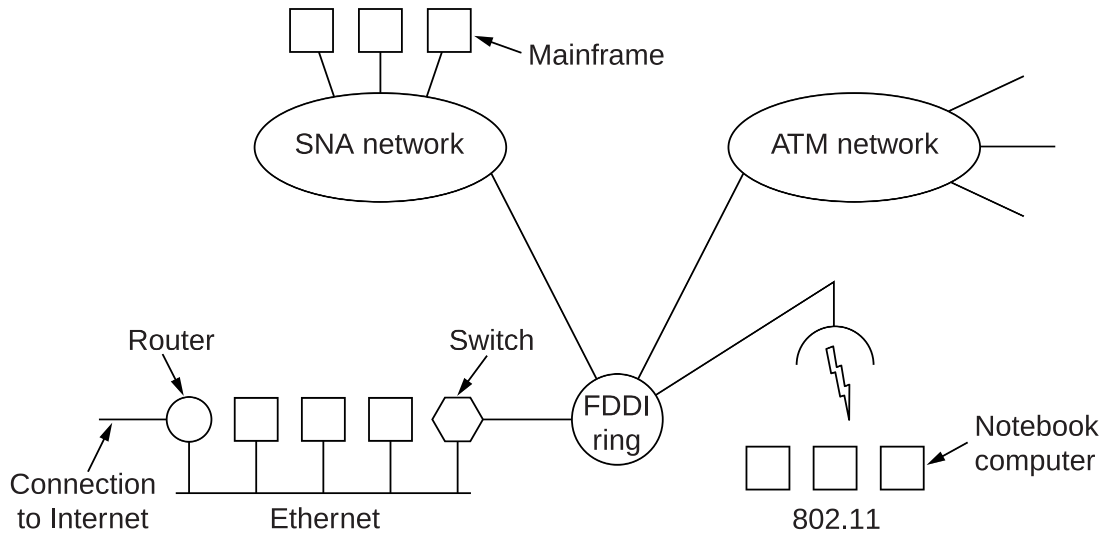
- Differences that can occur at network layer, which makes internetworking difficult:

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

- It is impossible to resolve all differences, and the solution is to take a simple approach (as in the Internet)

Internetworking

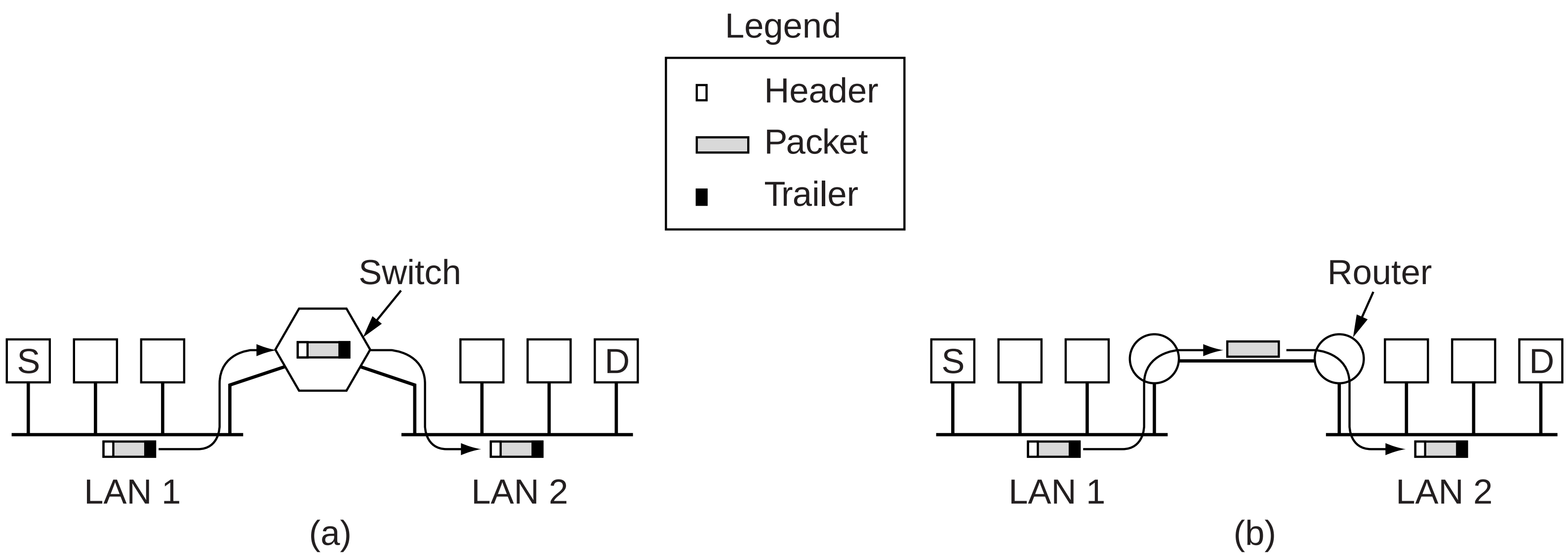
- **Problem**: Networks with different protocol stacks → how to let them talk to each other?
- **Nonsolution**: Why not enforce all networks to run same protocol stack? → ask for troubles ! and this is effectively saying no progress is allowed
- **Solution**: Construct some *gateways* that connect different kinds of networks
  - *Repeaters or hubs* at physical layer: operate on bits, do not understand protocols, just regenerate signals
  - *Bridges or switches* at data link layer: operate on frames, examine MAC addresses, do minor protocol translation, e.g. Ethernet to 802.11
  - **Multiprotocol router** at network layer: operate on packets, translate between different packet formats, possibly split up packets



  - *Transport gateways* at transport layer: coupling byte streams in different networks, interface between different transport connections
  - *Application gateways* at application layer: translate message semantics between OSI and TCP/IP networks

Internetworking at Network Layer

- With a router, packet is extracted from frame and address in packet is used to decide where to send it → so router has to understand network protocols

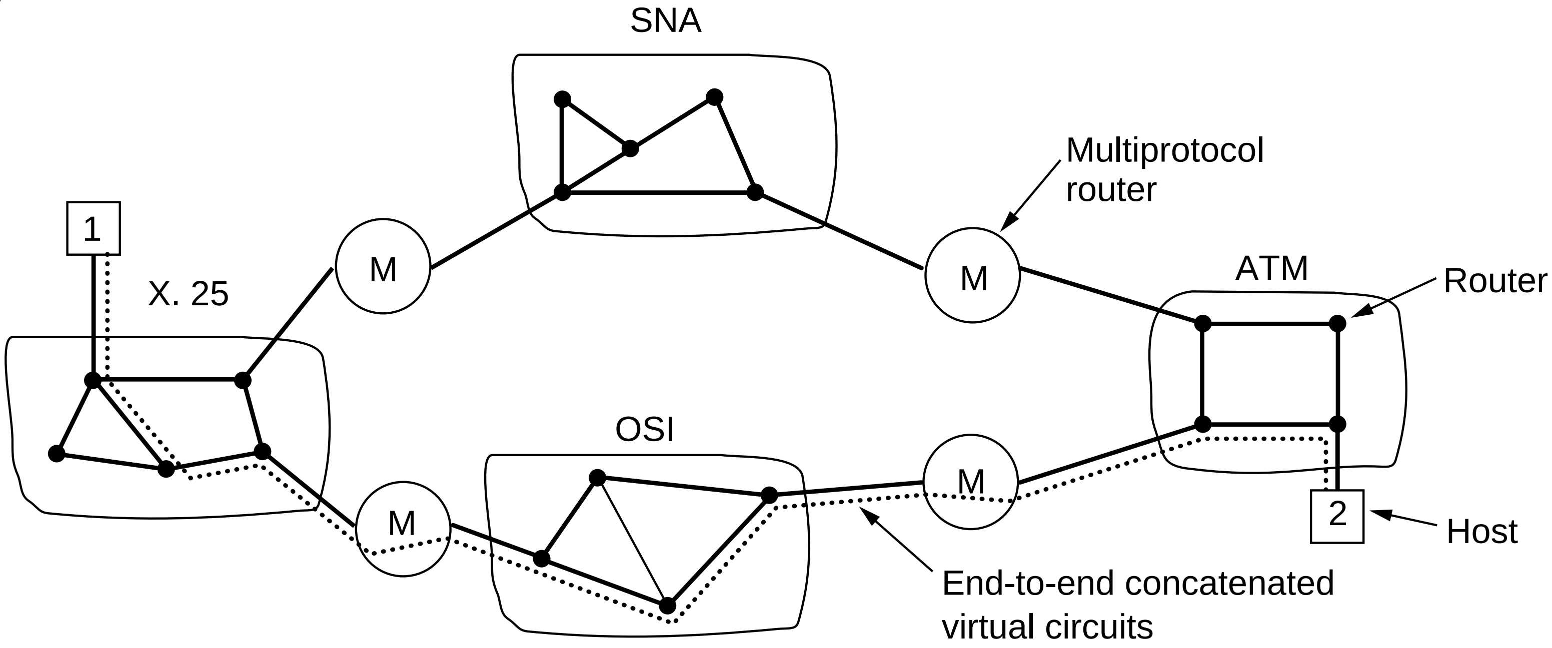


- Switch or bridge do not understand network protocols. In a switch (or bridge) entire frame is transported on basis of its MAC address

- Two basic internetworking approaches: connection-oriented concatenation of virtual-circuit subnets, and connectionless datagram internet style

Concatenated Virtual Circuits

- A sequence of virtual circuits is set up from source through gateways (multiprotocol routers) to destination

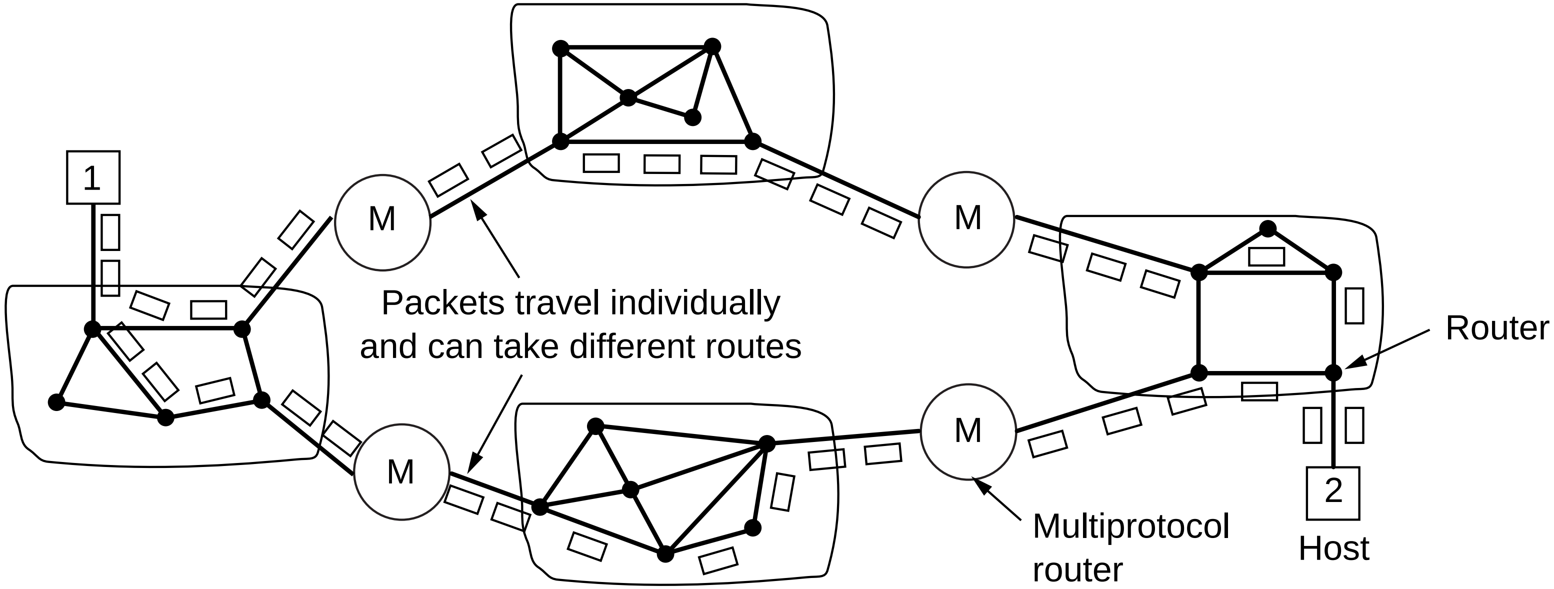


Each gateway maintain tables telling which virtual circuits pass through it, where they are to be routed and what new virtual circuit number is

- If one of constituent subnets does not support virtual circuits, simple concatenation becomes hard

Using Datagram

- This connectionless internetworking approach is similar to datagram service in subnets

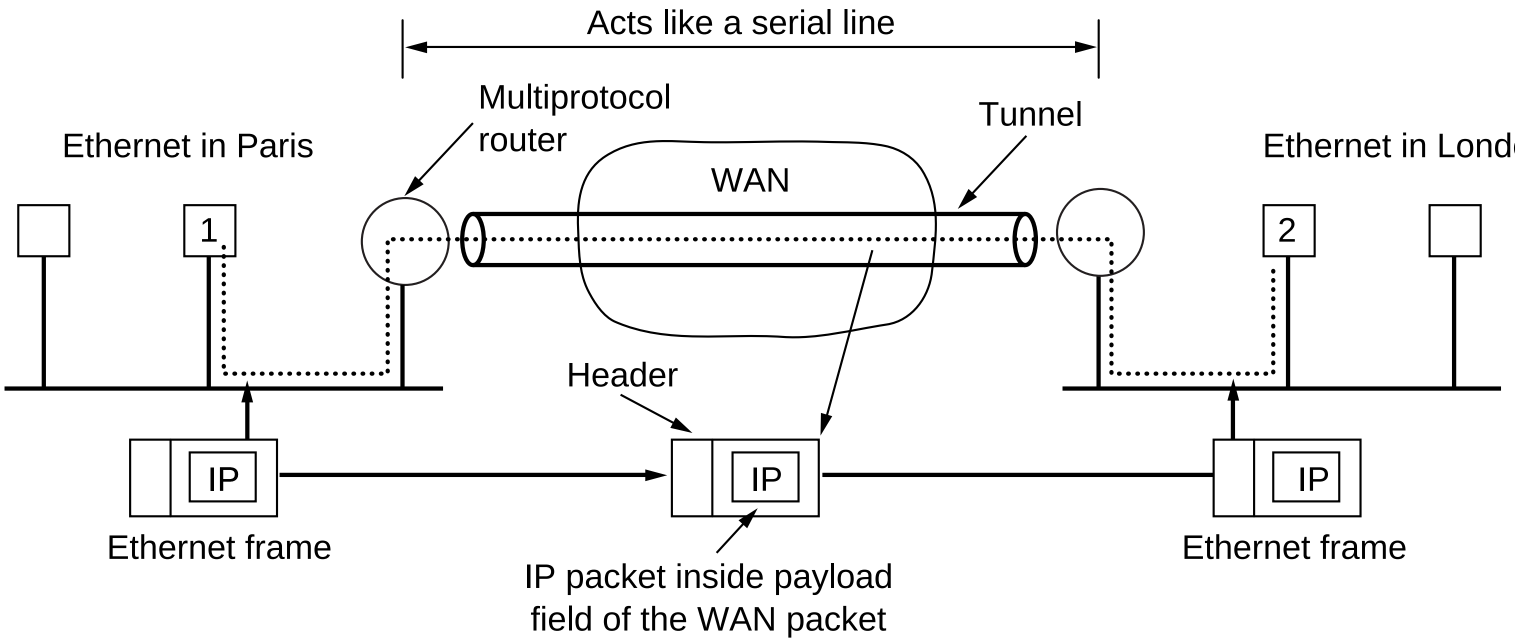


- *Main problem*: Addressing – different networks use completely different address systems. How do you address a host in a IP subnet when you are in a SNA subnet?

- *Solution*: Don't solve it, instead consider possibly some universal network protocol, e.g. IP, which can be carried through many networks

Tunneling

- Solving general internetworking problem is extremely hard, but thing becomes easier if source and destination are on same type of subnet – **tunnelling** can be used

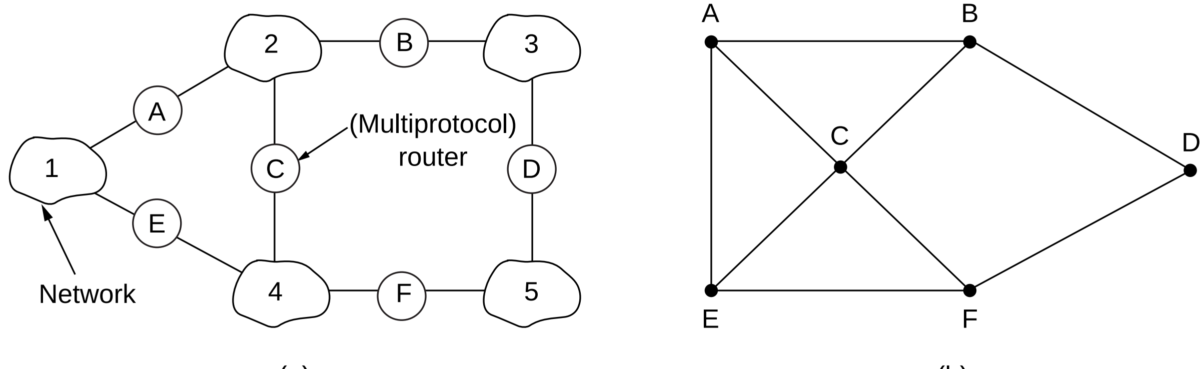


- Multiprotocol router simply places source host's packet inside payload of WAN packet and sends through

- Essentially, what between two hosts' subnets, WAN, can be seen as a "tunnel" extending from one multiprotocol router to the other

Internetworking Routing

- Routing through an internetwork is similar to routing within a subnet but with added complications
- Two-level routing algorithm needed
  - *Interior gateway protocol* used within each network
  - *Exterior gateway protocol* used between networks

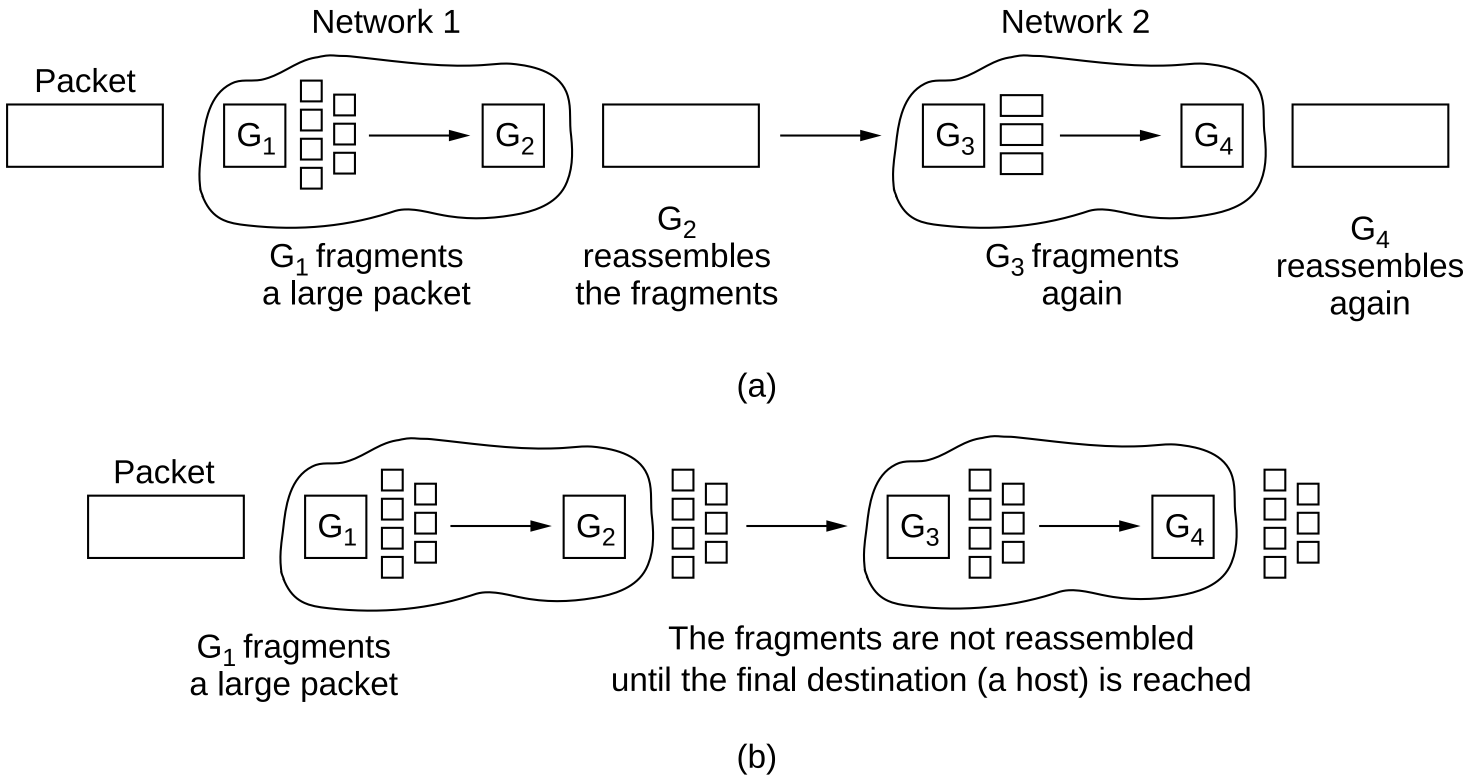


(a) (b)

- Thus, one has intranetwork routing and internetwork routing
- Consider an internet packet starts out from its LAN addressed to the local multiprotocol routine (in the MAC layer header)
  - After the packet has reached the local router, the network layer decides which mutliprotocol router to forward the packet to using its own routing table
  - If that router can be reached using the packet's native network protocol, the packet is directly forward there
  - Otherwise, it is tunnelled there (see previous slide), encapsulated in the protocol required by the intervening network

Fragmentation

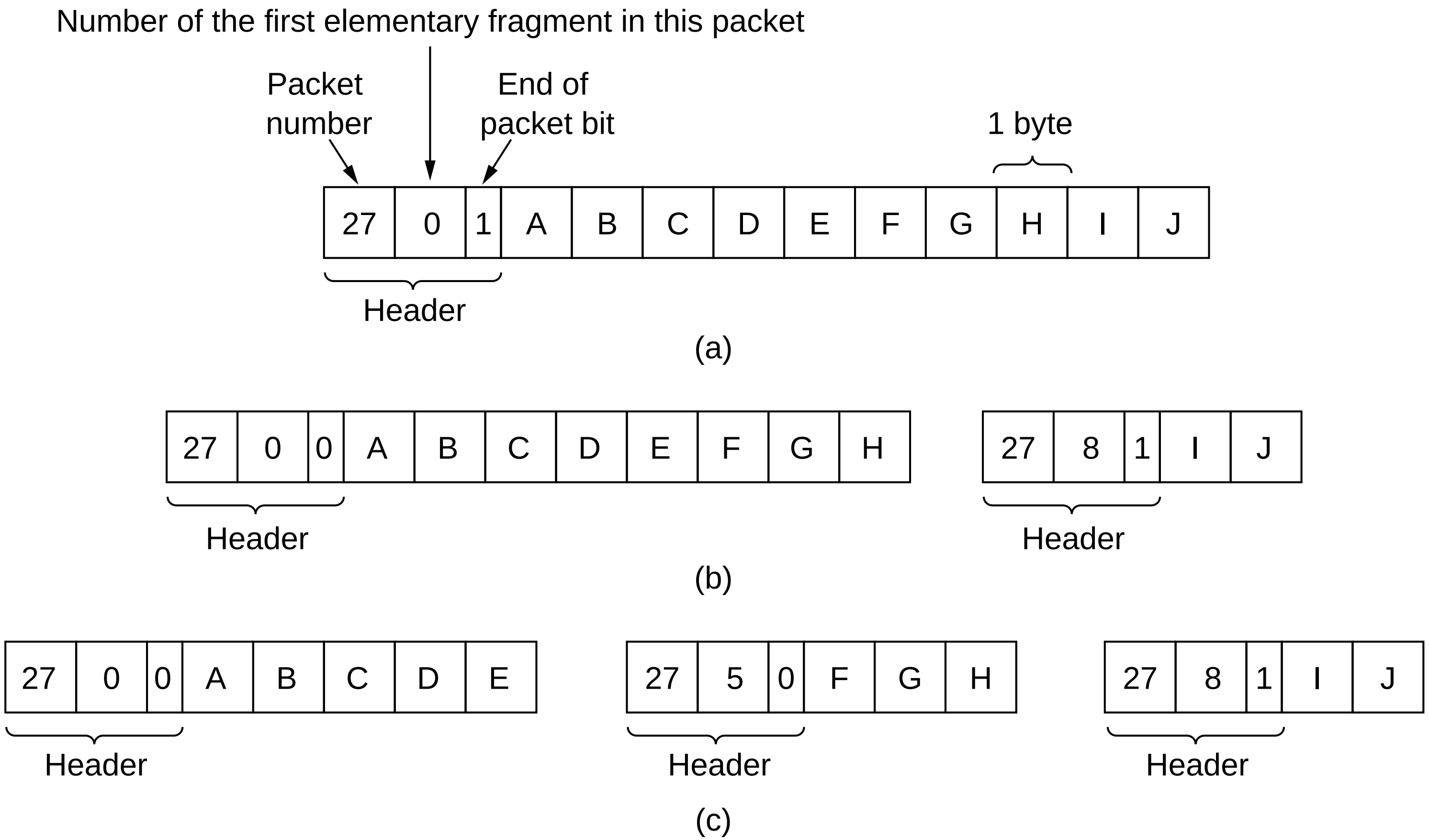
- Different networks have different maximum packet sizes, and solution is to allow gateway to split a packet into smaller ones ⟶ **fragmentation**



- **Transparent fragmentation**: "small-packet" network splits packet and recombines fragments back. Others do not need to know
- **Nontransparent fragmentation**: destination host recombines fragments back

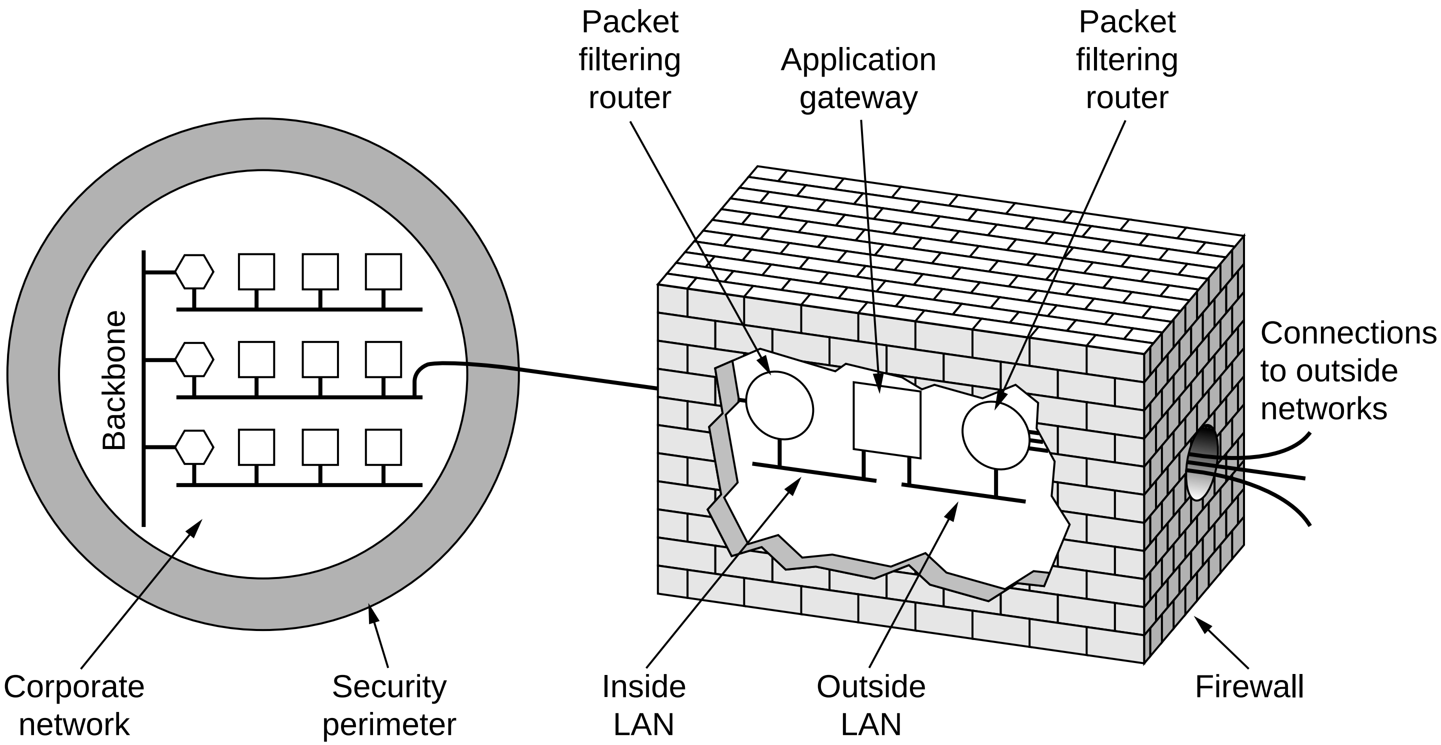
Fragmentation → Reassembly

- A fragment may be fragmented again by successive intermediate networks → when a fragment arrives at destination, we have to know exactly where it fits into original packet
- **Solution**: requires two sequence fields in header, original packet number and fragment number



Firewalls

- Sometimes it is better not all your computers can be easily reached by others, or vice versa → **firewall**



- Packet filter: standard router with extra functionality to inspect packets → Those packets meeting some criterion are forwarded normally, and those failed are dropped
- Application gateway: operates at application level

Summary

- Internetworking in general
- Internetworking at network layer

  Concatenated virtual circuits

  Connectionless internetworking

  Tunnelling

  Internetwork routing
- Transparent and nontransparent fragmentation, reassembly
- Firewalls