

# Computer Networks

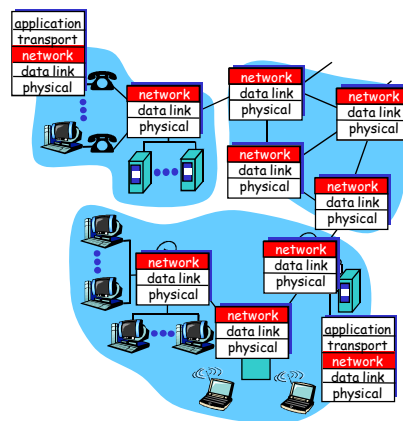
## Network Layer, Virtual Circuits and Datagram Networks

Based on Computer Networking, 4<sup>th</sup> Edition by Kurose and Ross

Stan Kurkovsky

### Network layer

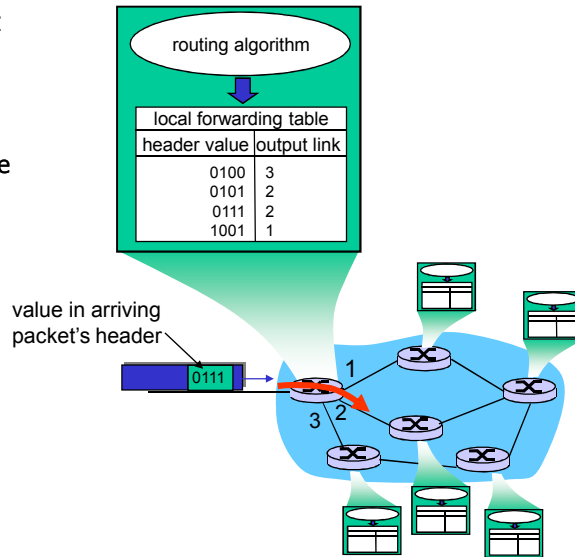
- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on rcving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- Router examines header fields in all IP datagrams passing through it



Stan Kurkovsky

## Key Network-Layer Functions

- *forwarding*: move packets from router's input to appropriate router output
  - Takes place inside a router
  - **analogy**: process of planning trip from source to dest
- *routing*: determine route taken by packets from source to dest
  - Takes place between routers
  - *Routing algorithms*
  - **analogy**: process of getting through single interchange



## Network service model

Q: What *service model* for "channel" transporting datagrams from sender to rcvr?

### Example services for individual datagrams:

- Guaranteed delivery
- Guaranteed delivery with less than 40 msec delay

### Example services for a flow of datagrams:

- In-order datagram delivery
- Guaranteed minimum bandwidth to flow
- Restrictions on changes in inter-packet spacing

Stan Kurkovsky

## Network layer service models

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Stan Kurkovsky

## Network layer connection and connection-less service

- Datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- Analogous to the transport-layer services, but:
  - **Service:** host-to-host
  - **No choice:** network provides one or the other
  - **Implementation:** in the core
- VC: ATM networks
  - Asynchronous Transfer Mode

Stan Kurkovsky

## Virtual circuits

"source-to-dest path behaves much like telephone circuit"

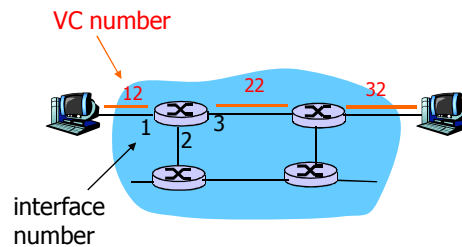
- performance-wise
- network actions along source-to-dest path
- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC

A VC consists of:

1. Path from source to destination
  2. VC numbers, one number for each link along path
  3. Entries in forwarding tables in routers along path
- Packet belonging to VC carries a VC number.
  - VC number must be changed on each link.
    - New VC number comes from forwarding table

Stan Kurkovsky

## Forwarding table



Forwarding table in northwest router:

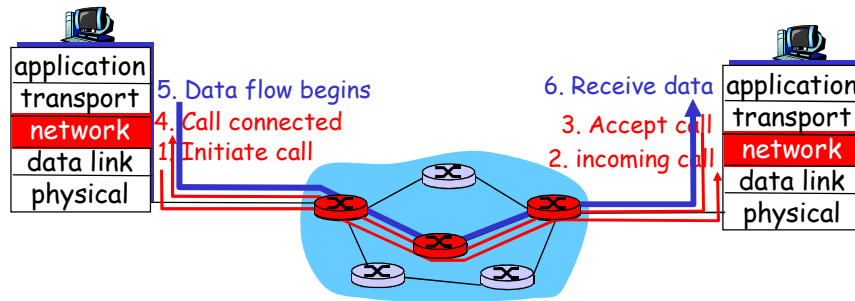
Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

**Routers maintain connection state information!**

Stan Kurkovsky

## Virtual circuits: signaling protocols

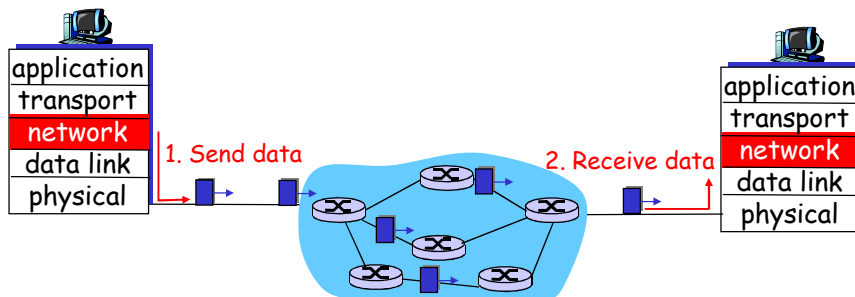
- used to setup, maintain, teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



Stan Kurkovsky

## Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths



Stan Kurkovsky

## Forwarding table

4 billion possible entries

<u>Destination Address Range</u>	<u>Link Interface</u>
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Stan Kurkovsky

## Longest prefix matching

<u>Prefix Match</u>	<u>Link Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

### Examples

DA: 11001000 00010111 00010110 10100001

DA: 11001000 00010111 00011000 10101010

Stan Kurkovsky

## Datagram or VC network: why?

### Internet

- data exchange among computers
  - "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

### ATM

- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network

Stan Kurkovsky