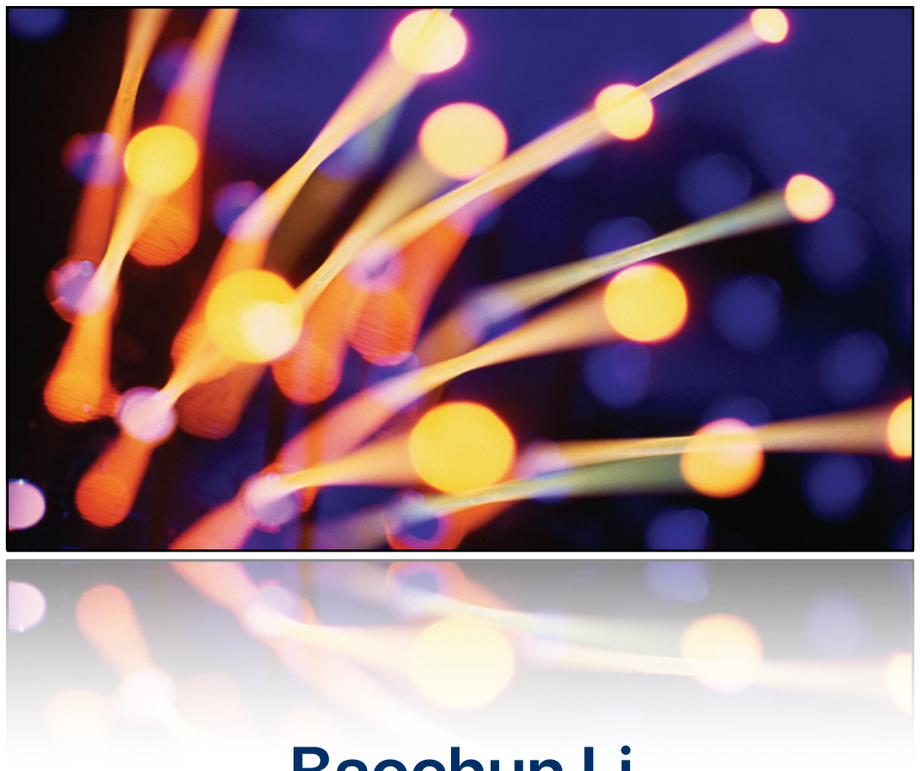
Episode 3. Principles in Network Design

Part 3



Baochun Li

Department of Electrical and Computer Engineering

University of Toronto

Case study: how are design principles applied in the network layer?

Reading: Salzer 7.4, 7.4.2, 7.4.3

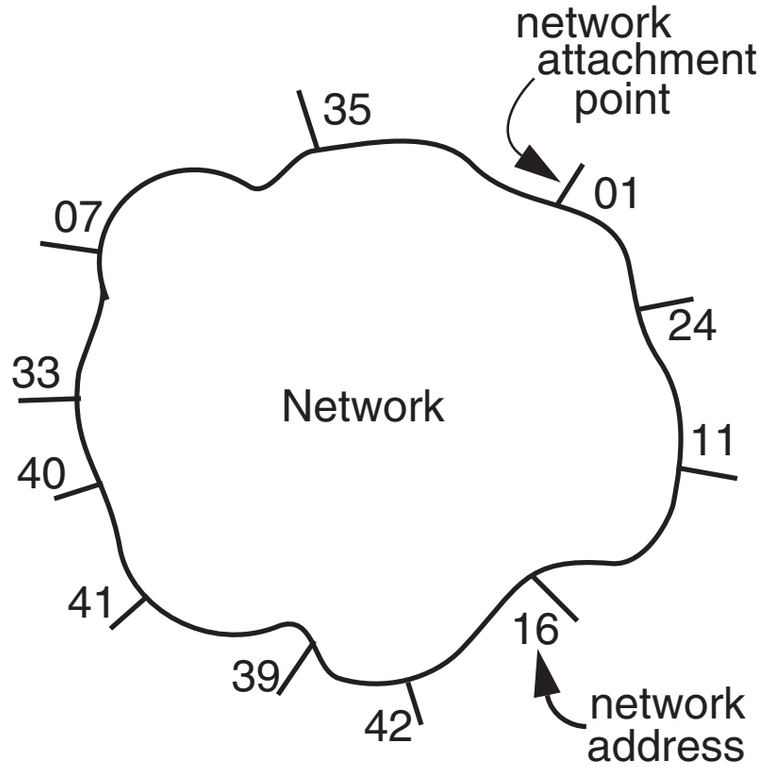
Addressing Interface

The conceptual model of a network

a cloud bristling with network attachment points identified by numbers known as network addresses

A **segment** enters the network at one attachment point, known as the **source**

The network layer wraps the segment in a packet and carries the packet across the network to another attachment point, known as the **destination**, where it unwraps the original segment and delivers it to the end-to-end layer



Managing the Forwarding Table: Routing

The primary challenge in a packet forwarding network is to set up and manage the forwarding tables

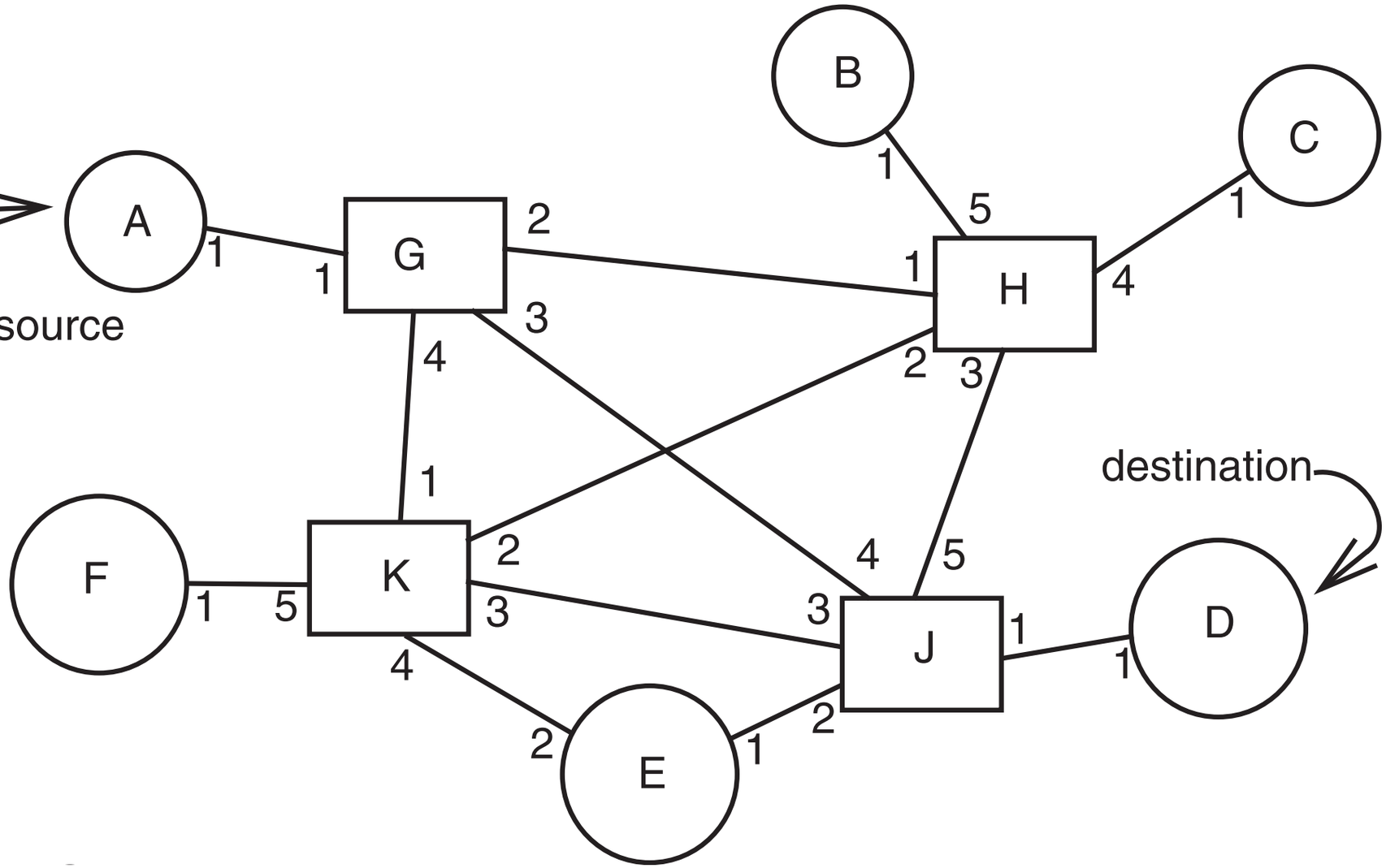
Constructing these tables requires first figuring out appropriate paths (sometimes called routes) to follow from each source to each destination — routing

Setting these tables by hand is not scalable!

- When links are added, removed, failed or repaired, the forwarding tables need to be recalculated
- It would be nice for forwarding tables to automatically adapt to avoid congestion

A packet forwarder that also participates in an adaptive routing algorithm is called a router

Network-layer routing protocols



destination	link
A	end-layer
all other	1

Routing protocol design: challenges

How to construct a consistent, efficient set of forwarding tables — so that there are no loops in routes?

What defines a “better” routing protocol?

A smaller number of hops to the destination

Adaptive routing: able to adapt to a change in topology

Scalability: handles a large number of destinations

Path vector algorithm (e.g., the Border Gateway Protocol)

exchanges information of about 100,000 routes in the core of the Internet

The path vector algorithm

Basic idea —

Each participant maintains, in addition to its forwarding table, a **path vector**, each element of which is a complete path to some destination

Initially, the only path it knows about is the zero-length path to itself

As the algorithm proceeds it gradually learns about other paths

Eventually its path vector accumulates paths to every point in the network

After each step of the algorithm it can construct a new forwarding table from its new path vector, so the forwarding table gradually becomes more and more complete!

Two steps —

advertising and **path selection**

Step 1: Advertising

In the advertising step, each participant sends its own network address and a copy of its path vector down every attached link to its immediate neighbours

From A, via link 1

to	path
A	< >

From H, via link 2:

to	path
H	< >

From J, via link 3:

to	path
J	< >

From K, via link 4:

to	path
K	< >

to	path
G	< >

Initial state of path vector for G.

Path vector received by G in the first round

path vector

to	path
A	<A>
G	< >
H	<H>
J	<J>
K	<K>

forwarding table

to	link
A	1
G	end-layer
H	2
J	3
K	4

First-round path vector and forwarding table for G.

Step 2. Path Selection

G now performs the path selection step by merging the information received from its neighbours with that already in its own previous path vector

To do this merge, G takes each received path, prepends the network address of the neighbour that supplied it, and then decides whether or not to use this path in its own path vector

For previously unknown destinations, the answer is yes

For previously known destinations, G compares the paths that its neighbours have provided with the path it already had in its table, to see if the neighbour has a better path (e.g., a smaller number of hops to the destination)

Each router discards any paths that a neighbour stops advertising — to discard links that go down

Loops are still possible in forwarding tables

Temporary loops are still possible

If a link has gone down, some packets may loop for a while until everyone agrees on the new forwarding tables!

Solution: hop limit

Add a field to the network-layer header containing a hop limit counter

Decrements the hop limit counter by each router

If a router finds it to be zero, it discards the packet

Introducing hierarchies

Two problems in our solutions so far —

- Every attachment point must have a unique address — it is hard to maintain a complete and accurate list of addresses already assigned when the number of addresses is large
- The path vector grows in size with the number of attachment points

Solution: introducing hierarchies

- Network addresses should be designed to have a hierarchical structure
- Both for decentralizing address assignments and for reducing the size of forwarding tables and path vectors

Hierarchical address assignment: benefits

Assume that we have two hierarchies — a "region" and a "station"

We may assign to A the network address "11, 75", where 11 is a region identifier and 75 is a station identifier

Key benefit — reduction of path vectors

If we can adopt a policy that regions must correspond to the set of network attachment points served by a group of closely-connected routers, we can use it to reduce the size of forwarding tables and path vectors

For example, when a router for region 11 gets ready to advertise its path vector to a router serving region 12, it can condense all paths for region 11 into a single path

Hierarchical address assignment: benefits

Now the problem of assigning unique addresses in a large network is also solved

- The “station” part of a network address needs to be unique only within its region
- A central authority assigns region identifiers
- Local authorities assign station identifiers within each region

Hierarchical address assignment: complexities

The table lookup process is more complicated

the forwarder needs to first extract the region component of the destination address, and look that up in its forwarding table

Either the forwarding table contains an entry showing a link over which to send the packet to that region

Or the forwarding table contains an entry saying that this forwarder is already in the destination region — it is now necessary to extract the station identifier and look that up in a different part of the forwarding table

The addresses are becoming geographically dependent

Paths may no longer be the shortest possible

Reading: Saltzer 7.4, 7.4.2, 7.4.3