

More on Network Routing and Internet Protocol

Kai Shen

9/26/2011 CSC 257/457 - Fall 2011 1

Network Routing

"Link state" routing:

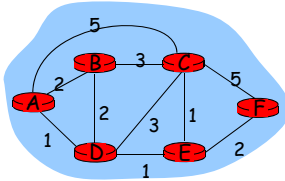
- Dijkstra's algorithm - efficient approach to calculate least cost routes
- all routers need complete topology, link cost info
 - ⇒ costly (or impossible) to acquire such information in large networks

Decentralized routing:

- router need to knows physically-connected neighbors, link costs to neighbors
 - ⇒ doable by info exchanges between neighbor routers

9/26/2011 CSC 257/457 - Fall 2011 2

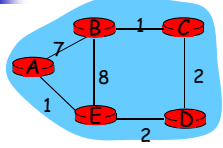
Distance Vector Routing



- routing table** (at each host): the next hop for each destination in the network
- distance vector routing:**
 - routing table can be derived from the distance vector at each node
 - distance vectors can be maintained in a decentralized fashion

9/26/2011 CSC 257/457 - Fall 2011 3

Distance Vector: An Example



Distance vector at node E

D (E,x,y)	via first hop x		
	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

$D(E,D,C) = c(E,D) + \text{shortest}(D,C) = 2+2 = 4$
 $D(E,D,A) = c(E,D) + \text{shortest}(D,A) = 2+3 = 5$
 $D(E,B,A) = c(E,B) + \text{shortest}(B,A) = 8+6 = 14$

9/26/2011 CSC 257/457 - Fall 2011 4

Distance Vector to Routing Table

		cost to destination via				
$D^E()$		A	B	D	Outgoing link to use, cost	
destination	A	1	14	5	A	A,1
	B	7	8	5	B	D,5
	C	6	9	4	C	D,4
	D	4	11	2	D	D,4

Distance vector → Routing table

9/26/2011 CSC 257/457 - Fall 2011 5

Distance Vector Routing: Constructing Distance Vector

Distance vector can be built on information about local links and neighbors' routing tables!

At node X, each entry $D(X,Y,Z)$ includes:

- the distance from X to neighbor Y
- the shortest distance from Y to Z

9/26/2011 CSC 257/457 - Fall 2011 6

Distance Vector Algorithm: An Example

D^X		cost via	
		Y	Z
d	e	2	∞
e	s	∞	7
s	t	∞	7

D^Y		cost via	
		X	Z
d	e	2	∞
e	s	∞	1
s	t	∞	1

D^Z		cost via	
		X	Y
d	e	7	∞
e	s	∞	1
s	t	∞	1

$D(X,Z,Y) = c(X,Z) + \text{shortest}(Z,Y) = 7 + 1 = 8$
 $D(X,Y,Z) = c(X,Y) + \text{shortest}(Y,Z) = 2 + 1 = 3$

9/26/2011 CSC 257/457 - Fall 2011 7

Distance Vector Algorithm: An Example

D^X		cost via	
		Y	Z
d	e	2	∞
e	s	∞	7
s	t	∞	7

D^Y		cost via	
		X	Z
d	e	2	∞
e	s	∞	1
s	t	∞	1

D^Z		cost via	
		X	Y
d	e	7	∞
e	s	∞	1
s	t	∞	1

Are we done?

9/26/2011 CSC 257/457 - Fall 2011 8

Distance Vector Routing: Maintaining Distance Vector

Information:

- local link cost
- distance vector → routing table

Recompute caused by:

- local link cost change
- message from neighbor: its routing table has changed

Notify neighbor:

- if my own routing table has changed, notify my neighbors

Each node:

wait for (change in local link cost or msg from neighbor)

recompute distance vector

if the local routing table has changed, notify neighbors

9/26/2011
CSC 257/457 - Fall 2011
9

Distance Vector: Link Cost Changes

Link cost changes:

- node detects local link cost change
- recompute - updates local distance vector
- if cost change in least cost path, notify neighbors

$\begin{matrix} D^Y & \text{via} & X & Z \\ X & (4) & 6 \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & (1) & 6 \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & (1) & 6 \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & (1) & 3 \end{matrix}$	algorithm terminates
$\begin{matrix} D^Z & \text{via} & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (2) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (2) \end{matrix}$	

time t_0 t_1 t_2

$c(X,Y)$ change

9/26/2011
CSC 257/457 - Fall 2011
10

Distance Vector: Link Cost Changes

A pathological case:

- recursive distance vector updates
- applies only to link cost increase ⇒ bad news settles slowly

$\begin{matrix} D^Y & \text{via} & X & Z \\ X & (4) & 6 \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (6) \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (6) \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (8) \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (8) \end{matrix}$	algorithm continues on!
$\begin{matrix} D^Z & \text{via} & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (7) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (7) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (9) \end{matrix}$	

time t_0 t_1 t_2 t_3 t_4

$c(X,Y)$ change

9/26/2011
CSC 257/457 - Fall 2011
11

Distance Vector: Link Cost Changes

If Z routes through Y to get to X :

- Z tells Y its (Z's) distance to X is infinity (so Y won't route to X via Z)
- Will this completely solve the problem of recursive distance vector updates?

$\begin{matrix} D^Y & \text{via} & X & Z \\ X & (4) & \infty \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & \infty \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & \infty \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (51) \end{matrix}$	$\begin{matrix} D^Y & X & Z \\ X & 60 & (51) \end{matrix}$	algorithm terminates
$\begin{matrix} D^Z & \text{via} & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & (5) \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 60 & 61 \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 60 & 61 \end{matrix}$	$\begin{matrix} D^Z & X & Y \\ X & 50 & \infty \end{matrix}$	

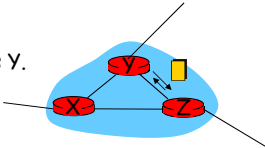
time t_0 t_1 t_2 t_3 t_4

$c(X,Y)$ change

9/26/2011
CSC 257/457 - Fall 2011
12

Routing Loops

- In packet switching networks, each node computes its own routing table independently
- What if?
 - Y's next hop to X is Z;
 - and Z's next hop to X is Y.
- Routing loops
 - during recursive distance vector updates
 - unsynchronized information updates in dynamic networks



9/26/2011 CSC 257/457 - Fall 2011 13

Comparison of LS and DV Algorithms

Speed of convergence

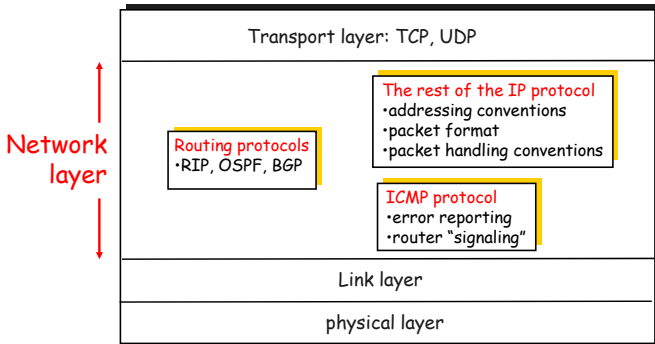
- Link state routing:
O(nlogn) algorithm in local computation
- Distance vector routing:
 - convergence time?
 - recursive distance vector updates in a pathological case

Robustness: what happens if router malfunctions?

- Link state routing: node can advertise incorrect link cost
- Distance vector routing: node can advertise incorrect routing table

9/26/2011 CSC 257/457 - Fall 2011 14

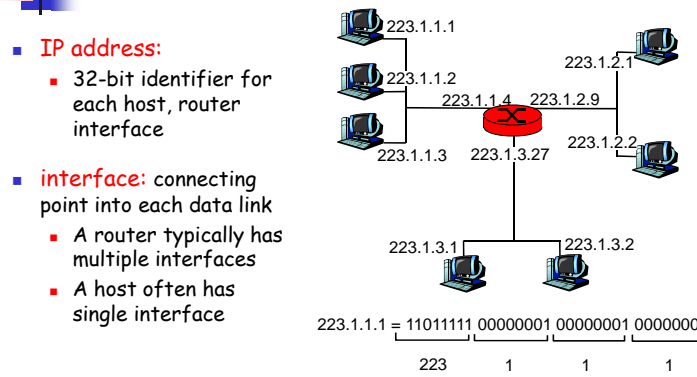
The Internet Network Layer



9/26/2011 CSC 257/457 - Fall 2011 15

IP Addressing: Introduction

- **IP address:**
 - 32-bit identifier for each host, router interface
- **interface:** connecting point into each data link
 - A router typically has multiple interfaces
 - A host often has single interface



9/26/2011 CSC 257/457 - Fall 2011 16

IP Network and Hierarchical Addressing

- What's an IP network?
 - can physically reach each other without intervening router (*intervening switches?*)
- IP address:
 - network part (high order bits); host part (low order bits)
 - devices with same network part of IP address are in the same IP network

network consisting of 3 IP networks

9/26/2011 CSC 257/457 - Fall 2011 17

IP Addresses: Original Standard

Given notion of "network", let's re-examine IP addresses:

"class"-based addressing:

class				
A	0	network	host	1.0.0.0 to 127.255.255.255
B	10	network	host	128.0.0.0 to 191.255.255.255
C	110	network	host	192.0.0.0 to 223.255.255.255
D	1110	multicast address		224.0.0.0 to 239.255.255.255

← 32 bits →

9/26/2011 CSC 257/457 - Fall 2011 18

IP Addressing: CIDR

- "Class"-based addressing:
 - inefficient use of address space, address space exhaustion
 - e.g., class B network allocated enough addresses for 65K hosts, even if only 2K hosts in that network
- **CIDR**: classless addressing
 - network portion of address of arbitrary length
 - address format: **a.b.c.d/x**, where x is # bits in network portion of address

11001000 00010111 00010000 00000000

200.23.16.0/23

9/26/2011 CSC 257/457 - Fall 2011 19

IP Address Depletion (DHCP)

Address depletion ⇒ 32-bit address space soon to be used up.

Observation: not every host is online at a given time.

DHCP: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected)

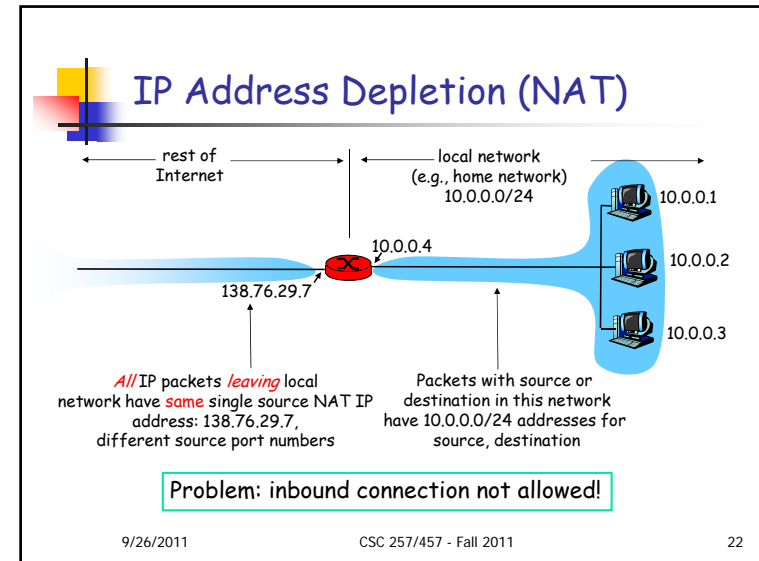
9/26/2011 CSC 257/457 - Fall 2011 20

IP Address Depletion (NAT)

Observations:

- A lot of traffic is local
- Although IP addresses are few, possible (IPaddr, port) tuples are more abundant and they can identify comm. end point

9/26/2011 CSC 257/457 - Fall 2011 21



Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).

9/26/2011 CSC 257/457 - Fall 2011 23