CS2105 Lecture 7
Routing

3 March, 2014

CS2105 Lecture 7

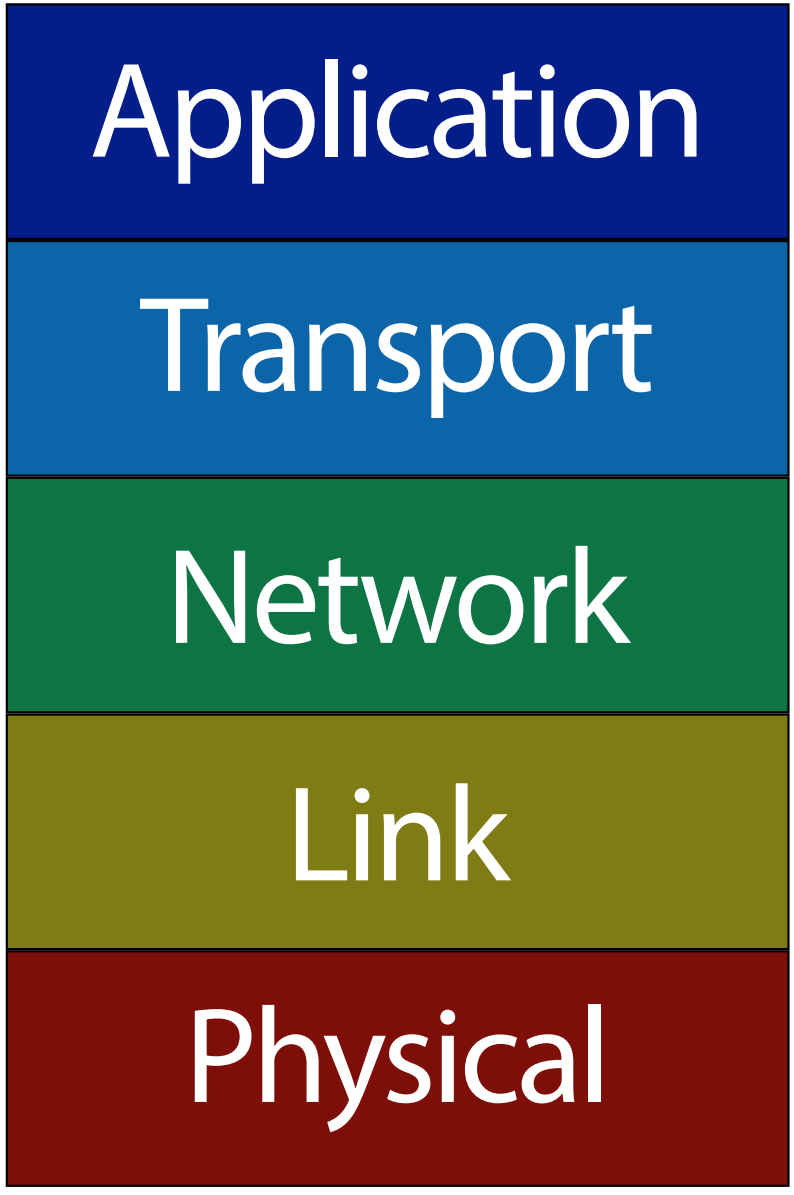
Routing

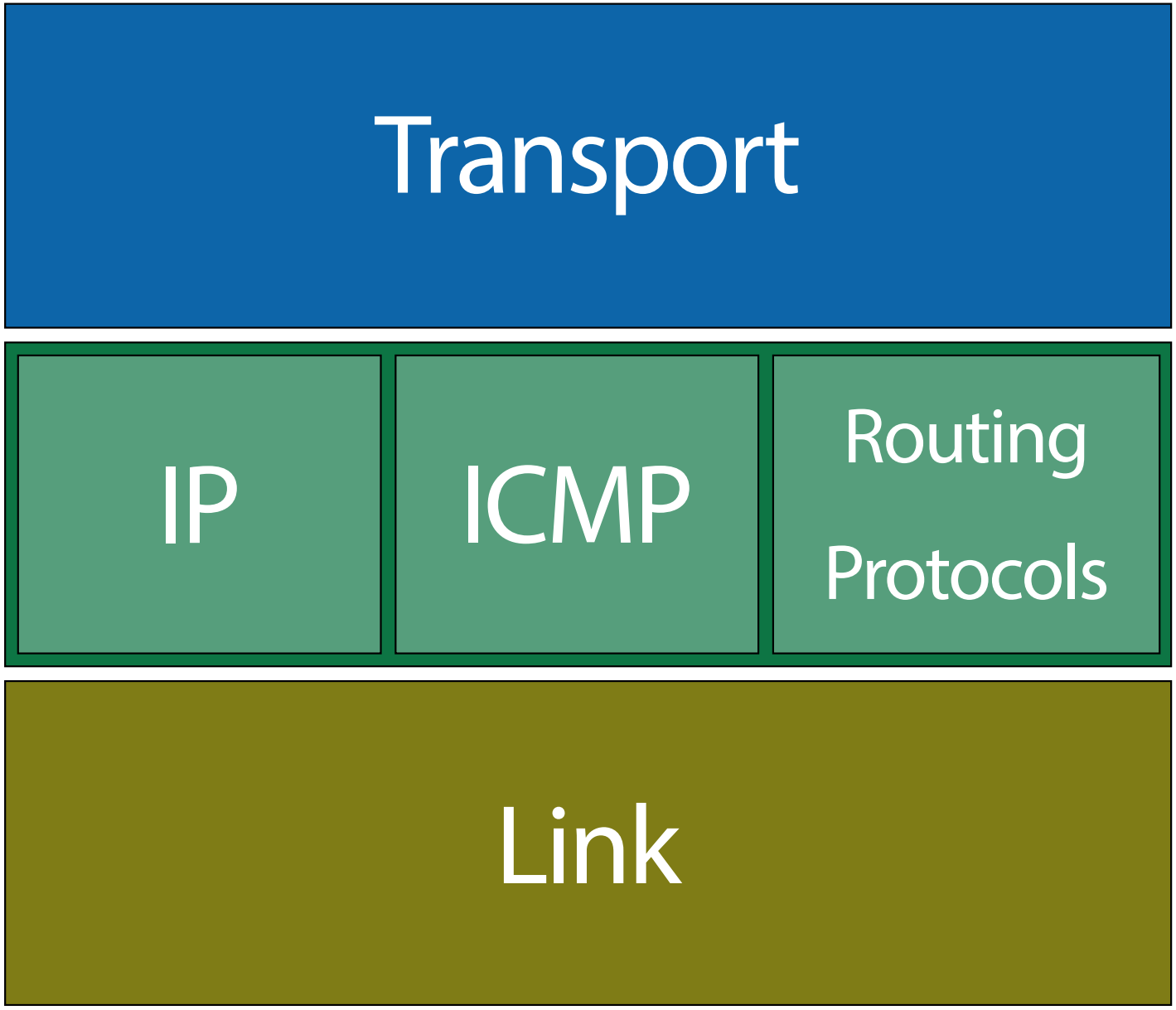
3 March, 2014

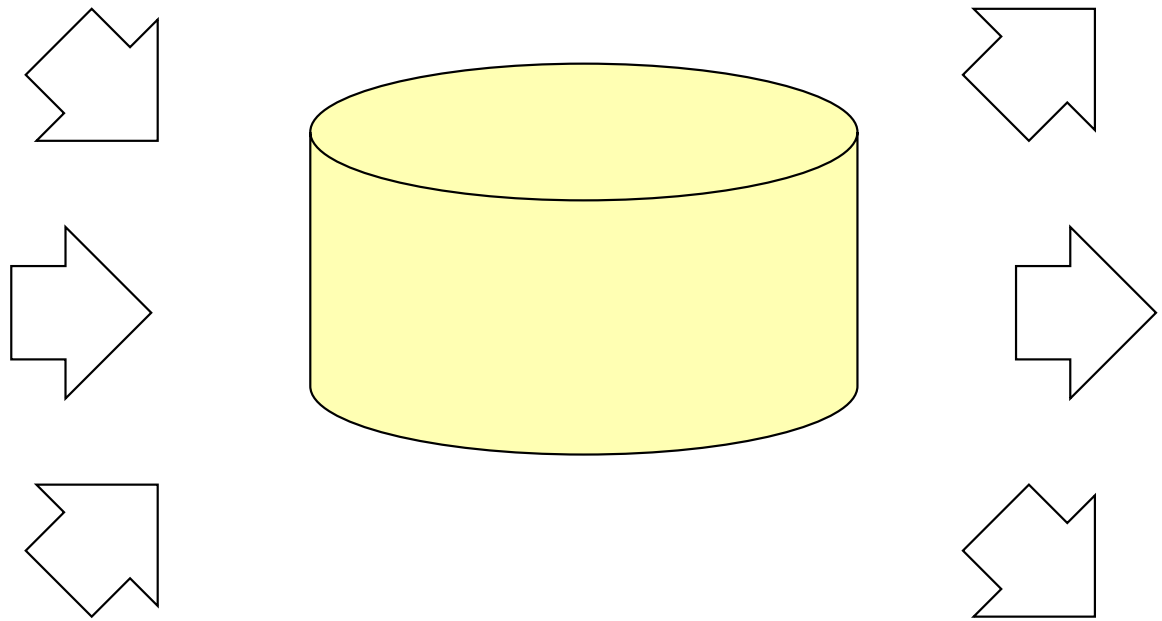
After this class, you are expected to understand

- how longest prefix forwarding in a router works
- the purpose of routing protocols on the Internet
- the differences between inter-domain and intra-domain routing.
- the workings of link-state and distance vector routing
- the principle of Bellman-Ford equation
- how RIP works

``Millions of Routers Work in Concert to Route Packets on the Internet, Based on This One Simple Equation. Amazing."


Application
Transport
Network
Link
Physical


Transport
IP
ICMP
Routing
Protocols
Link



Forwarding Table

destinations	outgoing interface

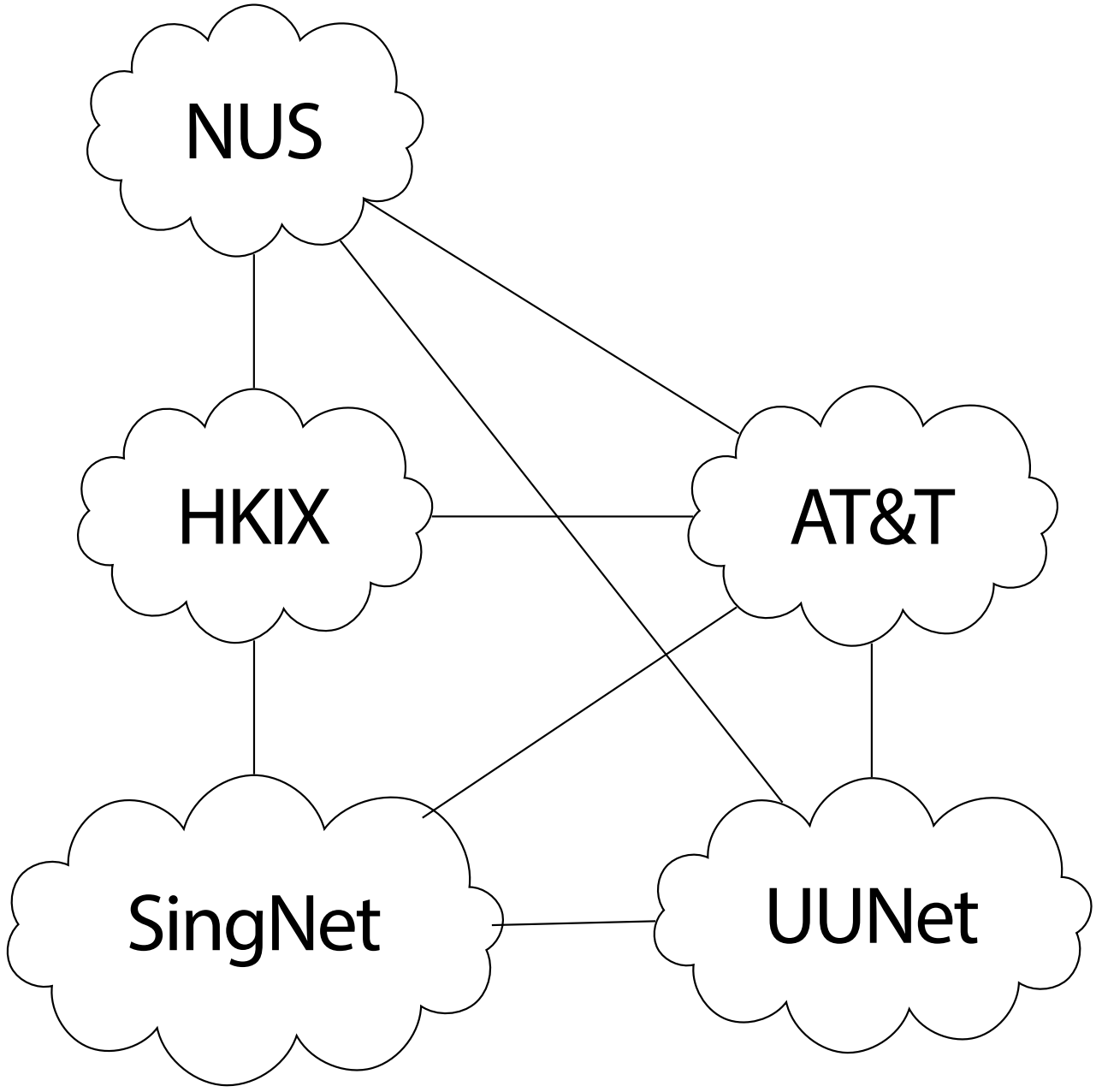
Longest Prefix Matching

address prefix	outgoing interface

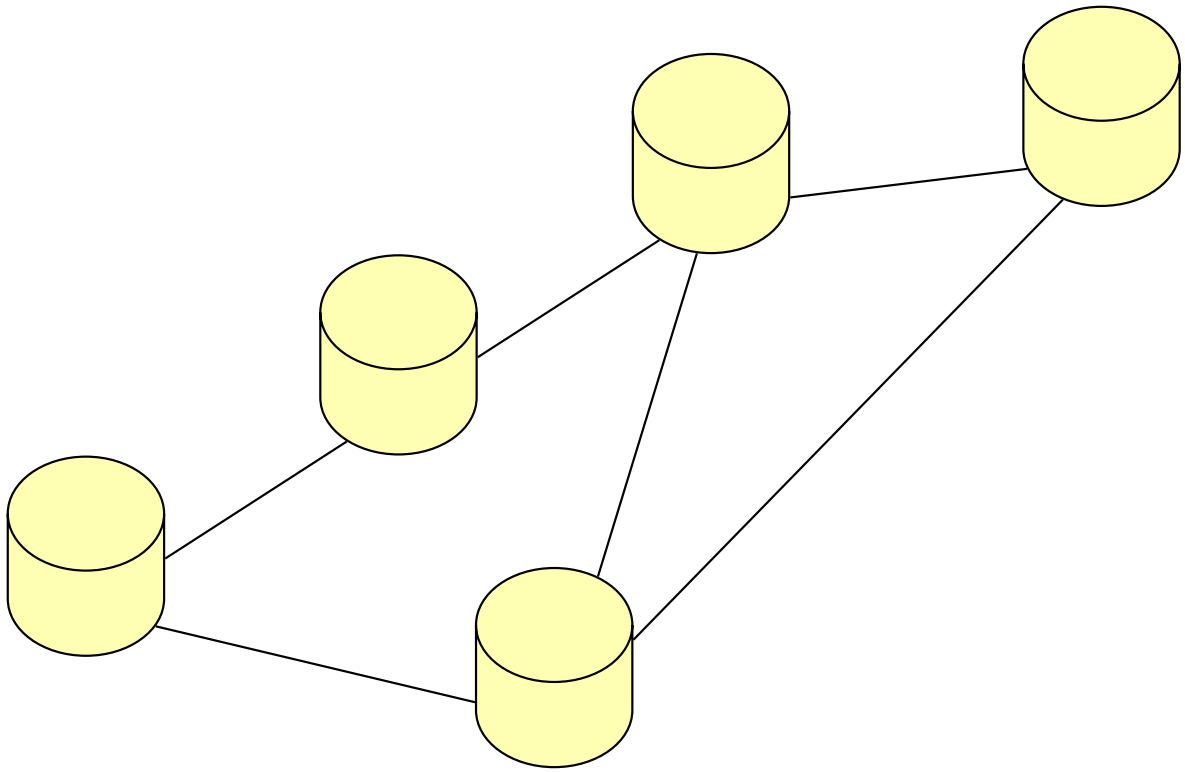
The Internet is a ``network-of-networks", organized into autonomous systems (AS), each is owned by an organization.

Inter-AS routing

Intra-AS routing


NUS
HKIX
AT&T
SingNet
UUNet

Abstract view of intra-domain routing



Routing: finding the **least cost path** in a graph.

Link-State Routing

1. Broadcast link cost to each other
2. Compute least cost path locally

1. Broadcast link cost to each other
2. Compute least cost path locally

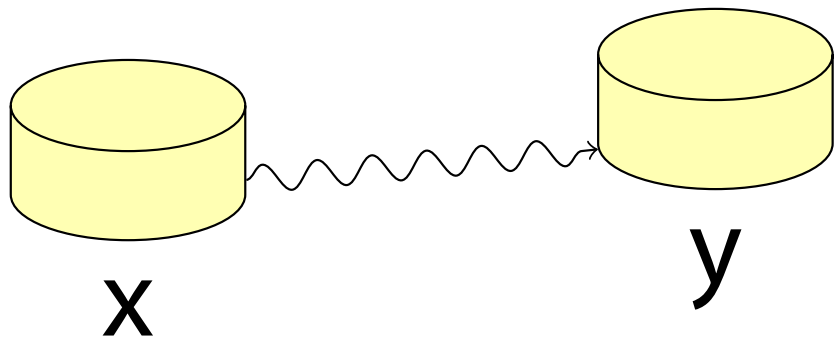
We will not cover Step 2 in CS2105. The algorithm will be covered in CS2010.

Distance Vector Routing

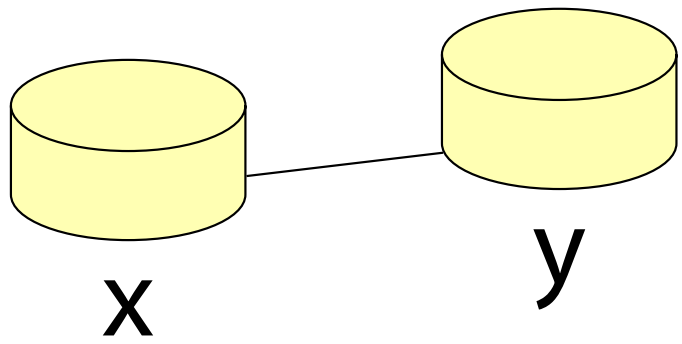
1. Swap local view with neighbors
2. Update own's local view
3. Repeat

decentralized
self-terminating
iterative
asynchronous

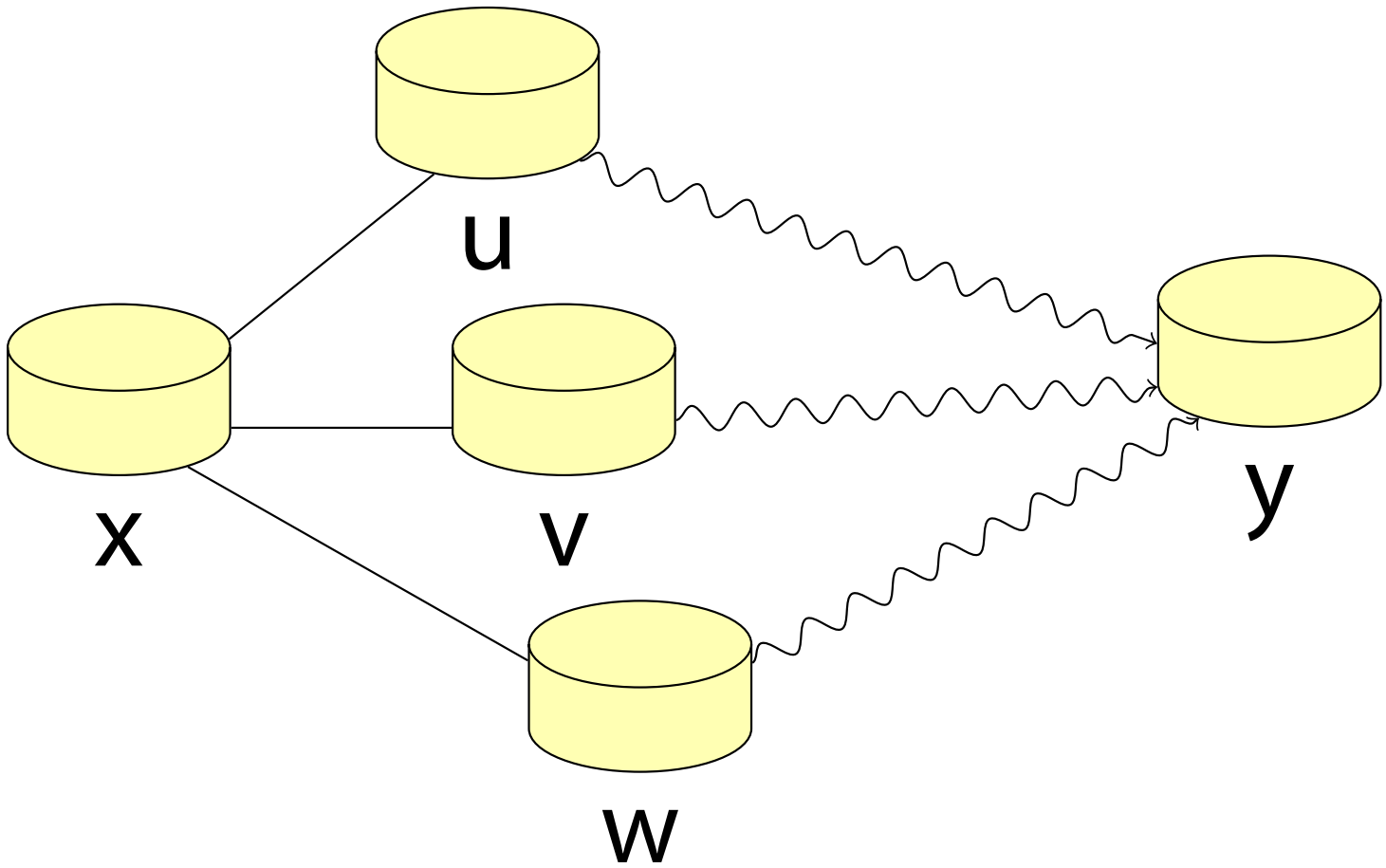
$d_x(y)$: the total cost to reach *y* from *x*'s view



$c(x, y)$: the link cost between x and y.



$$d_x(y) = \min_i\{c(x,i) + d_i(y)\}$$

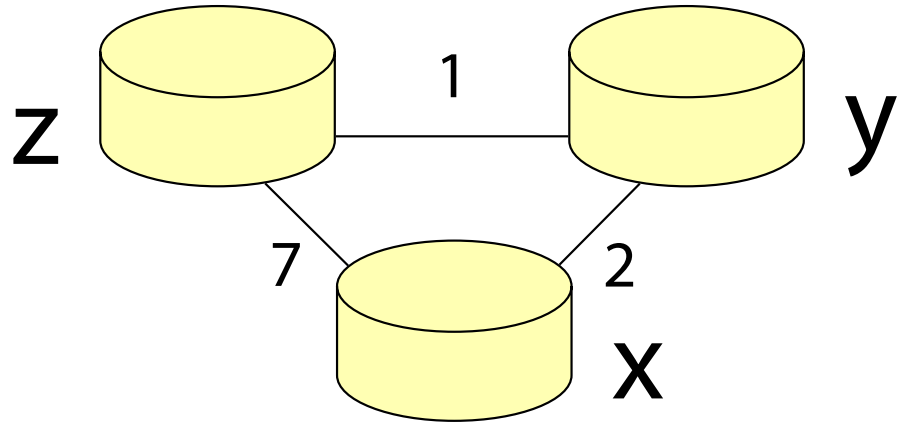


$$d_x(y) = \min_i\{c(x,i) + d_i(y)\}$$

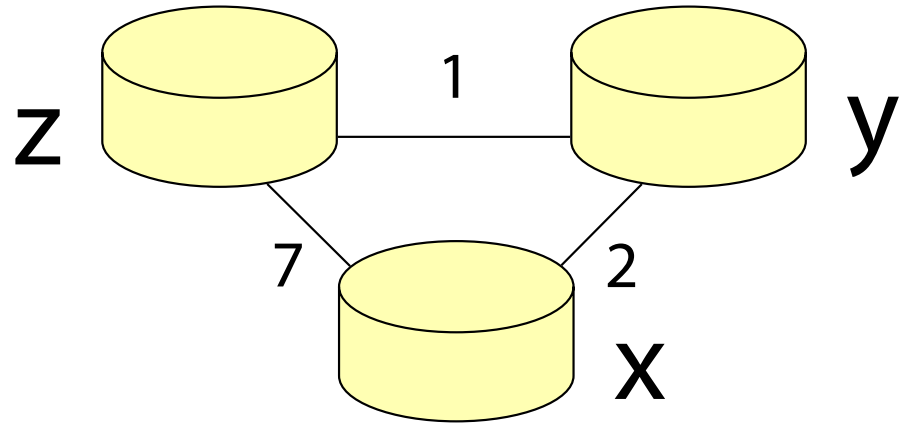
dest	next	cost
v	z	10
w	z	23
x	x	11
y	x	17

distance vector is the ``local view'' exchanged between neighbors.

dest	cost
v	10
w	23
x	11
y	17



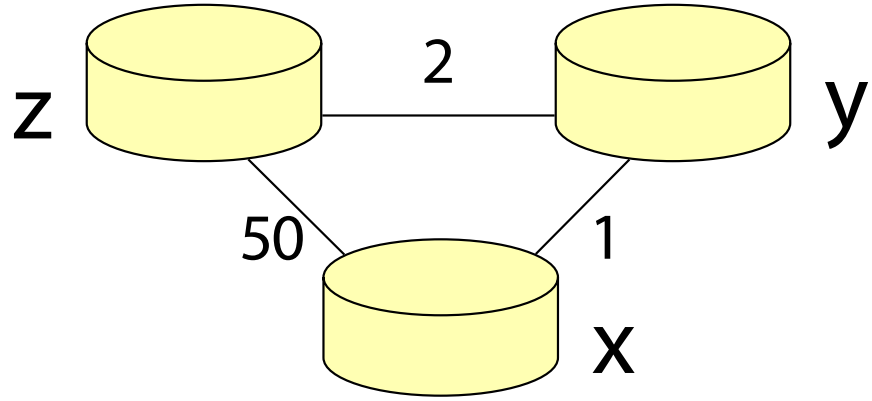
	to *x*	to *y*	to *z*
@*x*			
@*y*			
@*z*			



$$d_x(y) = \min_i\{c(x,i) + d_i(y)\}$$

	to x	to y	to z
@x			

decentralized
self-terminating
iterative
asynchronous



$$d_x(y) = \min_i\{c(x,i) + d_i(y)\}$$

	to x	to y	to z
@x			
@y			
@z			

	to x	to y	to z
@x			
@y			
@z			

``count-to-infinity"

``poisoned reverse"

Without poisoned reverse

for all neighbor *n* **do**
    distance vector $v_n \leftarrow [\,]$
    for all entry *e* in local table **do**
        add (*e*.dest, *e*.cost) to v_n
    end for
    send v_n to *n*
end for

With poisoned reverse

for all neighbor *n* **do**
    distance vector $v_n \leftarrow [\,]$
    for all entry *e* in local table **do**
        if *e*.next is *n* **then**
            add (*e*.dest, ∞) to v_n
        else
            add (*e*.dest, *e*.cost) to v_n
        end if
    end for
    send v_n to *n*
end for

link state vs. distance vector

message overhead

convergence speed

robustness (error)

RIP: Routing Information Protocol

Routing Table

dest	next hop	hop count
v	z	3
w	z	7
x	x	1
y	z	5

Route is **static** and **load insensitive**.

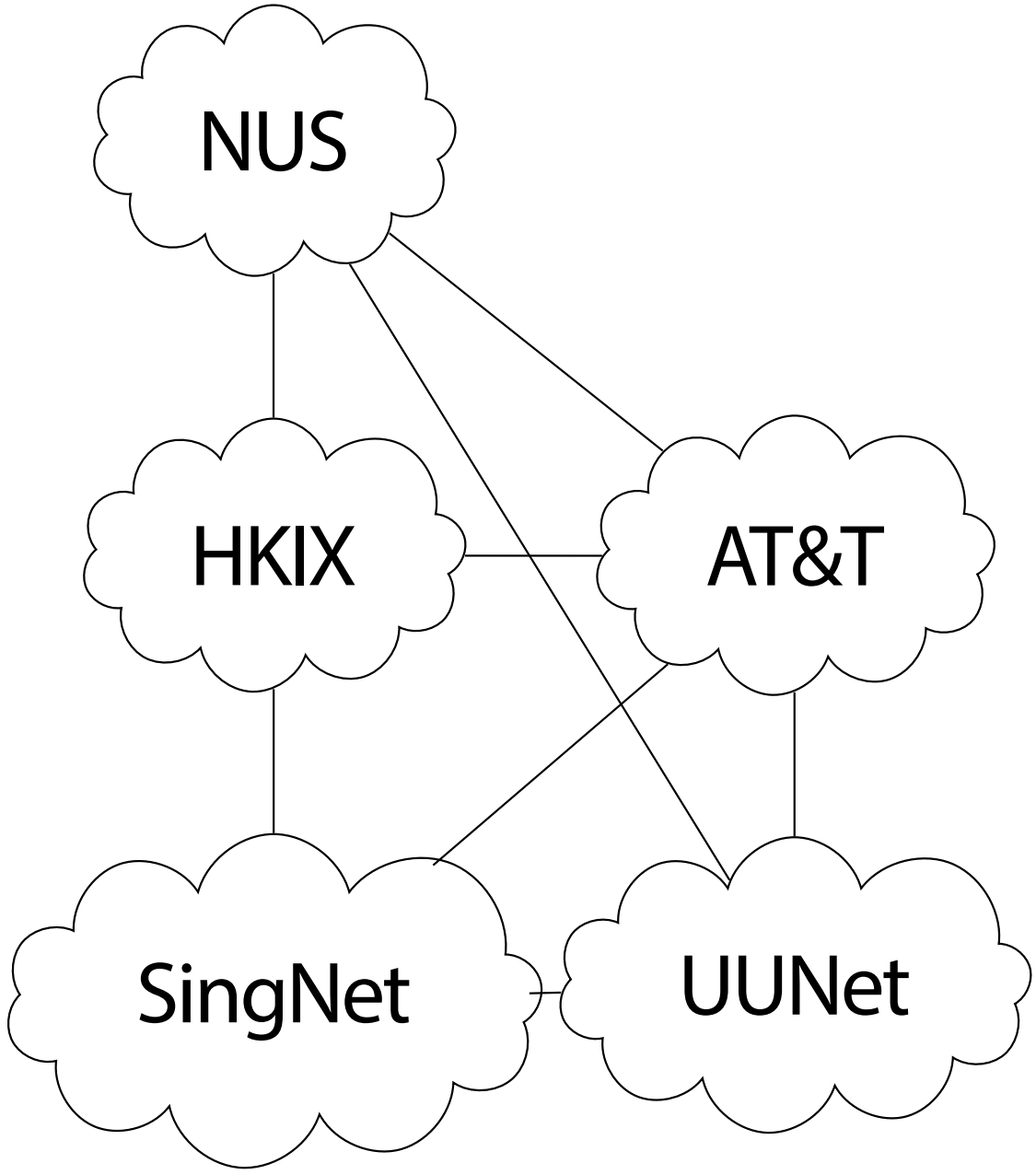
Exchange routing table every 30 seconds over UDP port 520

``Self-repair'': if no update from a neighbor for 3 minutes, assume neighbor has failed

Inter-AS routing
Intra-AS routing

BGP: Border Gateway Protocol

Hot Potato Routing: route to AS whose gateway has the least cost.


NUS
HKIX
AT&T
SingNet
UUNet