

Các giao thức định tuyến

Các giải thuật định tuyến

PGS. Trương Diệu Linh

Bộ môn Truyền thông & Mạng máy tính

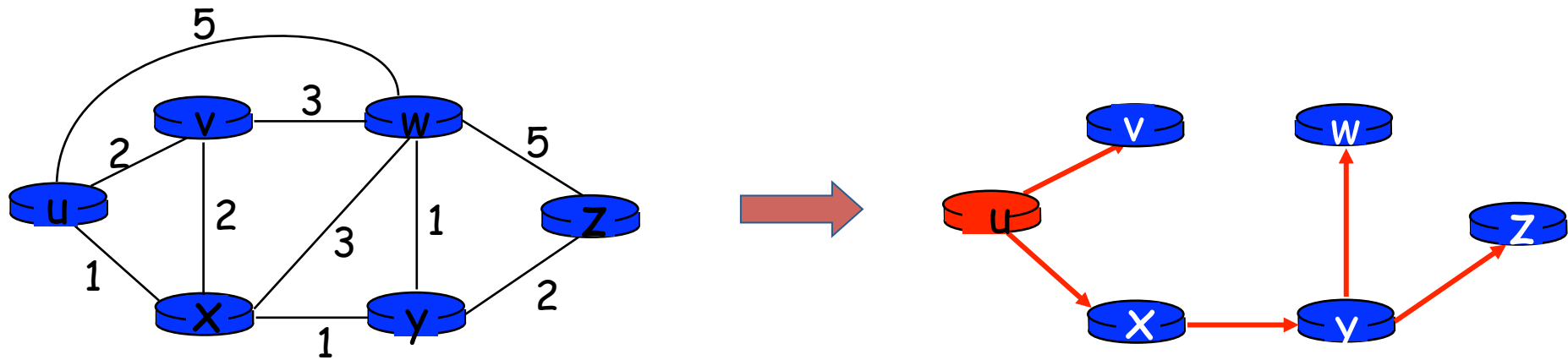
Các giải thuật tìm đường

- ◆ Link-state: Dijkstra
- ◆ Distance vector: Bellman Ford
- ◆ Flooding
- ◆ Giải thuật tìm đường phân cấp
- ◆ Giải thuật tìm hai đường đi phân biệt Suurball
- ◆ Giải thuật Prim-Dijkstra
- ◆ Định tuyến cho các trạm di động
- ◆ Định tuyến trong mạng Ad-hoc

Các giải thuật định tuyến

- ◆ Thuật toán định tuyến/ tìm đường là một bộ phận của tầng mạng có nhiệm vụ quyết định đường ra/ vào cả một gói tin có thể được truyền lên đó,
- ◆ Thuật toán tìm đường đòi hỏi các tính chất sau:
 - ✓ Tính chính xác,
 - ✓ Tính đơn giản,
 - ✓ Khả năng mở rộng
 - ✓ Tính ổn định,
 - ✓ Tính công bằng và tối ưu

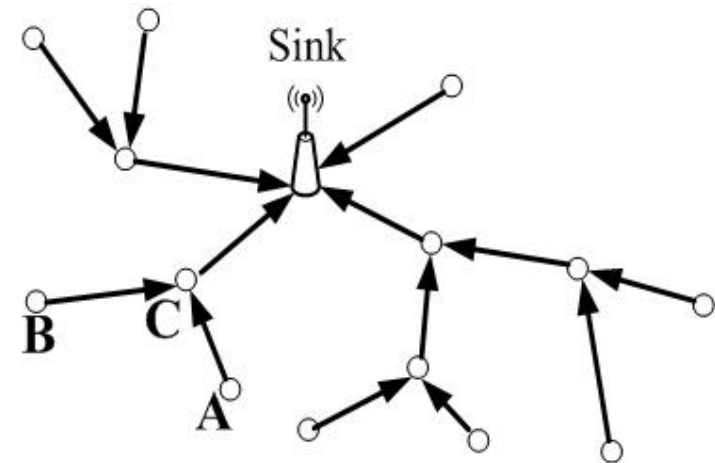
Cây đường đi ngắn nhất - SPT



- ◆ SPT – Shortest Path Tree
- ◆ Các cạnh xuất phát từ nút gốc và tới các lá
- ◆ Đường đi duy nhất từ nút gốc tới nút v, là đường đi ngắn nhất giữa nút gốc và nút v
- ◆ Mỗi nút sẽ có một SPT của riêng nút đó

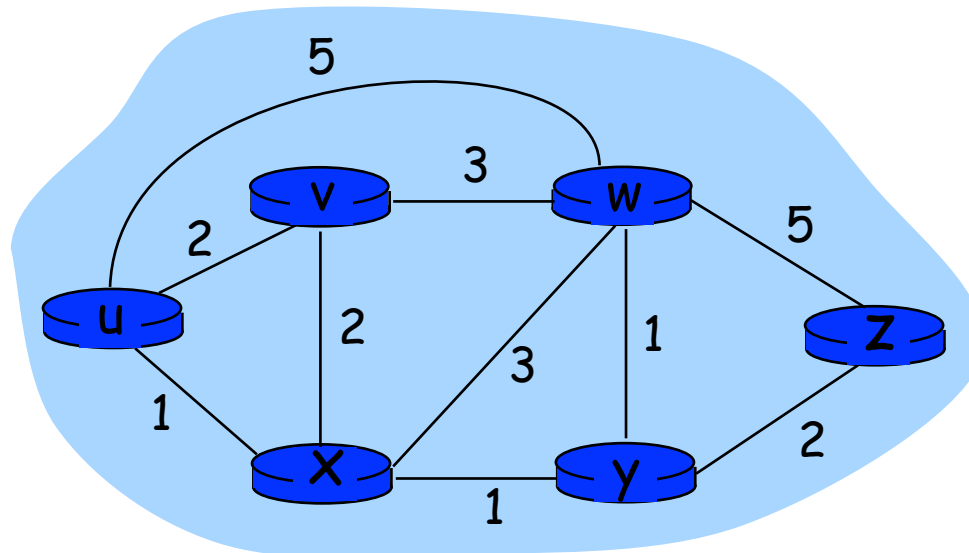
Các giải thuật tìm đường

- ◆ Tìm đường đi từ 1 nguồn đến tất cả các nút khác thường dựa trên cây khung
- ◆ Cây khung là 1 cây có gốc là nguồn đi qua tất cả các đỉnh của một đồ thị
- ◆ Nguyên tắc tối ưu của các giải thuật tìm đường:
 - ✓ Cây khung tối thiểu: tổng trọng số min.
 - ✓ Một cây khung tối thiểu có thể không phải là duy nhất.
 - ✓ Một cây khung tối thiểu không chứa bất kỳ một vòng lặp nào,



Biểu diễn mạng bởi đồ thị

- ◆ Đồ thị với các nút (bộ định tuyến) và các cạnh (liên kết)
- ◆ Chi phí cho việc sử dụng mỗi liên kết $c(x,y)$
 - ✓ Băng thông, độ trễ, chi phí, mức độ tắc nghẽn...
- ◆ Giải thuật chọn đường: Xác định đường đi ngắn nhất giữa hai nút bất kỳ



Các giải thuật tìm đường kiểu link-state: Dijkstra

- ◆ Giải thuật chọn đường đi ngắn nhất Dijkstra (1959):
 - ✓ Thuật toán Dijkstra, mang tên của nhà khoa học máy tính người Hà Lan Edsger Dijkstra, là một thuật toán giải quyết bài toán đường đi ngắn nhất nguồn đơn trong một đồ thị có hướng.
 - ✓ Thuật toán thực hiện tìm đường đi từ một đỉnh đến tất cả các đỉnh còn lại của đồ thị có trọng số không âm.
 - ✓ Thuật toán Dijkstra bình thường sẽ có độ phức tạp là trong đó m là số cạnh, n là số đỉnh của đồ thị đang xét. $O(n^2 + m)$
 - ✓ Để minh họa các giải thuật tìm đường, thông thường người ta ký hiệu N là số nodes trong mạng, i và j là nhãn của các node trong mạng.

Dijkstra

- **Ký hiệu:**
- $G = (V, E)$: Đồ thị với tập đỉnh V và tập cạnh E
- $c(x, y)$: chi phí của liên kết x tới y ; $= \infty$ nếu không phải 2 nút kề nhau
- $d(v)$: chi phí hiện thời của đường đi từ nút nguồn tới nút đích. v
- $p(v)$: nút ngay trước nút v trên đường đi từ nguồn tới đích
- T : Tập các nút mà đường đi ngắn nhất đã được xác định

Dijkstra

◆ Init():

Với mỗi nút v , $d[v] = \infty$, $p[v] = \text{NIL}$

$d[s] = 0$

◆ Update(u, v), trong đó (u, v) là một cạnh nào đó của G

if $d[v] > d[u] + c(u, v)$ then

$d[v] = d[u] + c(u, v)$

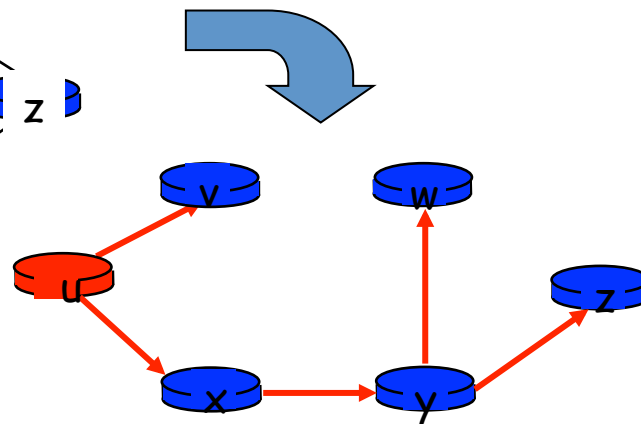
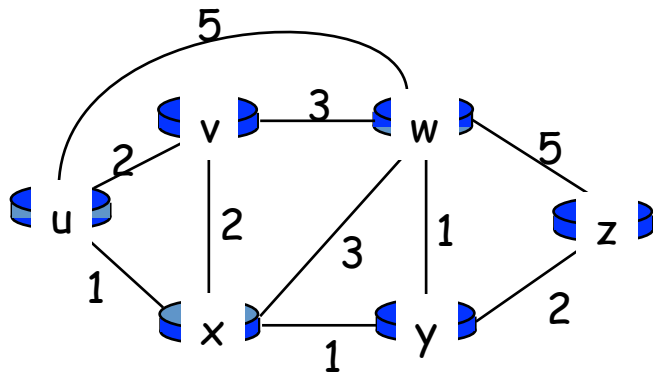
$p[v] = u$

Dijkstra

1. **Init()** ;
2. $T = \Phi$;
3. **Repeat**
4. $u: u \in T \mid d(u)$ là bé nhất ;
5. $T = T \cup \{u\}$;
6. for all $v \in \text{neighbor}(u)$ và $v \notin T$
7. $\text{update}(u, v)$;
8. **Until** $T = V$

Dijkstra

Step	T	$d(v), p(v)$	$d(w), p(w)$	$d(x), p(x)$	$d(y), p(y)$	$d(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



SPT của u:

Bảng chọn đường của u:

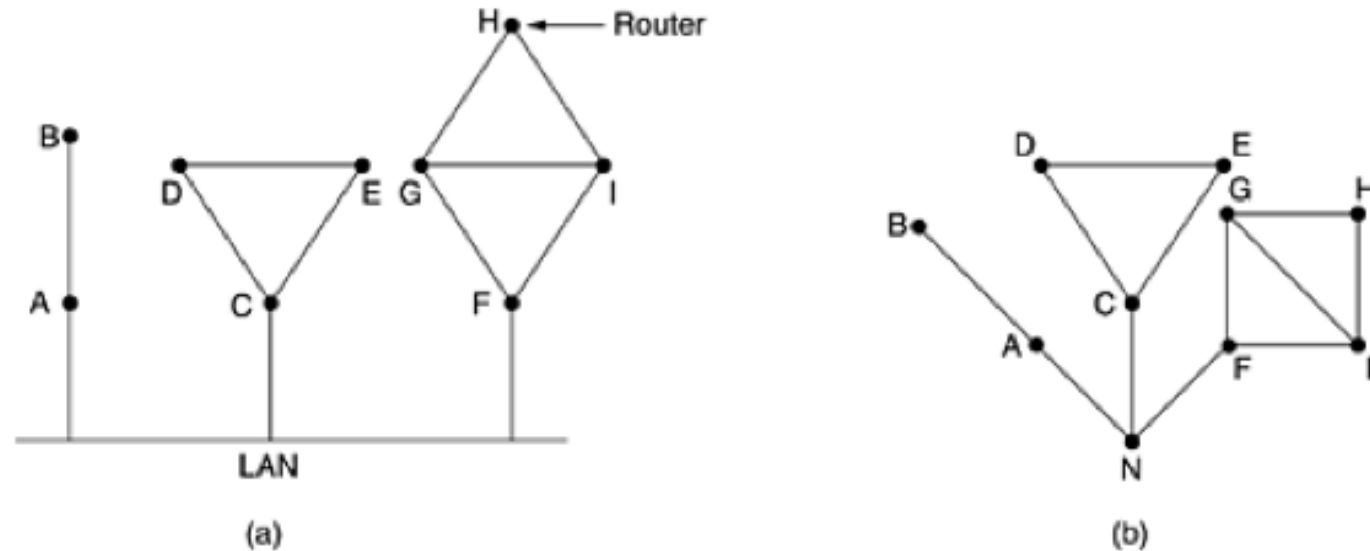
destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Giải thuật tìm đường link-state

- ◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):
 1. Khám phá các láng giềng và học các địa chỉ mạng của chúng
 2. Đo độ trễ (delay), hay giá (cost) tới các láng giềng
 3. Xây dựng một gói tin báo cho các trạng thái/thông tin vừa học
 4. Gửi gói tin cập nhật đến tất cả các routers khác
 5. Tính đường dẫn ngắn nhất cho từng routers (sử dụng giải thuật Dijkstra để tìm đường ngắn nhất tới từng routers)

Giải thuật tìm đường link-state

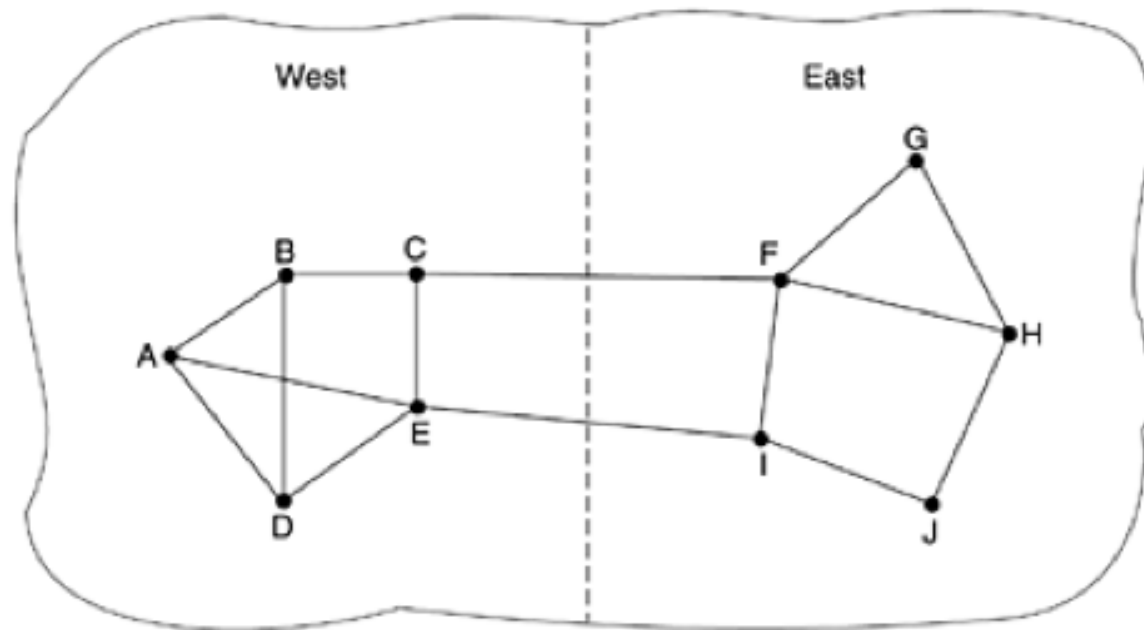
- ◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):
 1. Khám phá các láng giềng và học các địa chỉ mạng của chúng
 - ✓ Khám phá các routers láng giềng bằng cách gửi 1 gói tin HELLO trên mỗi đường dẫn,
 - ✓ Khi 2 hay nhiều routers kết nối bởi một LAN, tình huống sẽ phức tạp



Hình 24: 9 routers nối qua 1 mạng LAN, mạng LAN đc xem như một nút ảo.

Giải thuật tìm đường link-state

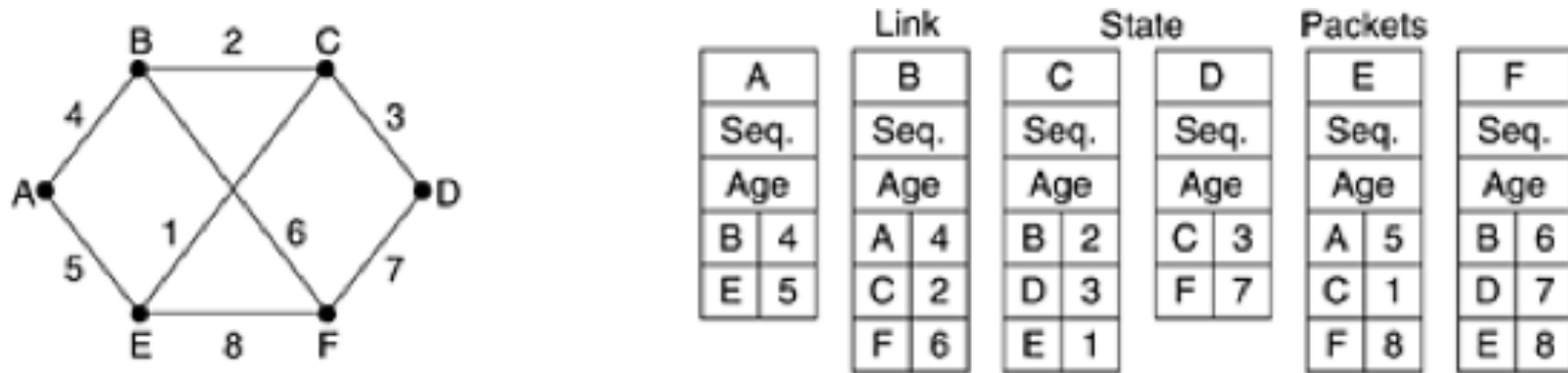
- ◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):
 2. Đo độ trễ (delay), hay giá (cost) tới các láng giềng, các routers phải có sự ước lượng về các đường dẫn tới các routers láng giềng để làm trọng số cho giải thuật định tuyến.



Giải thuật tìm đường link-state

- ◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):

3. Xây dựng một gói tin báo cho các trạng thái/ thông tin vừa học: Gói tin bắt đầu với định danh của máy gửi, theo sau là thứ tự trạng thái, tuổi (age) và một danh sách các láng giềng. Thông thường các gói tin trạng thái được xây dựng một cách định kỳ.



Hình 26: Ví dụ về các gói tin trạng thái liên kết cho subnet

Giải thuật tìm đường link-state

◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):

4. Gửi gói tin cập nhật đến tất cả các routers khác:

- ✓ Sử dụng thuật toán ngập lụt (flooding) để gửi các gói tin trạng thái,
- ✓ Các gói tin chứa thông tin về tuổi (age) để tránh trùng lặp và cập nhật thông tin. Khi bộ đếm tuổi quay về zero, thông tin về routers đấy sẽ bị hủy.
- ✓ Trường tuổi cũng giảm theo từng routers trong quá trình ngập lụt để đảm bảo không có gói tin nào có thể tồn tại vô hạn,
- ✓ Các gói tin trạng thái thường được lưu vào bộ nhớ đệm để xử lý tuần tự, nếu có trùng lặp sẽ bị loại bỏ.

Source	Seq.	Age	Send flags			ACK flags			Data
			A	C	F	A	C	F	
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Hình 27: Ví dụ về buffer đệm lưu trữ gói tin trạng thái của router B

Giải thuật tìm đường link-state

◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols):

5. Tính đường dẫn ngắn nhất cho từng routers:

- ✓ Sau khi các routers có đầy đủ thông tin trạng thái các đường dẫn sẽ sử dụng thuật toán Dijkstra để tính toán/xây dựng đường dẫn ngắn nhất cho mọi nơi đến có thể,
- ✓ Chọn đường trạng thái liên kết (link-state) được dùng nhiều trong các mạng hiện nay,
 - ✓ Các giao thức chọn đường trạng thái liên kết phổ biến là OSPF (Open Shortest Path First) và IS-IS (Intermediate System-Intermediate System).

Giải thuật tìm đường link-state

- ◆ Giải thuật tìm đường trạng thái liên kết (link-state routing protocols)
 - ✓ Bộ định tuyến dùng nhiều bộ nhớ và thực thi nhiều hơn so với các giao thức định tuyến theo vectơ khoảng cách.
 - ✓ Lý do cần thiết phải lưu trữ thông tin của tất cả các láng giềng, cơ sở dữ liệu mạng đến từ các nơi khác và việc thực thi các thuật toán định tuyến trạng thái.
 - ✓ Các nhu cầu về băng thông cần phải tiêu tốn để khởi động sự phát tán gói trạng thái.

Bellman-Ford

Phương trình Bellman-Ford (quy hoạch động)

Định nghĩa

$d_x(y) :=$ chi phí của đường đi ngắn nhất
từ x tới y

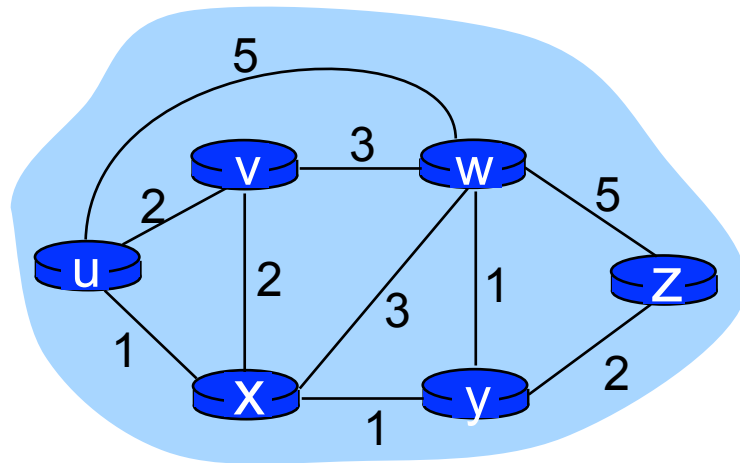
Ta có

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cho tất cả các v là hàng xóm của x

Bellman-Ford

Dễ thấy, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$



B-F eq. cho ta biết:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

Nút nào làm giá trị trên nhỏ nhất → Lựa chọn là nút kế tiếp trong bảng chọn đường

Giải thuật tìm đường vector khoảng cách

◆ Giải thuật tìm đường vector khoảng cách (distance-vector):

- ✓ Mỗi router duy trì một bảng vector khoảng cách cho cụ thể tốt nhất tới từng đích và đường truyền dùng để tới đích đó,
- ✓ Các bảng vector khoảng cách được cập nhật thông tin nhờ trao đổi định kỳ với các láng giềng của nó,
- ✓ Sử dụng thuật toán Bellman-Ford Fulkerson,
- ✓ Là thuật toán chọn đường gốc cho mạng ARPANET và được dùng rộng rãi trong định tuyến IP trong giao thức định tuyến RIP (Routing Information Protocol), IDP của Novell.
- ✓ Được sử dụng đến năm 1979 sau đó được thay thế bởi các giải thuật chọn đường trạng thái liên kết do độ trễ lớn, thời gian hội tụ lâu do đòi hỏi cần phải trao đổi các thông điệp định tuyến lớn.

Giải thuật tìm đường dạng distance-vector

ý tưởng cơ bản:

- ◆ DV: Vector khoảng cách, tạm coi là đường đi ngắn nhất của từ một nút tới những nút khác
- ◆ Mỗi nút định kỳ gửi DV của nó tới các nút bên cạnh
- ◆ Khi nút x nhận được 1 DV, nó sẽ cập nhật DV của nó qua pt Bellman-ford
- ◆ Với một số điều kiện, ước lượng $D_x(y)$ sẽ hội tụ dần đến giá trị nhỏ nhất $d_x(y)$

Mỗi nút:

Chờ (Thay đổi trong DV của nút bên cạnh)

Tính lại ước lượng DV

Nếu DV thay đổi, *Báo* cho nút bên cạnh

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

nút x

		chi phí tới		
		x	y	z
từ	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		chi phí tới		
		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	7	1	0

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

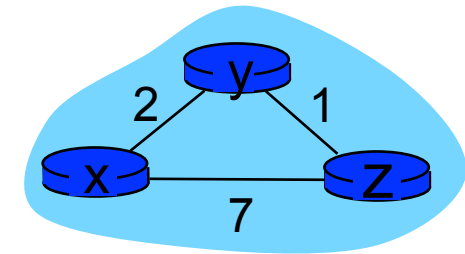
$$= \min\{2+1, 7+0\} = 3$$

nút y

		chi phí tới		
		x	y	z
từ	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

nút z

		chi phí tới		
		x	y	z
từ	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0



thờigian

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \\ = \min\{2+1, 7+0\} = 3$$

nút x

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

nút y

chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

nút z

chi phí tới

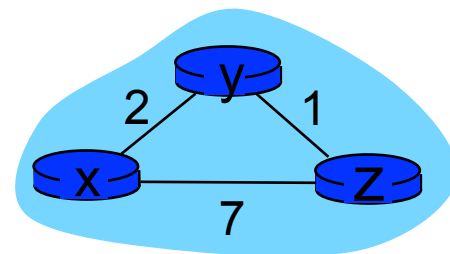
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



thời gian

So sánh link-state và distance vector

Thông điệp trao đổi

- LS: n nút, E cạnh, $O(nE)$ thông điệp
- DV: Chỉ trao đổi giữa các hàng xóm
 - Thời gian hội tụ thay đổi

Tốc độ hội tụ

- LS: Thuật toán: $O(n^2)$ cần $O(nE)$ thông điệp
- DV: Thay đổi

Sự chắc chắn: Giải sử một router hoạt động sai

LS:

- nút gửi các chi phí sai
- Mỗi nút tính riêng bảng chọn đường -> có vẻ chắc chắn hơn

DV:

- DV có thể bị gửi sai
- Mỗi nút tính toán dựa trên các nút khác
 - Lỗi bị lan truyền trong mạng

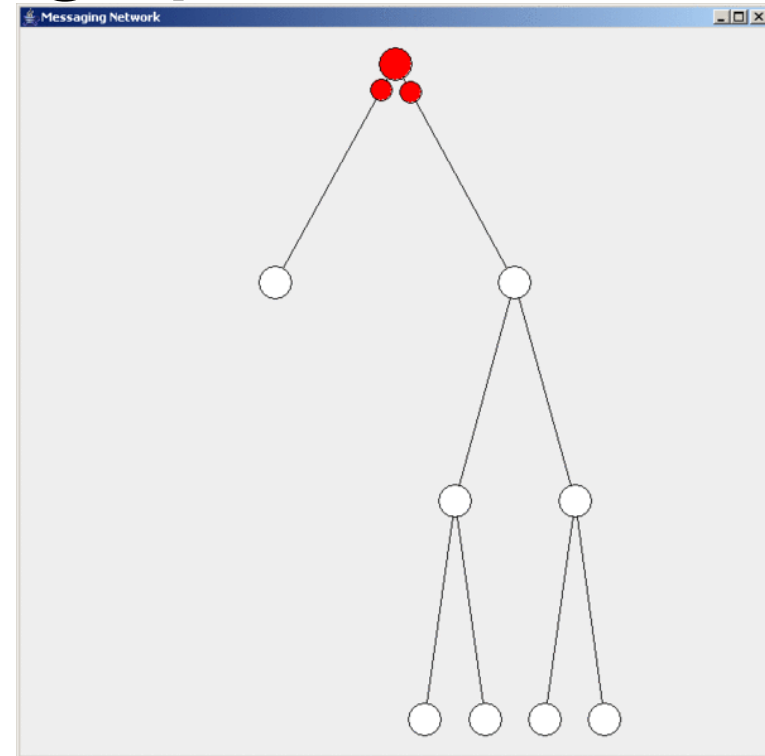
Các giải thuật tìm đường

◆ Tổng kết về các giao thức định tuyến phổ dụng trên mạng IP:

Đặc trưng	RIPv1	RIPv2	IRGP	EIGRP	OSPF
Distance-vector	X	X	X	X	
Link-state					X
Tự động tóm tắt định tuyến	X	X	X	X	
Hỗ trợ VLSM		X		X	X
3 rd party compatible	X	X			X
Thích hợp	Nhỏ	Nhỏ	Vừa	Lớn	Lớn
Thời gian hội tụ	Chậm	Chậm	Chậm	Nhanh	Nhanh
Giá trị định tuyến	Hop count	Hop count	BW+D	BW+D	10E8/BW
Giới hạn hop	15	15	100	100	
Cân bằng tải cùng giá trị đt	X	X	X	X	X
Cân bằng tải ko cùng giá trị đt			X	X	
Thuật toán	Bellman-Ford	Bellman-Ford	Bellman-Ford	Dual	Dijkstra

Giải thuật ngập lụt

- ◆ Giải thuật ngập lụt (flooding):
 - ◆ Mỗi nút vừa là nút nhận vừa là nút gửi.
 - ◆ Mỗi nút gửi gói tin nó nhận được ra tất cả các nút kề trừ nút mà từ đó nó đã nhận gói tin.
 - ◆ Giải thuật có thể có thêm các cơ chế chống lặp, chống vận chuyển vòng tròn, cơ chế hết hạn...
- ◆ Giải thuật này thường được sử dụng trên các mạng ad hoc, sensor networks.
- ◆ selective flooding algorithm:
 - ◆ chỉ gửi gói tin trên các đường đi theo hướng của thiết bị đích



Giải thuật ngập lụt

- Ngập lụt có điều khiển
 - SNCF(Sequence-number-controlled flooding)
 - Nút gửi đưa địa chỉ của nó và số hiệu gói vào gói tin broadcast.
 - Mỗi nút lưu địa chỉ và số hiệu của gói tin nó đã gửi đi.
 - Nếu nút gặp lại gói tin nó đã gửi, nó sẽ bỏ gói tin mà không chuyển tiếp đi nữa.

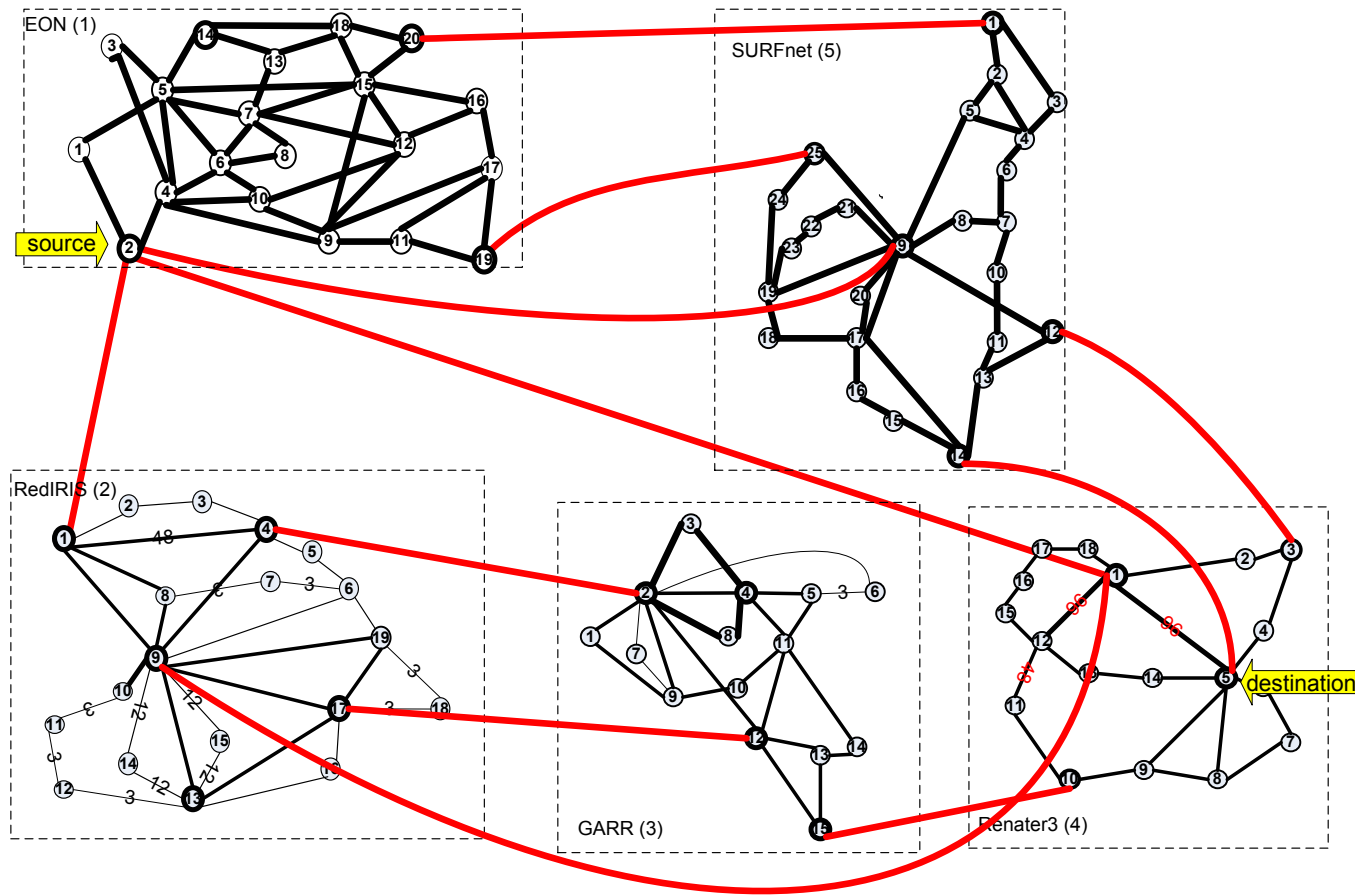
Giai thuật tìm đường phân cấp

- ◆ Tìm đường trong mạng đa miền
 - ✓ Phân cấp: mức inter-domain, intra-domain
 - ✓ Ảo hóa các domain:
 - Ảo hóa kiểu nút:
 - Mỗi miền được ảo hóa thành 1 nút
 - Ảo hóa hình sao
 - Một số nút của miền gọi là nút biên, có các kết nối inter-domain đến các miền khác
 - Mỗi miền ảo hóa thành 1 hình sao, có 1 nút ảo trung tâm nối hình sao đến các nút biên.
 - Ảo hóa hình lưới
 - Một số nút của miền gọi là nút biên, có các kết nối inter-domain đến các miền khác
 - Mỗi miền được ảo hóa thành một đồ thị gồm các nút biên được nối nhau được bởi các liên kết ảo.

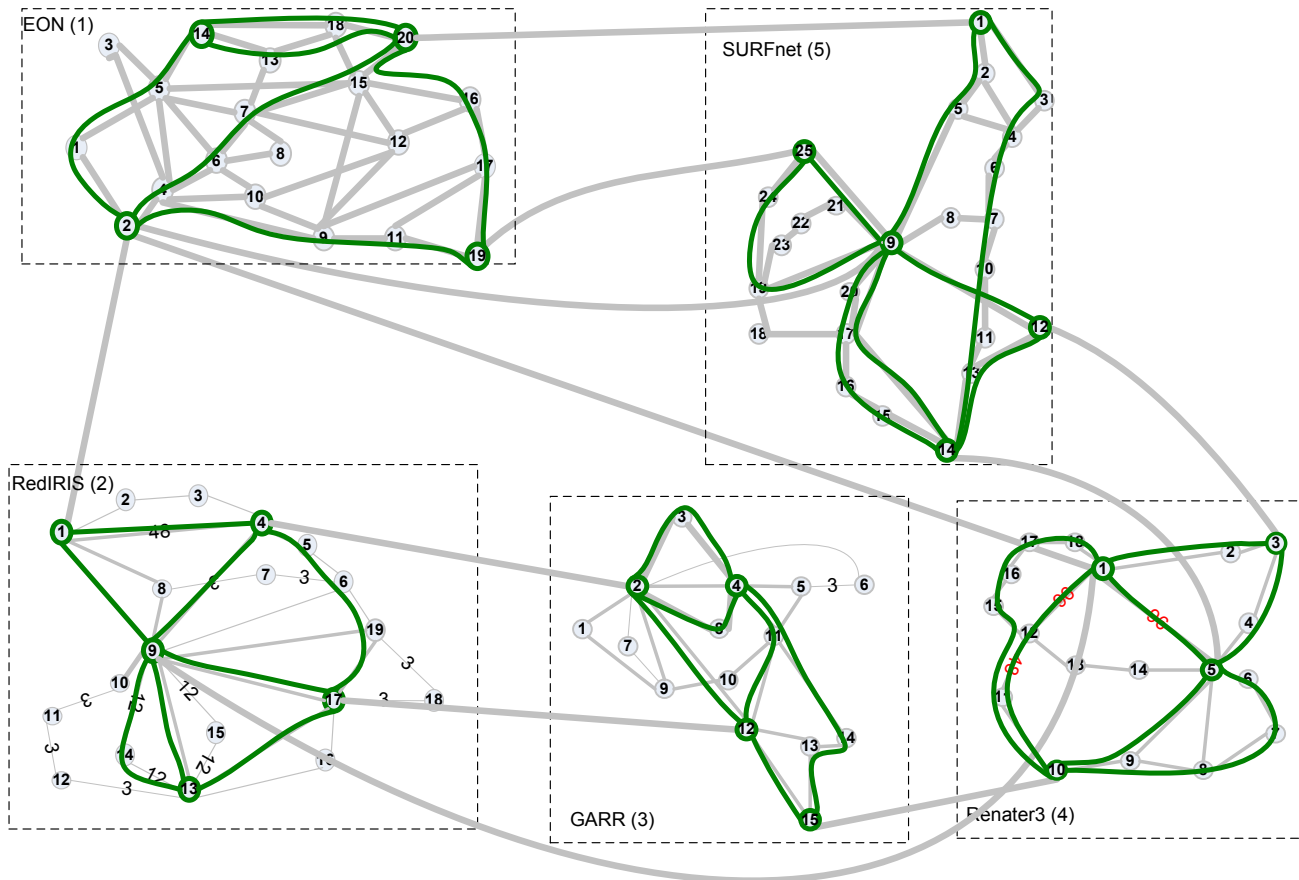
Giải thuật tìm đường phân cấp

- ✓ Mạng mức liên miền trở thành mạng của các miền đã được ảo hóa và các liên kết liên miền
- ✓ Định tuyến ở mức liên miền
- ✓ Định tiến ở mức nội miền.

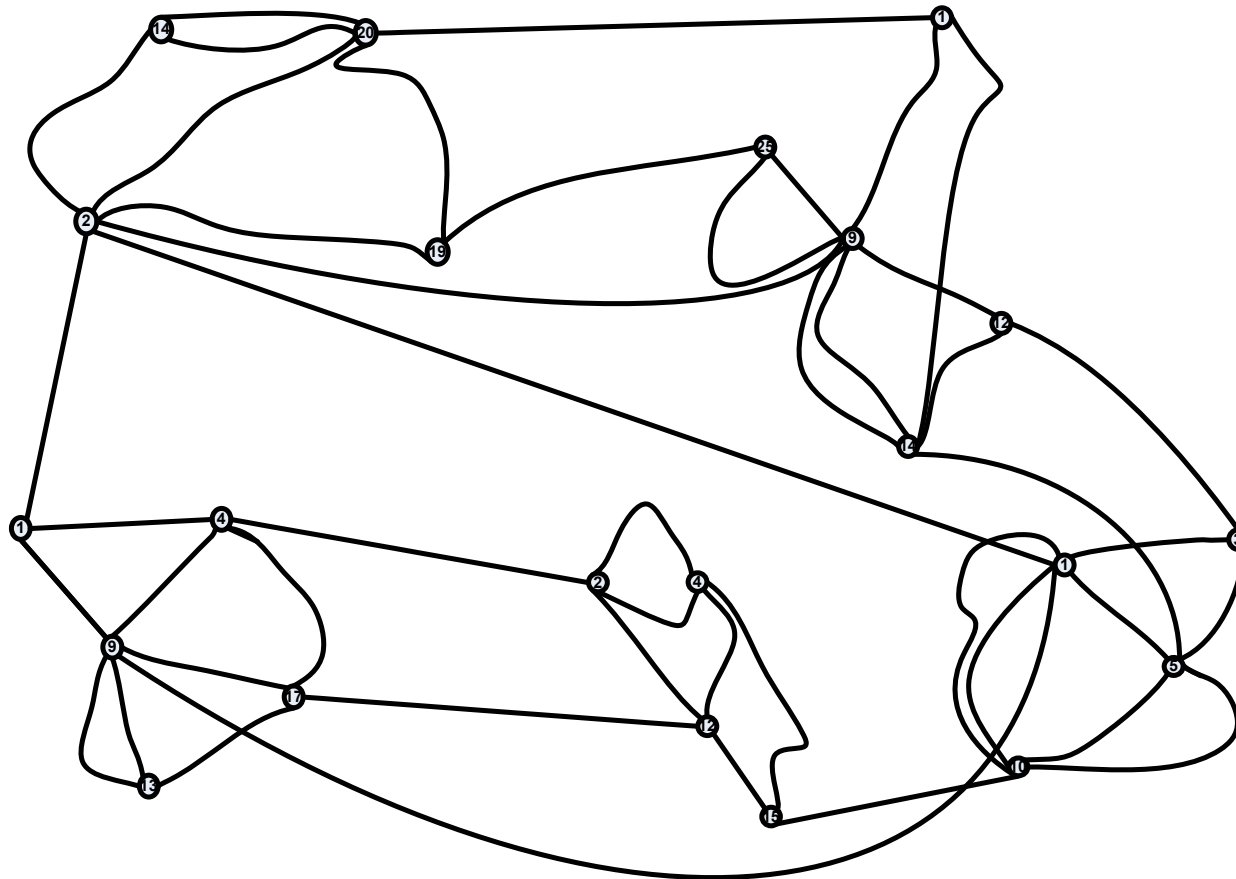
Giải thuật tìm đường phân cấp



Giải thuật tìm đường phân cấp



Giải thuật tìm đường phân cấp



Giải thuật tìm đường phân cấp

◆ Path vector

- ✓ Sử dụng để định tuyến trong các mạng đa miền
- ✓ Mỗi miền được ảo hóa thành một nút mạng gọi là speaker
- ✓ Nút mạng ảo này trao đổi thông tin định tuyến với các nút mạng khác:
 - thông tin về đường đi
 - Không trao đổi thông tin về đơn vị định tuyến (trọng số)
- ✓ Tương tự giải thuật loại distance vector

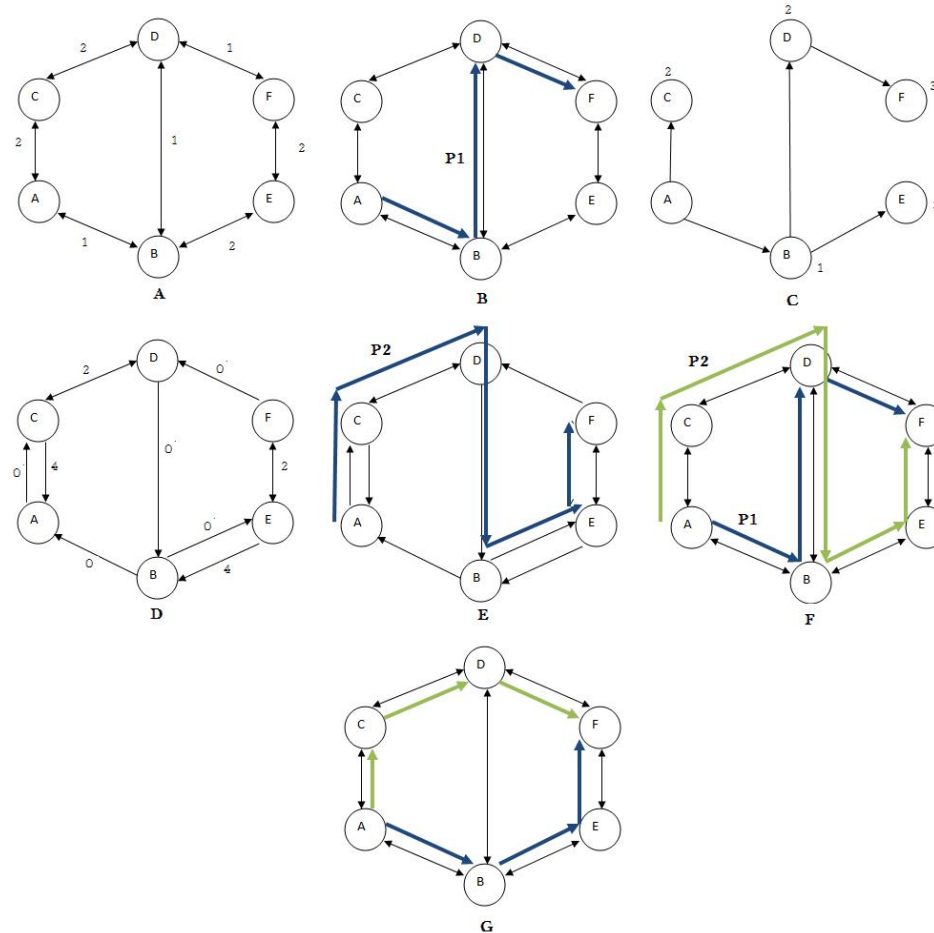
◆ VD: BGP

Giải thuật tìm 2 đường đi phân biệt

- ◆ Giải thuật tìm 2 đường đi cạnh phân biệt
 - ✓ Với một cặp nguồn, đích tìm 2 đường đi không có cạnh chung
 - ✓ Thuật toán dùng cho định tuyến có dự phòng
- ◆ Giải thuật tối ưu Suurballe trong đồ thị có hướng với trọng số không âm
 - ✓ Đồ thị G , trọng số các cạnh $w(u,v)$. Nguồn s . Đích t
 - ✓ Tìm cây đường đi ngắn nhất T từ đỉnh s bằng Dijkstra.
 - ✓ P_1 là đường đi ngắn nhất từ s đến t .
 - ✓ Điều chỉnh trọng số các cạnh của G
 - $w'(u,v) = w(u,v) - d(s,v) + d(s,u)$.
 - → Mọi cạnh của cây T đều có trọng số 0,
 - → Các cạnh không thuộc cây T có trọng số không âm
 - ✓ Tạo đồ thị G_t từ G bằng cách bỏ đi các cạnh hướng vào s và đảo ngược chiều các cạnh có trọng số 0 trên đường đi P_1 .
- ◆ Tìm đường đi ngắn nhất P_2 trên G_t bằng Dijkstra's.
- ◆ Bỏ các cạnh ngược với chiều của P_2 trên cả 2 đường đi.
- ◆ Các cạnh còn lại của P_1 và P_2 tạo thành 2 đường đi không chung cạnh

Giải thuật tìm 2 đường đi phân biệt

- A: Đồ thị G .
- B: Tính đường đi ngắn nhất
- C: Cây T
- D: Điều chỉnh trọng số
- E: Tính P_2 trên G_t .
- F: thu được P_1, P_2 .
- G: 2 đường đi không chung cạnh

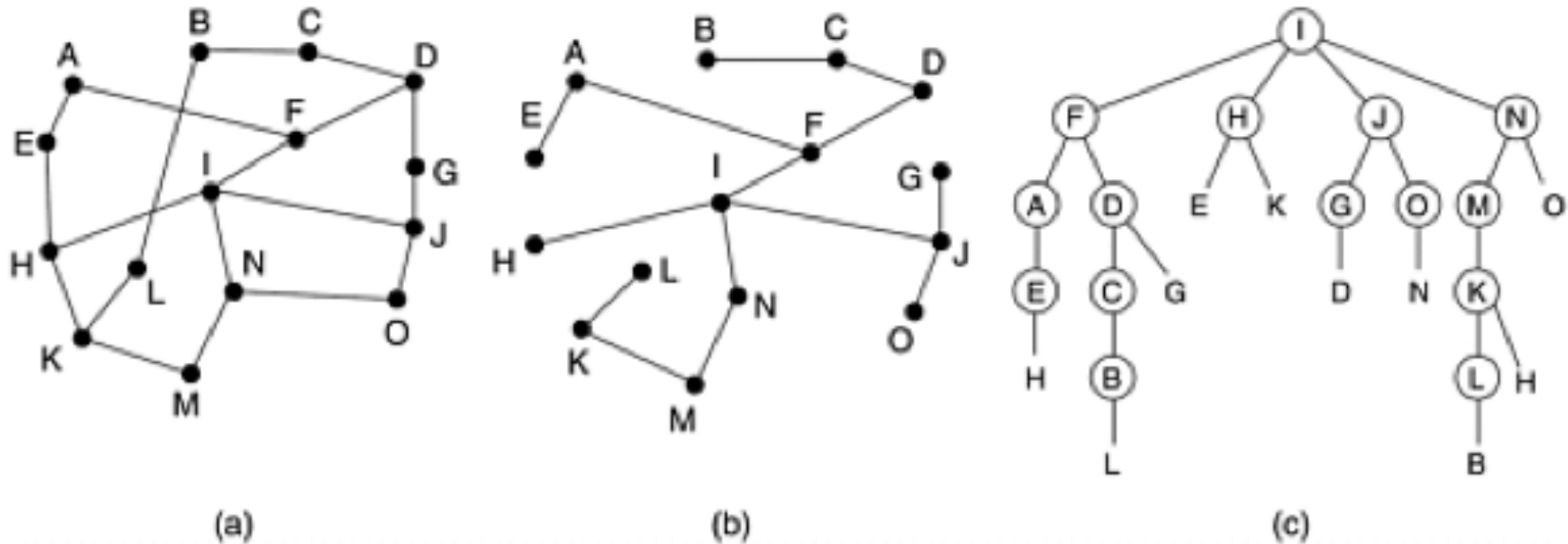


Các giải thuật tìm đường

- ◆ Giải thuật chọn đường quảng bá (broadcast routing):
 - ✓ Trong một số ứng dụng, host cần gửi một số thông tin tới tất cả các host khác trong một subnet hay một liên mạng,
 - ✓ Một số cách thức để thực hiện broadcast routing:
 - Flooding:
 - thường sinh nhiều gói tin thừa và gây lãng phí băng thông,
 - Multi-destination routing:
 - địa chỉ gói tin gửi định nghĩa một tập các địa chỉ gốc mà ánh xạ tới một nhóm hay một nhóm các mạng đích.
 - Spanning tree:
 - sử dụng cây khung nhỏ nhất,
 - khá hiệu quả do không chứa vòng lặp
 - nhược điểm là mỗi router còn chứa thông tin về cây khung nhỏ nhất này.
 - Reserve path forwarding:
 - các router trung chuyển xác định đã nhận gói tin từ cổng nào của router.
 - Nếu đó là **cổng cho đường đi ngắn nhất từ router ngược lại phía nguồn** thì gói tin sẽ được lặp lại trên tất cả các cổng khác. Nếu không gói tin sẽ bị hủy đi.
 - Điều này đảm bảo gói tin được phát tán theo đường tốt nhất.

Các giải thuật tìm đường

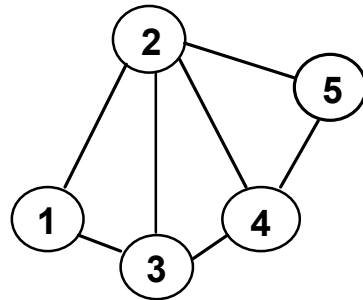
◆ Giải thuật chọn đường quảng bá (broadcast routing):



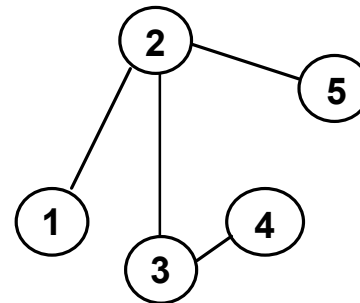
Hình 29: Ví dụ về chọn đường quảng bá với thuật toán spanning tree và thuật toán reserve path forwarding

Cây khung (Spanning tree)

- Cây khung của một đồ thị là cây đi qua tất cả các đỉnh của đồ thị



Graph G



Spanning tree of G

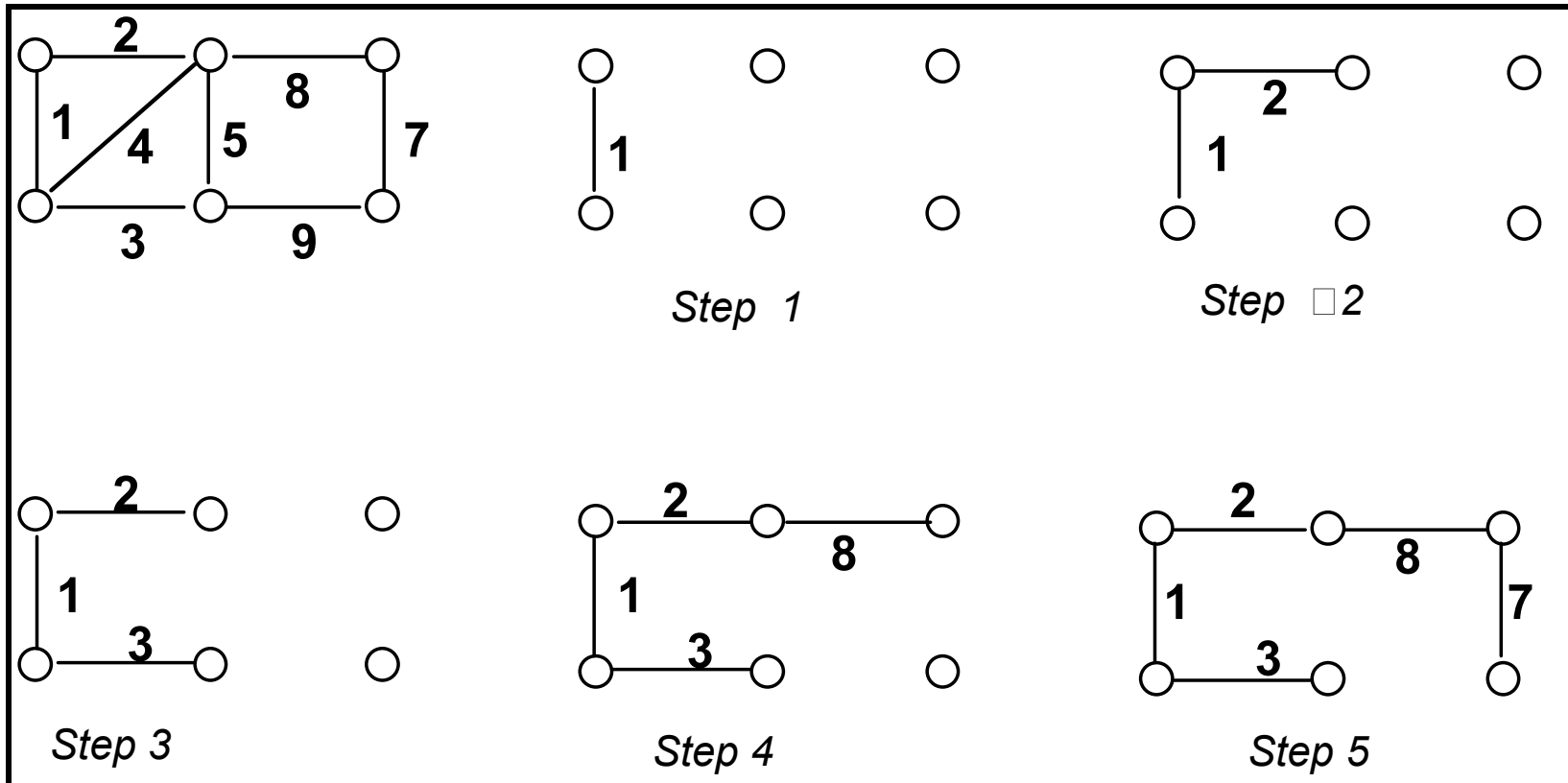
Giải thuật xây dựng cây khung

- Cho đồ thị $G=(N, A)$
 - 1) Chọn một nút bất kỳ n trong tập N , $N'=\{n\}$, $A'=\{\}$
 - 2) Nếu $N'=N$, thuật toán dừng, cây $T=(N',A')$ là cây khung
 - 3) Chọn một cung $(i,j) \in A, i \in N', j \notin N'$
 $N' := N' \cup \{j\}$; $A' := A' \cup \{(i,j)\}$;
Lặp lại bước 2.

Cây khung nhỏ nhất

- Cây có tổng trọng số trên các cạnh là nhỏ nhất
- Các giải thuật
 - Prim-Dijkstra:
 - Bắt đầu từ một nút bất kỳ, nút trở thành 1 phần cây.
 - Thêm vào cây cạnh có trọng số nhỏ nhất nối một nút của cây với 1 nút chưa có trong cây

Prim-Dijkstra



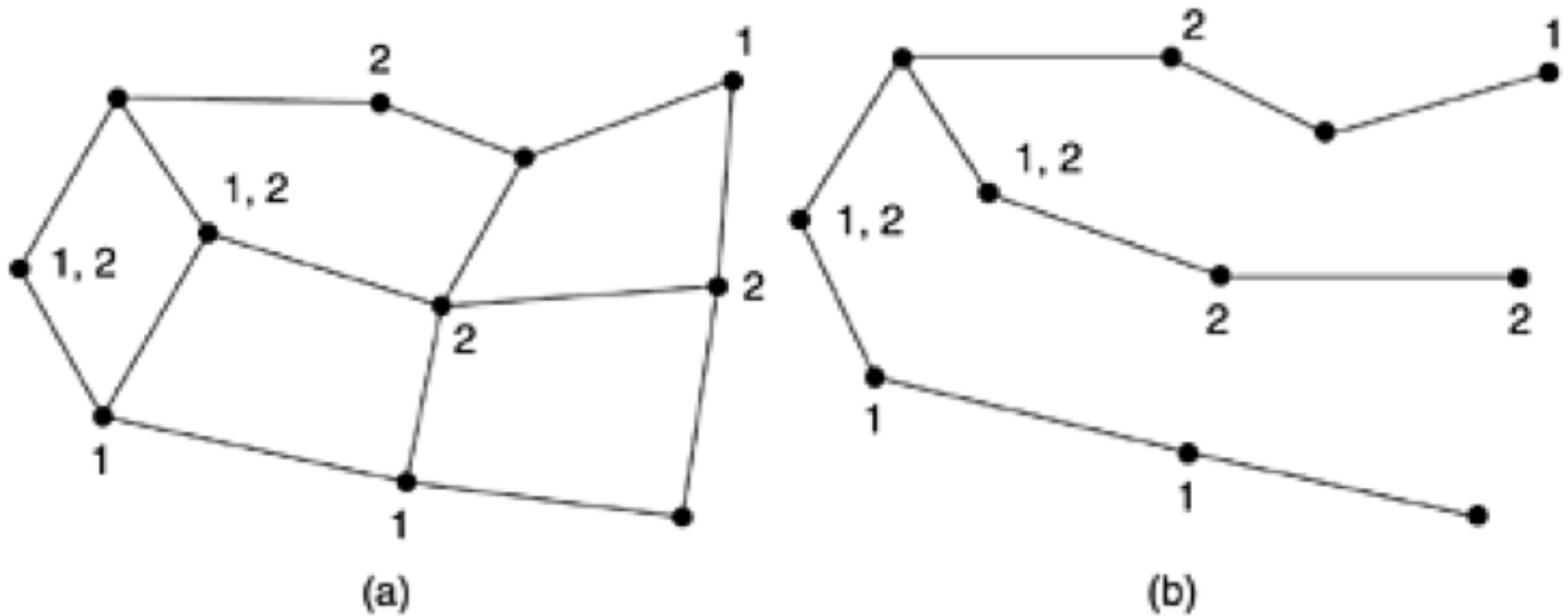
Cây khung nhỏ nhất

- Giải thuật Kruskal
 - Tạo một rừng F , mỗi đỉnh của đồ thị coi như 1 cây
 - Tập A chứa tất cả các cạnh của đồ thị.
 - Chừng nào A còn chưa rỗng lặp lại:
 - Lấy 1 cạnh có trọng số nhỏ nhất từ A
 - Nếu cạnh đó nối 2 cây của F thì cho cạnh vào rừng F và nối 2 cây thành 1 cây. Nếu không, bỏ cạnh đó
 - Kết quả thu được cây khung nhỏ nhất F .

Giải thuật tìm đường multicast

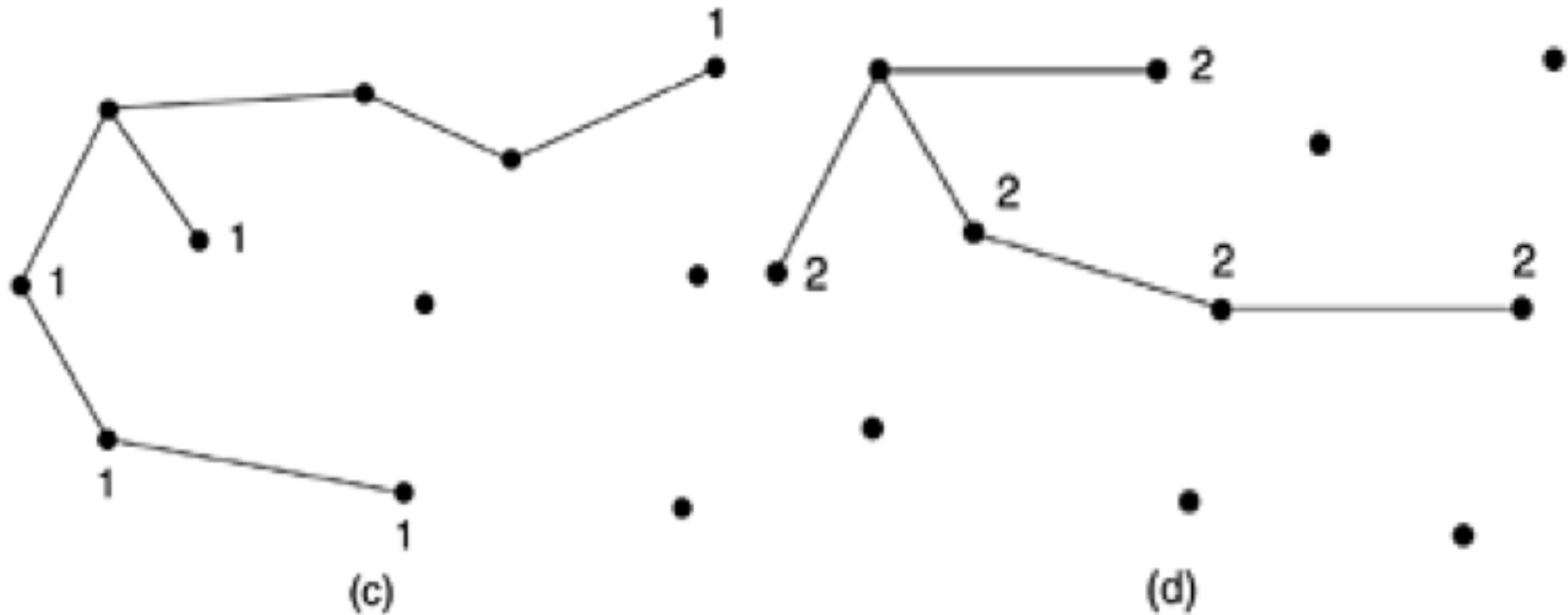
- ✓ Việc gửi gói tin đến một nhóm các dịch vụ/ nhóm địa chỉ định trước được gọi là multicasting và giải thuật gửi gói tin multicast được gọi là chọn đường đa hướng/ multicast routing,
- ✓ Multicasting khác với gói tin quảng bá (broadcasting) bởi số lượng đích đến cần chuyển tiếp.
- ✓ Để multicasting thì cần phải quản lý nhóm, một số cách thức cơ bản là tham gia nhóm, tạo nhóm, hủy nhóm và rời nhóm. Những tác vụ này thì ko liên quan đến các cách thức định tuyến.
- ✓ Để thực hiện chọn đường multicasting, các router phải tính toán được một cây khung bao trùm tất cả các router khác trong subnet, ví dụ hình 30.a và 30.b
- ✓ Sau khi xây dựng cây khung, router xây dựng các cây cắt tỉa cho từng nhóm, các gói tin multicast sẽ chỉ được chuyển đi theo cây cắt tỉa tương ứng với từng nhóm mà thôi.

Giải thuật tìm đường multicast



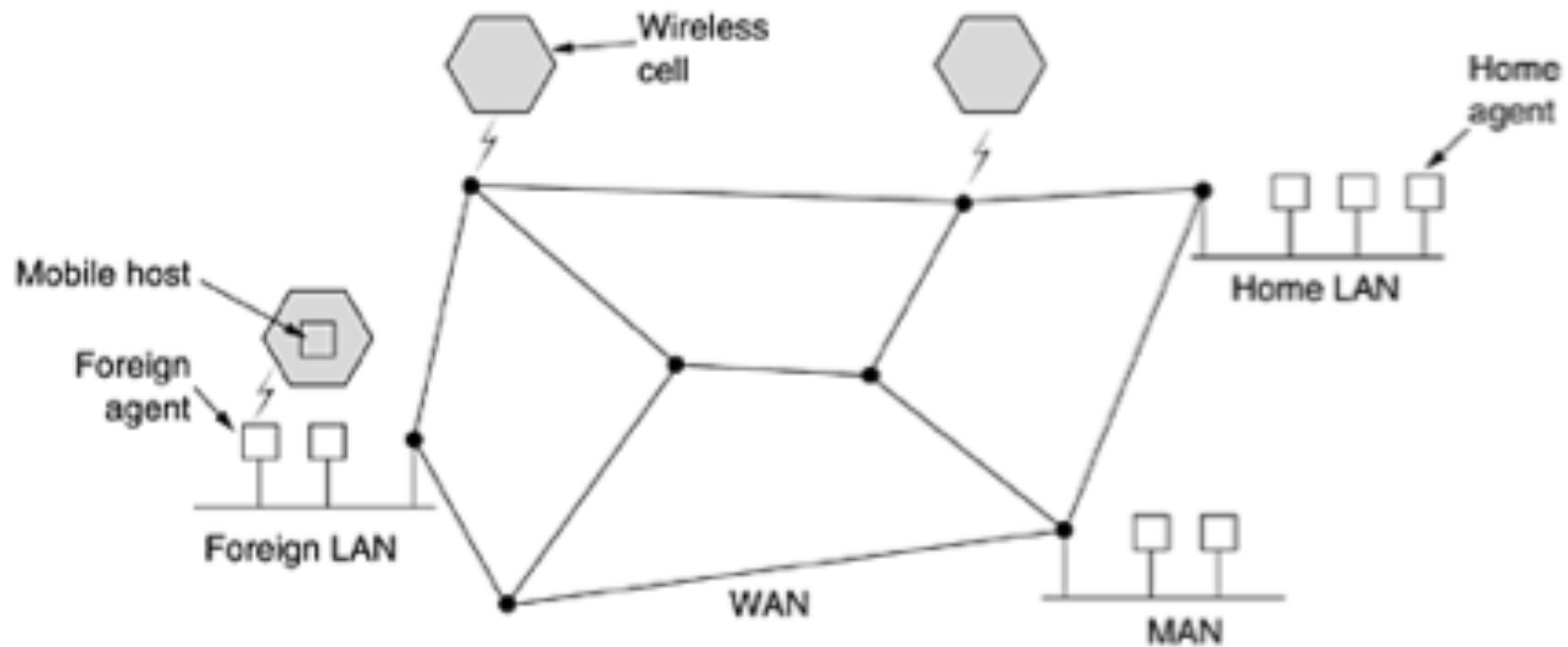
Hình 30: (a) Subnet gồm 2 nhóm 1-2; (b) Spanning tree bao trùm các nhóm

Giải thuật tìm đường multicast



Hình 30: (c) Cây multicast cho nhóm 1; (d) Cây multicast cho nhóm 2

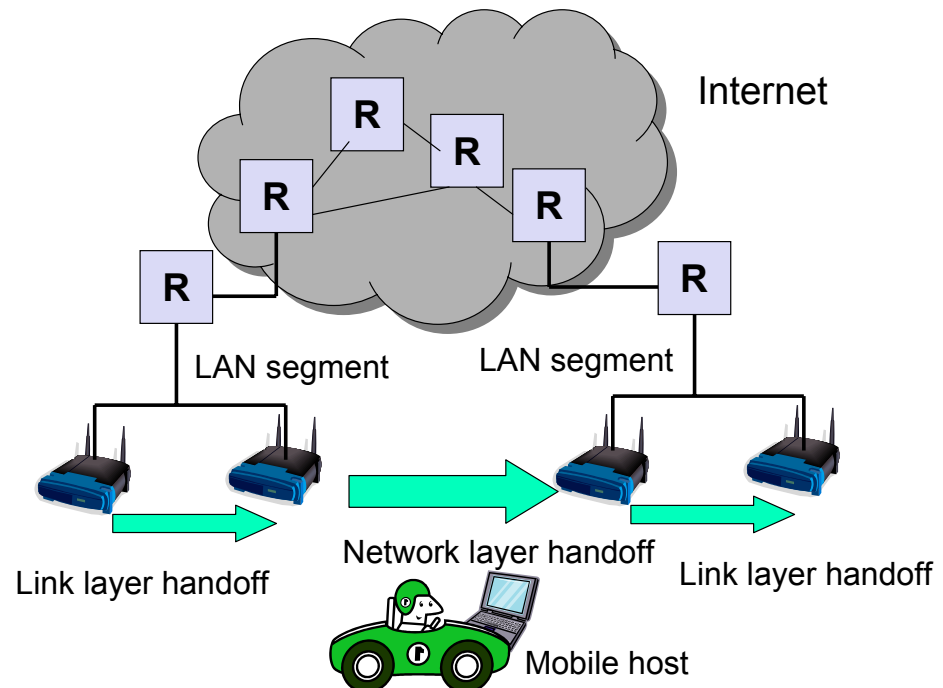
Giải thuật chọn đường cho các host di động- Mobile IP



Hình 31: Một WAN cho các LAN, MAN và các cell vô tuyến được nối kết với nhau

Chọn đường cho thiết bị di động- Mobile IP

- Mỗi thiết bị di động kết nối với 1 Base station (BS) hay Access Point (AP).
- Khi thiết bị di động, chuyển từ kết nối với BS cũ với 1 dải IP sang BS của 1 dải IP khác.
→ handoff



Chọn đường cho thiết bị di động- Mobile IP

- Thách thức:
 - Nếu đổi địa chỉ IP của thiết bị theo mạng mới → kết nối tầng trên như TCP sẽ đứt
 - Nếu giữ IP cũ, router không biết cách định hướng gói tin đến thiết bị.
- Giải pháp Mobile IP:
 - Giữ IP cũ và thêm IP của mạng mới cho thiết bị COA (Care-Of Address)

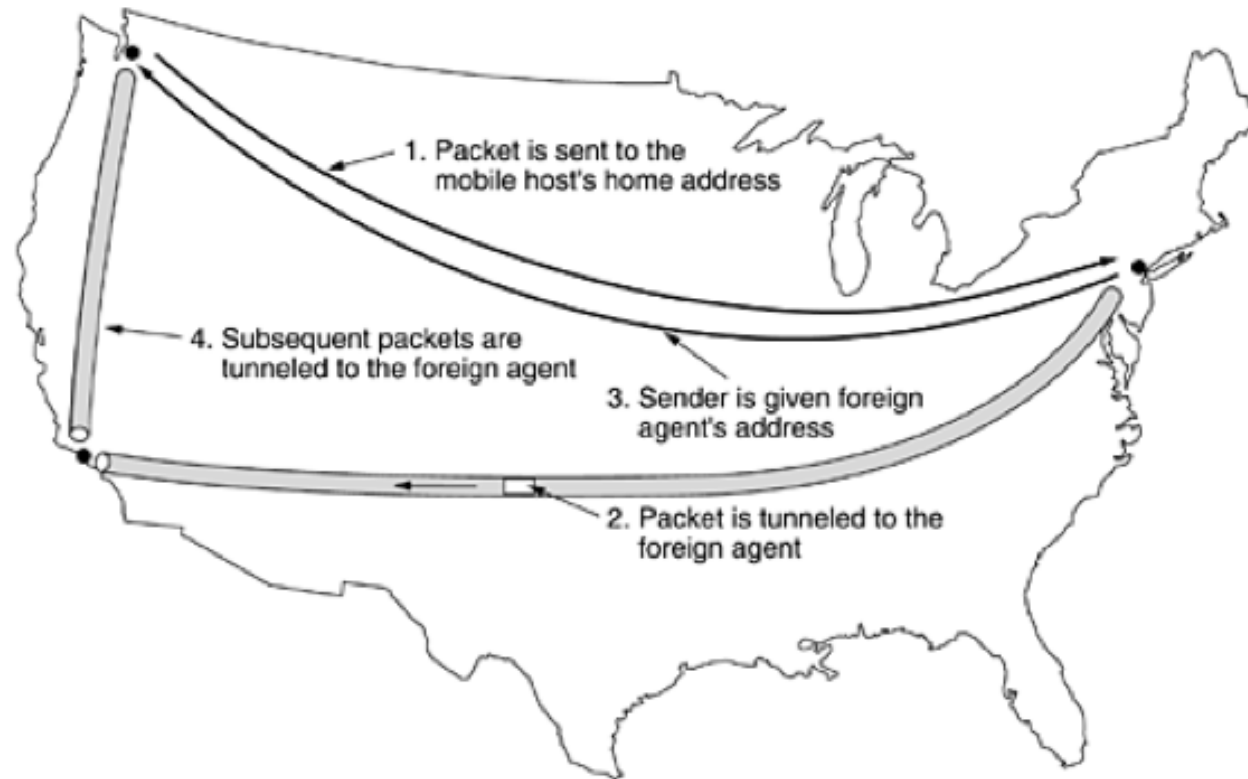
Chọn đường cho thiết bị di động- Mobile IP

- ◆ Mỗi thiết bị di động đều có một home location
- ◆ Toàn bộ không gian được chia thành các vùng (area)
- ◆ Mỗi vùng có foreign agent, quản lý các thiết bị đang đến thăm vùng
- ✓ Mỗi vùng có một home agent quản lý các thiết bị của vùng đang di chuyển thăm vùng khác
- ✓ Khi một thiết bị di động đi vào một vùng mới thiết bị di động đó phải đăng ký với một foreign agent (hoặc base station), thủ tục đăng ký thông thường như sau:
 1. Foreign agent quảng bá định kỳ 1 gói tin báo hiệu sự tồn tại của nó. Các thiết bị đầu cuối di động có thể tóm các gói tin này, hoặc ngược lại có thể gửi một gói tin báo hiệu, “Có 1 foreign agent ở vị trí này không”,
 2. Host di động đăng ký với foreign agent để cung cấp địa chỉ và các thủ tục xác thực,
 3. Foreign agent liên lạc với home agent để xác thực các thông tin này
 4. Home agent kiểm tra thông tin bảo mật, sau đó báo hiệu ngược với foreign agent
 5. Foreign nhận được thông báo từ Home agent, nó thông báo cho host di động các xác thực và thủ tục đã hoàn tất và cho phép việc kết nối được thực thi với host di động.

Chọn đường cho thiết bị di động- Mobile IP

- ✓ Khi một gói tin được gửi đến thiết bị động, gói tin sẽ được định tuyến đến vùng home của thiết bị
- ✓ Home agent xác định địa chỉ của foreign agent đang quản lý thiết bị di động
- ✓ Home agent đóng gói lại gói tin, gửi cho foreign agent
- ✓ Foreign agent chuyển gói tin cho thiết bị nhận
- ✓ Home agent cũng gửi cho người gửi địa chỉ của foreign agent.

Chọn đường cho thiết bị di động- Mobile IP

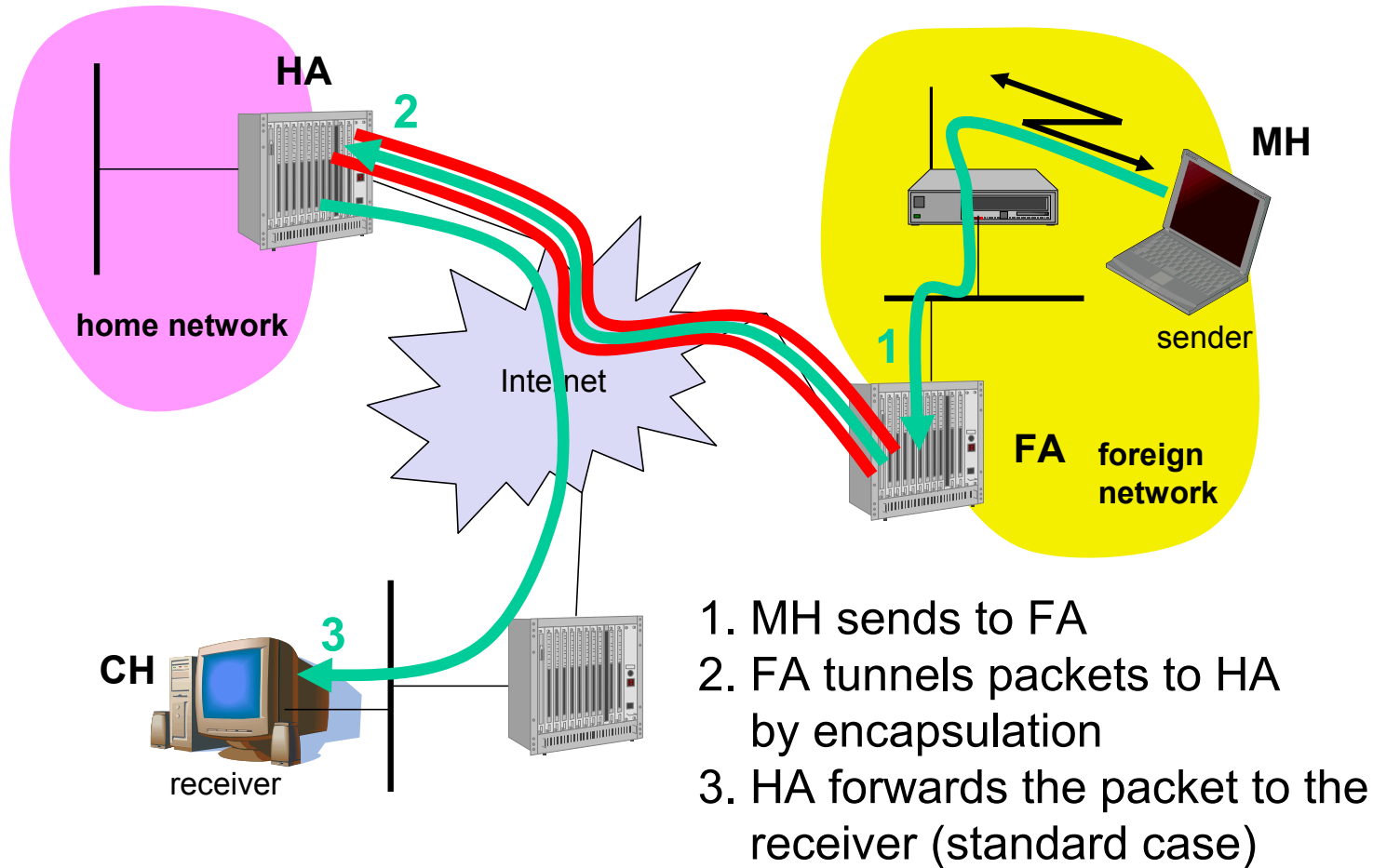


Hình 32: Thủ tục chọn đường cho các gói tin di động

Chọn đường cho thiết bị di động- Mobile IP

- Quá trình truyền ngược từ thiết bị di động đến 1 host khác.
- Thiết bị di động gửi về Foreign Agent
 - Với địa chỉ nguồn không thuộc Foreign Network, gói tin có thể bị lọc mất trong mạng
- Foreign Agent gửi theo cơ chế đường ngầm về Home Agent (đóng gói lại)
- Home Agent gửi đến host đích

Chọn đường cho thiết bị di động- Mobile IP



Giải thuật tìm đường mạng adhoc

- ✓ MANET = Mobile wireless ad-hoc network= Mạng tùy biến không dây
- ✓ MANET là một tập hợp các nút mạng có khả năng nối không dây với nhau không theo một hạ tầng định sẵn
- ✓ Mỗi nút vừa hoạt động như một nút đầu cuối (host) vừa như một thiết bị chuyển tiếp dữ liệu (router)
- ✓ Lựa chọn nút chuyển tiếp dữ liệu dựa trên tình trạng kết nối của mạng.
- ✓ Topo của mạng ad-hoc nói chung là động, do sự chuyển động của nút, sự xuất hiện của nút mới và việc nút cũ rời khỏi mạng.
- ✓ Các nút mạng phải thường xuyên khám phá topo: nghe thông báo về sự xuất hiện của nút khác, thông báo về sự xuất hiện của bản thân.
- ✓ Các giao thức định tuyến kế thừa các giải thuật định tuyến truyền thống nhưng tối ưu/ thay đổi để phù hợp với đặc điểm mạng ad-hoc

Giải thuật chọn đường cho mạng ad hoc

- ◆ Các giao thức được thiết kế cho mạng ad hoc cần được đảm bảo các tiêu chí sau:
 - ✓ Thích ứng nhanh khi topo mạng thay đổi.
 - ✓ Trong trường hợp các nút mạng di chuyển nhanh, các giao thức hoạt động theo cơ chế tập trung sẽ giảm hiệu quả rõ rệt do phải tốn nhiều thời gian để thu thập thông tin về trạng thái hiện tại và phát tán lại nó trong khi đó cấu hình mạng có thể đã thay đổi khác đi rồi,
 - ✓ Đảm bảo hiệu quả trong môi trường truyền khi các nút đứng yên,
 - ✓ Các gói tin không bị lặp gây nên hiện tượng tắc nghẽn, hao hụt băng thông trong mạng trong khi tài nguyên của các nút mạng ad hoc thường hạn chế,
 - ✓ Bảo mật gói tin:
 - ✓ truyền dẫn trong môi trường không dây sẽ dễ nguy cơ bị tấn công bằng các phương pháp như xâm nhập đường truyền, phát lại, thay đổi các gói tin tiêu đề, điều hướng các thông điệp định tuyến.

Giải thuật chọn đường cho mạng ad hoc

Uniform routing	Proactive routing	Wireless Routing Protocol (WRP)	
		Destination Sequence Distance Vector (DSDV) Routing protocol	
		Fisheye State Routing (FSR)	
		Distance Routing Effect Algorithm for Mobility (DREAM)	Location-based routing.
	Reactive routing	Dynamic Source Routing (DSR) protocol	
		Temporally Ordered Routing Algorithm (TORA)	
		Ad hoc On-demand Distance Vector Routing (AODV) protocol	
		Location Aided Routing (LAR)	Location-based routing.
		Associativity Based Routing (ABR) protocol	Link-stability based routing protocol.
		Signal Stability-base adaptive Routing protocol (SSR)	Link-stability based routing protocol.
Non-uniform routing	Zone-based routing	Zone Routing Protocol (ZRP)	Hybrid routing protocol.
		Hybrid Ad hoc Routing Protocol (HARP)	Hybrid routing protocol also.
		Zone-based Hierarchical Link State routing (ZHLS)	Hybrid routing protocol also.
		Grid Location Service (GLS)	Location service.
	Cluster-based routing	Clusterhead Gateway Switch Routing (CGSR)	
		Hierarchical State Routing (HSR)	
		Cluster Based Routing Protocol (CBRP)	
	Core-node based routing	Landmark Ad hoc Routing (LANMAR)	Proactive routing
		Core-Extraction Distributed Ad hoc Routing (CEDAR)	Reactive routing
		Optimized Link State Routing protocol (OLSR)	Proactive routing

Giải thuật tìm đường cho mạng Adhoc

- Table driven (proactive) routing
 - Các nút định tuyến dựa trên bảng định tuyến được lưu và cập nhật thường xuyên tại các nút.
- On demand (reactive) routing
 - Thực hiện tìm đường khi có yêu cầu vận chuyển dữ liệu và được thực hiện tại nguồn (source routing)

Giải thuật chọn đường cho mạng ad hoc

- Table driven (proactive) routing
 - Bảng định tuyến duy trì một danh sách các đích và đường đi.
 - Thông tin topo mạng thường xuyên được trao đổi giữa các nút mạng để duy trì hình ảnh cập nhật về topo
 - Các giao thức loại này khác nhau về:
 - Loại thông tin trao đổi giữa các nút
 - Cách các thông tin được trao đổi giữa các nút mạng
 - Nhược điểm
 - Lượng dữ liệu phải duy trì
 - Phản ứng chậm với sự thay đổi cấu trúc mạng và lỗi
 - OLSR, DSDV

Giải thuật chọn đường cho mạng ad hoc

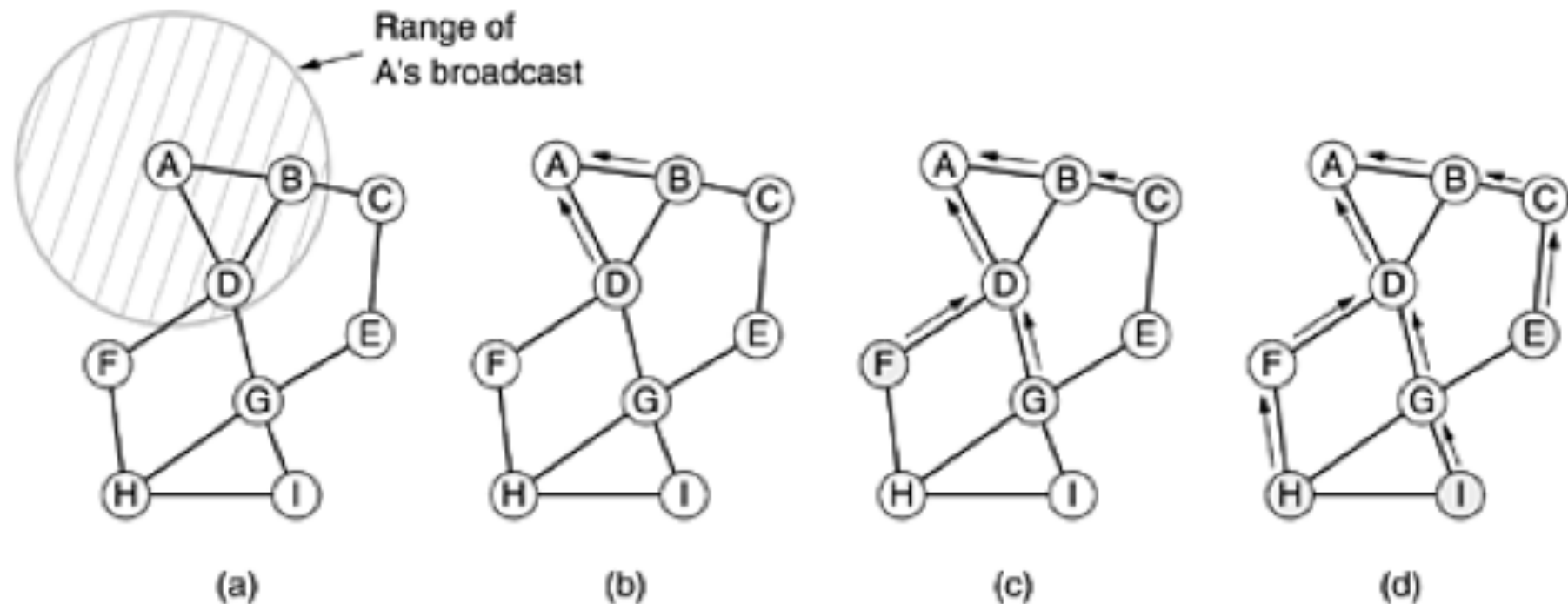
- On demand (reactive) routing
 - Khi một nút nguồn muốn gửi thông tin đến 1 nút đích, nó mới khởi động quá trình tìm đường.
 - Gửi gói tin Route Request đi toàn mạng
 - Nhược điểm:
 - Độ trễ để tìm kiếm đường đi lớn
 - Quá nhiều gói tin trên mạng khi yêu cầu đường đi
 - AODV, Dynamic source routing ...

Giải thuật chọn đường cho mạng ad hoc

- Hybrid (proactive, reactive) routing:
 - Một số đường đi triển vọng được chuẩn bị sẵn theo kiểu proactive
 - Với những yêu cầu từ những nút mạng mới, đường đi được tìm kiếm theo kiểu reactive
- Các giao thức phân cấp
 - Lựa chọn hình thức định tuyến proactive hay reactive tùy theo độ sâu của các nút mạng
 - VD: Cluster based routing

AODV

- AODV: Ad hoc On-Demand Distance Vector Routing (RFC 3561)
- là giao thức định tuyến phổ biến trên mạng ad hoc, sử dụng các gói tin route request, route reply để xây dựng các thông tin định tuyến



Hình 34: Ví dụ về tìm đường cho mạng ad hoc với giao thức AODV

AODV

- Mạng im lặng cho đến khi có một yêu cầu kết nối
- Nút cần truyền tin phát gói tin Route request broadcast toàn bộ mạng
- Các nút chuyển tiếp gói tin đồng thời lưu lại thông tin nút gửi
- Nếu nút chuyển tiếp đã có đường đi đến đích, nút này gửi lại nguồn đường đi tạm thời.
- Nút nguồn chọn đường đi ngắn nhất

Kết luận

- ◆ Các giao thức định tuyến với trọng tâm vẫn kể các giải thuật tìm đường, xây dựng đường đi ngắn nhất, cây khung nhỏ nhất như Ford-Bellman, Dijkstra, Kruskal, Suurballe.
- ◆ Các giao thức định tuyến khác nhau chủ yếu do:
 - ✓ Môi trường, đường dẫn mạng khác nhau
 - ✓ Topo mạng khác nhau
 - ✓ Các thiết bị đầu cuối có đặc tả khác nhau
- ◆ Các giao thức định tuyến nâng cao sẽ hướng đến:
 - ✓ Chất lượng dịch vụ (cost, bandwidth, delay, etc)
 - ✓ Bảo mật
 - ✓ Tiêu hao năng lượng/ tài nguyên sử dụng