

GIAO THỨC ĐỊNH TUYẾN LINK-STATE

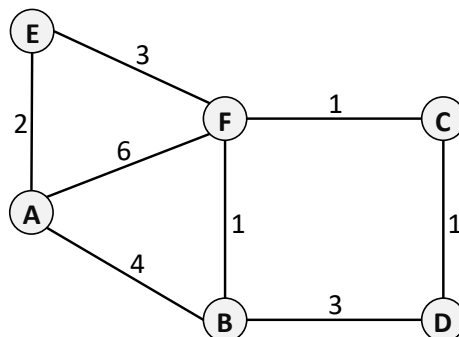
Khi áp dụng thuật toán Link-State (LS), mỗi node tạo một cơ sở dữ liệu định tuyến. Cơ sở dữ liệu này được biểu diễn dưới dạng một bản đồ mạng. Để làm điều này, mỗi node phát thông tin chi phí của nó đến các node kề cận lên toàn mạng. Sau khi thu thập đủ thông tin, mỗi node tự xây dựng bảng định tuyến cho mình. Giao thức này sử dụng thuật toán Dijkstra.

Thuật toán Dijkstra:

Gọi tập các node khảo sát là V , node đang khảo sát là s , $s \in V$

```
dist[s,s] ← 0                                (khoảng cách node nguồn là 0)
for all  $v \in V - \{s\}$ 
    do dist[s,v] ← ∞                          (đặt các khoảng cách khác là ∞)
S ← ∅                                          (tạo tập hợp các visited node S là rỗng)
Q ← V                                         (tạo tập hợp các unvisited node Q)
while Q ≠ ∅                                  (Khi Q không rỗng)
do u ← Mindistance (Q|dist)                  (Chọn node gần nguồn nhất trong Q)
    S ← S ∪ {u}                              (đưa vào tập hợp được chọn S)
    for all ( $v \in \text{neighbours}[u]$ ) and ( $v \notin S$ )
        do if dist[s,v] > dist[s,u] + w(u,v)    (nếu có đường ngắn hơn đến v)
            then dist[s,v] ← dist[s,u] + w(u,v) (cập nhật đường mới)
            (lặp cho đến điều kiện dừng)
return dist
```

Ví dụ: áp dụng thuật toán định tuyến Link-state (Dijkstra) cho hệ thống sau:



Các node có mối liên hệ neighbour như sau:

$$N(A) = \{B, E, F\};$$

$$N(B) = \{A, D, F\};$$

$$N(C) = \{D, F\}$$

$$N(D) = \{B, C\};$$

$$N(E) = \{A, F\};$$

$$N(F) = \{A, B, C, E\}$$

	Q	S	Current	A	B	C	D	E	F
0	{A,B,C,D,E,F}	{}		0	∞	∞	∞	∞	∞

Vòng lặp thứ 0: Giả sử đang khảo sát node mạng A, hàng thứ 0 là bước khởi tạo bảng định tuyến của A: node này sẽ đặt cost cho mình là 0 và tất cả các node còn lại sẽ là ∞ .

Q là tập hợp các node chưa được định tuyến $Q = \{A, B, C, D, E, F\}$, và S là tập hợp các node được định tuyến $S = \{\emptyset\}$. A sẽ so sánh Cost của hàng 0 và chọn ra Cost bé nhất: $Cost(A) = 0$ để làm current node cho vòng lặp kế tiếp. Node A sẽ được xóa khỏi Q và thêm vào S: $S = \{A\}$.

	Q	S	Current	A	B	C	D	E	F
0	{A,B,C,D,E,F}	{}		0	∞	∞	∞	∞	∞
1	{B,C,D,E,F}	{A}	A		4 _A	∞	∞	2 _A	6 _A

Vòng lặp thứ 1: Tính cost của A đến các neighbours của current node thông qua current node, nếu neighbour thuộc S thì bỏ qua việc tính toán node này.

$$d[A, B] = \text{dist}[A, A] + w(A, B) = 0 + 4 = 4_A$$

$$\square \quad d[A, B] < \text{dist}[A, B] \rightarrow \text{Cập nhật.}$$

$$d[A, E] = \text{dist}[A, A] + w(A, E) = 0 + 2 = 2_A$$

$$\square \quad d[A, E] < \text{dist}[A, E] \rightarrow \text{Cập nhật.}$$

$$d[A, F] = \text{dist}[A, A] + w(A, F) = 0 + 6 = 6_A$$

$$\square \quad d[A, F] < \text{dist}[A, F] \rightarrow \text{Cập nhật.}$$

Xét hàng 1, ta thấy giá trị của E là thấp nhất vì vậy E được chọn làm current node cho vòng lặp tới. Node E sẽ được xóa ra khỏi Q và thêm vào S: $S = \{A, E\}$.

	Q	S	Current	A	B	C	D	E	F
0	{A,B,C,D,E,F}	{}		0	∞	∞	∞	∞	∞
1	{B,C,D,E,F}	{A}	A		4 _A	∞	∞	2 _A	6 _A

2	{B,C,D,F}	{A,E}	E		4 _A	∞	∞		5 _E
---	-----------	-------	---	--	----------------	---	---	--	----------------

Vòng lặp thứ 2: Tính cost của A đến các neighbours của current node - E thông qua E, nếu neighbour thuộc S thì bỏ qua việc tính toán node này.

$$d[A,F] = \text{dist}[A,E] + w(E,F) = 2 + 3 = 5_E$$

□ $d[A,E] < \text{dist}[A,E] \rightarrow$ Cập nhật.

Xét hàng 2, mặc dù tìm được một đường tốt hơn đến F nhưng cost đến B nhỏ hơn nên B được chọn làm current node cho vòng lặp kế tiếp. B được xóa ra khỏi Q và thêm vào S: $S=\{A,E,B\}$.

Sau 6 vòng lặp thì bảng định tuyến hình thành, node A có thể gửi tin đến tất cả các node còn lại với đường đi tối ưu.

	Q	S	Current	A	B	C	D	E	F
0	{A,B,C,D,E,F}	{}		0	∞	∞	∞	∞	∞
1	{B,C,D,E,F}	{A}	A		4 _A	∞	∞	2 _A	6 _A
2	{B,C,D,F}	{A,E}	E		4 _A	∞	∞		5 _E
3	{C,D,F}	{A,E,B}	B			∞	7 _B		5 _E
4	{C,D}	{A,E,B,F}	F			6 _F	7 _B		
5	{D}	{A,E,B,F,C}	C				7 _B		
6	{}	{A,E,B,F,C,D}	D						