

# KIẾN TRÚC ĐỊNH TUYẾN:

## Hạt nhân, Ngang hàng và Các thuật toán

### Phần mở đầu: Tổng quan

#### Ta đã biết

- Truyền tin thông thường theo hình thức **Point-to-Point**:
  - *Thông qua các giao thức TCP và UDP*
- **Truyền tin liên mạng** được thực hiện thông qua **Router**:
  - *Thông qua các giao thức IP, ICMP*
  - *Ta xét đến các Router là các thiết bị chuyên dụng, phân biệt với các host*
  - Dựa trên giả định: Router “**bằng cách nào đó**” tìm ra con đường truyền tin giữa các host.

#### Đặt vấn đề

- Truyền tin liên mạng thường thông qua nhiều Router trung gian
  - *Và thường có nhiều con đường để kết nối giữa 2 host*
- Dẫn tới bài toán định tuyến:
  - Làm thế nào để Router có thể tìm ra đường kết nối **nếu có** giữa 2 host và chọn ra được đường **tốt nhất**

#### Các nội dung chính

- Nội dung 1: Đưa ra những **khái niệm ban đầu** về vấn đề định tuyến
- Nội dung 2: Tìm hiểu về **kiến trúc Internet** và ảnh hưởng của nó đến vấn đề định tuyến
- Nội dung 3: Tìm hiểu về một số **lớp ý tưởng** cho các thuật toán định tuyến

# Nội dung 1: Khái niệm nền tảng

## Nguồn gốc của Forwarding Table

### Nhắc lại về Forwarding Table

- Forwarding Table - Bảng chuyển tiếp
  - Ở các chương trước đã nói đến Forwarding Table hay IP-table (trong một số trường hợp nhất định còn gọi là Routing Table)
  - Forwarding Table được Router dùng để lưu trữ **thông tin giúp cho việc chuyển tiếp** gói tin đến đích
- Forwarding Table trên Host và trên Router
  - Nhìn chung là có sự khác nhau giữa bảng của host và bảng của router: trên host thường đơn giản và ít thông tin hơn
  - Trong nội dung bài, tập trung nhiều đến Forwarding Table trên Router

### Hai câu hỏi chính về Forwarding Table:

- Mỗi bảng chuyển tiếp chứa **những giá trị gì?**
- Các giá trị đó được **lấy từ đâu, như thế nào?**

### Một số thông tin ban đầu:

- Việc thiết lập bảng chuyển tiếp dựa trên 2 giai đoạn: Khởi tạo và Cập nhật
- Khởi tạo bảng định tuyến:
  - Mỗi router sẽ tạo bảng định tuyến ban đầu khi nó được khởi chạy
  - 3 cách khởi tạo chính:
    - Đọc từ 1 thiết bị lưu trữ như flash memory hay disk (Tĩnh)
    - Chạy scripts khởi tạo và cấu hình ban đầu (Tĩnh)
    - Yêu cầu thông tin từ các router xung quanh (Động)
  - Chúng ta không quá quan tâm tới việc khởi tạo, bởi nếu có một thuật toán đủ tốt thì nếu khởi tạo có lỗi hay sai sót thì hoàn toàn có thể sửa lỗi khi update.
- Cập nhật bảng định tuyến:
  - Cập nhật thủ công: Trong một mạng nhỏ, ít sự thay đổi
  - Cập nhật tự động: Trong mạng lớn, nhiều thành phần, thường xuyên thay đổi

Đầu tiên, chúng ta sẽ xem xét về việc định tuyến 1 cách cơ bản và chung nhất qua 3 mô hình dưới đây.

## Lý thuyết tổng quan về định tuyến - Chuyển tiếp với thông tin không đầy đủ:

### Định nghĩa:

- Thông tin đầy đủ:
  - Là thông tin mô tả đầy đủ về cấu trúc, hình thái, liên kết trong, ngoài của mạng đang xét

*Giống như chúng ta muốn di chuyển trong 1 khu vực mà có 1 tấm bản đồ, trên bản đồ có mô tả đầy đủ các thông tin về cấu trúc, vị trí ... giữa mọi tuyến đường, mọi giao lộ, mọi địa chỉ trong khu vực đó*
- Thông tin không đầy đủ
  - Chỉ có thông tin về một phần của mạng đang xét

*Giống như chúng ta chỉ có bản đồ của một phần khu vực hoặc đơn giản hơn là chỉ biết nơi ta đang đứng với đích cần tới.*

### Chuyển tiếp gói tin trên Host:

- Sự khác biệt cơ bản của host và router là các host thường biết rất ít về cấu trúc của mạng mà chúng kết nối vào.
  - Các host không biết toàn bộ thông tin về những địa chỉ có thể tới hay các mạng có thể kết nối tới.
  - Trên thực tế, thường host chỉ có 2 phần trong bảng chuyển tiếp của nó: một phần dành cho mạng cục bộ và một phần dành cho router kết nối trực tiếp.
- Host gửi tất cả các gói tin liên mạng cho router (cục bộ) để chuyển tiếp:
  - Một host có thể **chuyển gói tin** thành công dù nó chỉ có thông tin chuyển tiếp không đầy đủ vì nó có thể **dựa vào router**

*Host dựa vào Router để chuyển tiếp gói tin với thông tin không đầy đủ, vậy Router có làm được việc đó không? Câu trả lời là có, trong những trường hợp nhất định. Ta sẽ xét đến một số trường hợp như sau.*

### Chuyển tiếp gói tin với thông tin không đầy đủ:

- Trường hợp 1: Đi tiếp (default gate)

*(Ví dụ và hình minh họa)*

*Nêu cách hoạt động:*

- Chẳng hạn bạn muốn đi từ Hà Nội tới Hải Phòng trên đường cao tốc.

- Nếu không có bất kỳ thiết bị gì như bản đồ hoặc GPS, ta chỉ có thể dựa vào bảng chỉ đường trên đường cao tốc
  - Biển chỉ đường đầu tiên, không có đường rẽ vào Hải Phòng, chúng ta đi thẳng.
  - Biển chỉ đường tiếp theo, vẫn không có đường rẽ vào Hải Phòng, chúng ta tiếp tục đi thẳng.
  - Tiếp tục như vậy cho đến khi gặp bảng chỉ đường có đường rẽ vào Hải Phòng thì sẽ rẽ theo đường đó

*Phân tích ví dụ:*

- Trong ví dụ chúng ta thấy có thể phân tuyến đường ra làm 2 loại:
  - Đường rẽ: đường dẫn đến một đích cụ thể
  - Đường cao tốc (đi thẳng): đường mặc định đi tiếp nếu chưa tới đường rẽ cần đến
- Một số câu hỏi đặt ra trong ví dụ này:
  - Làm sao có thể chắc chắn rằng sẽ tới điểm đến mình cần?
  - Làm sao có thể chắc chắn rằng đây là con đường tốt nhất tới đích?
- Những câu hỏi này sẽ trở nên rất phiền phức nếu bạn đi qua rất nhiều biển chỉ đường mà vẫn chưa tới đích. Câu trả lời phụ thuộc vào cấu trúc của con đường và nội dung của biển chỉ đường. Các điều kiện bắt buộc phải có để việc di chuyển chính xác đó là:
  - Không được đi ra ngoài đường cao tốc và các đường rẽ.
  - Tại mỗi ngã rẽ, biển chỉ đường phải chỉ dẫn đúng và đủ các đường rẽ tại đó, và mỗi đường rẽ phải tới 1 điểm đến “tận cùng” - đích đến là đường cụt.
  - Đường cao tốc phải đi qua tất cả các điểm đến có thể.

*Nếu đúng các tiêu chí trên thì việc di chuyển sẽ tới đích và là ngắn nhất.*

- *Ưu điểm:* phương pháp này thì giúp cho việc di chuyển dễ dàng mà không có nhiều thông tin. Ta cứ đi theo một đường mặc định đến khi gặp 1 biển chỉ đường chỉ đến đích.
- *Hạn chế:* Việc định tuyến theo phương pháp này phụ thuộc 3 yếu tố về cấu trúc, hình thái con đường đã nêu trên.
  - Chúng ta buộc phải đi theo đường cao tốc trong khi có thể có một con đường ngắn hơn nếu đi ra ngoài, hoặc nếu một đích đến có đường nối tắt đến một đích khác.
  - Đường cao tốc có thể không đi qua hết các điểm đến tồn tại.

- Trường hợp 2: Tập trung

*(Ví dụ và hình minh họa)*

*Nêu cách hoạt động:*

- Giả sử đường cao tốc ở ví dụ 1 biến thành một quảng trường
  - Toàn bộ ngã rẽ trên cao tốc đều tụ về quảng trường trung tâm đó tạo nên một mô hình hình sao.
  - Các biển chỉ dẫn trên các ngã rẽ được gộp lại thành 1 biển chỉ dẫn ở quảng trường trung tâm, trên đó có thông tin về tất cả các ngã rẽ từ trung tâm đến các điểm đến có thể.
  - Lúc này, ta muốn đi tới đâu cũng sẽ đi qua giao lộ trung tâm.

*Phân tích ví dụ:*

- Theo ví dụ này, việc định tuyến sẽ trở nên đơn giản đến không ngờ, chỉ cần đi tới quảng trường trung tâm là có thể đi tới mọi nơi
- Vấn đề: Việc mọi lưu thông đều đi qua quảng trường trung tâm có thể dẫn đến tắc nghẽn
- Giải quyết: Mở rộng quảng trường trung tâm thành một tập các tuyến đường, giao lộ trung tâm
  - Tại mỗi giao lộ đều có biển chỉ dẫn đến tất cả các đích đến bên ngoài theo con đường tốt nhất.
- Vấn đề phát sinh: Khi có vấn đề xảy ra ở một giao lộ, ta phải phân luồng lại bằng cách: Sửa lại các biển chỉ dẫn ở tất cả các giao lộ khác, ở mỗi biển (có thể) phải sửa lại tất cả các chỉ dẫn ghi trên đó

- Trường hợp 3: Phân tán

*(Ví dụ và hình minh họa)*

*Nêu cách hoạt động:*

- Từ trường hợp 1 tập giao lộ trung tâm ở ví dụ 2, ta tách ra nhiều tập giao lộ trung tâm, giống như chia ra thành nhiều thành phố, giữa mỗi thành phố có đường kết nối với nhau.
- Giả sử ta tách ra làm 2 thành phố như vậy, gọi là A và B. 2 thành phố nối với nhau bởi 1 cây cầu
- Thành phố A có chỉ dẫn về mọi con đường, mọi đích đến của nó và thành phố B cũng vậy. Nếu cần đến 1 địa chỉ mà không thấy ở 1 thành phố thì sẽ được chỉ đi ra cây cầu
  - Cây cầu ở đây là default gate

*Phân tích ví dụ:*

- *Vấn đề: Nếu có 1 địa chỉ không tồn tại ở cả 2 thành phố*
  - *Giả sử có có 1 địa chỉ (do bị lỗi) không thuộc thành phố nào cả*
  - *Khi đó, người đó không tìm thấy nó ở thành phố A và được chỉ đi ra đường mặc định là cây cầu.*
  - *Sang đến thành phố B, tiếp tục không tìm thấy địa chỉ đó và lại được chỉ đi ra cây cầu*
  - *Cứ đi qua lại cây cầu đó, tạo thành một vòng lặp vô hạn.*

## Nội dung 2: Kiến trúc Internet: Hạt nhân, Ngang hàng

### Kiến trúc ban đầu – Core

#### ARPANET – Backbone Network

- ARPANET là mạng đầu tiên áp dụng và phát triển trên mô hình TCP/IP
- Cung cấp **kết nối liên mạng**, trong đó thành phần chính và quan trọng nhất là một **mạng lưới các Router** được gọi là **Backbone Network** (mạng xương sống)
- Kiến trúc ban đầu được thiết kế theo **kiến trúc hình sao**
  - *Mạng xương sống có vai trò như tập giao lộ trung tâm*

#### Hệ thống 2 bậc – Core Router

- Các Router được chia thành 2 bậc:
  - Các Router **bên ngoài** mạng xương sống là các Router bậc 2 (**Tier 2**)
    - Sử dụng chiến lược **đường mặc định**: *chỉ chứa các địa chỉ trong nội bộ mạng và 1 địa chỉ mặc định để kết nối đến mạng xương sống.*
  - Các Router **bên trong** mạng xương sống là các Router bậc 1 (**Tier 1**)
    - Sử dụng **thông tin đầy đủ**: *chứa toàn bộ thông tin về hình trạng mạng và các địa chỉ đích, không sử dụng địa chỉ mặc định*
- Ưu điểm: **Tính tự quản** cao
  - *Các mạng nội bộ có thể thay đổi bên trong mà không ảnh hưởng đến kết nối liên mạng ra bên ngoài*
- Nhược điểm: Rủi ro về **không nhất quán**
  - *Một lỗi ở Router nội bộ có thể làm sai lệch kết quả định tuyến của mạng, dẫn tới lỗi kết nối đến các mạng khác*
  - *Các giao thức và thuật toán cần đủ mạnh để có thể tìm, sửa và quan trọng nhất là giới hạn được ảnh hưởng của các lỗi không nhất quán khi chúng xảy ra*

- Các **Tier 1 Router** được gọi là các **Core Router** (bộ định tuyến hạt nhân).
  - Core Router **phải sử dụng thông tin đầy đủ**:
    - *Trường hợp địa chỉ đích không tồn tại: Gói tin sẽ liên tục đi theo đường mặc định cho đến khi quá số bước nhảy cho phép*
    - *Trường hợp địa chỉ đích nằm quá xa so với nguồn theo chiều của đường mặc định: Gói tin có thể sẽ phải đi qua hết chuỗi Router mặc định trước khi đến đích, mặc dù có thể có đường kết nối tắt nhanh hơn.*
    - *Bởi vậy, các Core Router phải sử dụng thông tin đầy đủ để có thể định tuyến một cách tốt nhất.*
  - Để làm được việc này thì các Core Router phải có khả năng **trao đổi thông tin với nhau**
    - *Mỗi Router đều có đầy đủ thông tin về nhau để định tuyến trong toàn bộ mạng*
  - Bài toán định tuyến chủ yếu xét trên Backbone Network, giữa các Core Router với nhau
    - *Do Tier-2 Router sử dụng chiến lược Đường mặc định nên việc tìm đường không có gì phức tạp*

## Kiến trúc mở rộng – Peer

### Mạng xương sống ngang hàng – Tính phức tạp

- Xuất hiện nhiều hơn các **mạng xương sống khác nhau**: mở đầu là NSFNET.
- Tạo nên một khái niệm mới – **Peer Backbone (Mạng xương sống ngang hàng)**:
  - Peer Backbone là các mạng xương sống tồn tại **song song, tương đối độc lập** với nhau, có cùng nhiệm vụ kết nối liên mạng
- Tính phức tạp của mạng xương sống ngang hàng:
  - Kết nối giữa các mạng xương sống ngang hàng:
    - *Các mạng xương sống có thể kết nối với nhau, và kết nối với nhau qua nhiều điểm (router)*
  - Chuyển tiếp gói tin qua các mạng xương sống ngang hàng:
    - *Gói tin có thể được chuyển tiếp từ mạng xương sống này qua mạng xương sống khác bằng nhiều con đường khác nhau*

### Các mục tiêu cơ bản – Các khó khăn

- Các mục tiêu:
  - Tránh đi qua 1 mạng xương sống 2 lần
  - Cố gắng đi trong 1 mạng xương sống nhiều nhất có thể
  - Cố gắng đi theo đường ngắn nhất, tốt nhất



- Các khó khăn:

- Việc chuyển tiếp giữa 2 mạng xương sống có thể **không hiệu quả**:
  - Theo giao thức IP, nếu chỉ xét đến truyền tin bằng địa chỉ IP (cụ thể dùng địa chỉ mạng – NetID), việc nhảy giữa 2 mạng xương sống tại bất kỳ kết nối nào đều coi là như nhau do mọi router trong 1 mạng xương sống đều coi là cùng NetID, không đảm bảo được hiệu quả truyền tin
- Việc chuyển tiếp giữa 2 mạng xương sống có thể xuất hiện **vòng lặp**:
  - Giữa 2 mạng xương sống phải có sự thống nhất ở một mức độ nhất định về định tuyến. Nếu không có thể xuất hiện sự không nhất quán trong định tuyến dẫn tới một vài router trong 2 mạng được cấu hình chuyển tiếp gói tin cho nhau tạo thành vòng lặp.

## Nội dung 3: Các thuật toán

### Lan truyền định tuyến tự động - FIB

#### Lan truyền định tuyến

- **Thông tin định tuyến** cần được **lan truyền** giữa tất cả các Core Router
- Các giao thức định tuyến cần thực hiện 2 chức năng:
  - Tính toán **đường đi tốt nhất**
  - Có khả năng **ứng phó** với những **thay đổi hoặc lỗi** xuất hiện trong mạng
- Bởi vậy, ta cần đảm bảo đến **tính cơ động, tự động** của những giao thức này

#### FIB

- Về mặt khái niệm, các **giao thức định tuyến** độc lập với các **cơ chế chuyển tiếp**
  - *Khác biệt về mục đích, chức năng*
  - *Chạy các tiến trình độc lập*
  - *Sử dụng dữ liệu, vùng nhớ độc lập*
- **Forwarding Information Base (FIB)**: Bảng lưu trữ các thông tin dùng cho định tuyến
  - *Tách biệt với Forwarding Table*
  - *Về cơ bản gần giống với Forwarding Table*
  - *Lưu trữ thêm một số thông tin cần thiết cho việc định tuyến*
- Khi FIB thay đổi, các thay đổi này sẽ được **cập nhật vào Forwarding Table**
  - Tuy nhiên, việc cập nhật còn tùy theo các chính sách, quy định riêng

### Lớp thuật toán Distance-Vector (Bellman-Ford):

#### Distance-Vector (Véc-tơ khoảng cách):

- Distance: **“khoảng cách”** từ một nguồn đến một đích cụ thể
  - Việc đánh giá dựa trên nhiều tiêu chí như: số hop, độ trễ, băng thông ...
- Distance-Vector: Một cặp (Đích, khoảng cách) là một Véc-tơ khoảng cách
  - *Giống như một véc-tơ với gốc là nguồn, hướng từ nguồn đến đích, độ lớn là khoảng cách*

### Cơ chế hoạt động:

- FIB của một Router lưu trữ các thông tin về Distance-Vector của tất cả các đích đến
  - Các thông tin chính: **Địa chỉ đích**, **Khoảng cách** và **Next Hop** (Router tiếp theo)
- Mỗi Router sẽ **truyền FIB** của mình **cho các Router kế cận**
- Khi nhận được FIB của Router kế cận
  - Router sẽ **dựa vào FIB đó để cập nhật FIB** của mình
  - Sau đó Router sẽ lại truyền FIB mới của mình đến các Router kế cận
- Việc **cập nhật** dựa trên **3 quy luật**:
  - Nếu FIB của A **có 1 đích mà B không có** trong FIB của mình thì B phải thêm đích đó vào FIB của nó với next hop là A.
    - *Nếu Router kế cận biết đường đi đến một đích mà tôi không biết, tôi sẽ đi qua Router kế cận để đến đích đó.*
  - Nếu A **biết đường ngắn hơn** đến một đích D thì B sẽ phải thay next hop đến D của nó là A
    - *Nếu Router kế cận biết tuyến đường ngắn hơn đến một đích nhất định, tôi sẽ chuyển sang đi đường thông qua Router kế cận để đến đích đó.*
  - Nếu FIB của B đã có next hop đến D là A và **Distance giữa A và D thay đổi** thì B cũng sẽ thay đổi Distance tương ứng trong FIB của nó
    - *Nếu tôi đã và đang đi qua Router kế cận để đến một đích nhất định, mà khoảng cách từ Router kế cận đến đích đó thay đổi, thì khoảng cách của tôi cũng phải thay đổi theo.*

### Đặc điểm

- Trao đổi thông tin định tuyến qua **bản tin FIB** (bản tin định tuyến chứa FIB)
- Bản tin FIB **lan truyền lần lượt**
  - Router chỉ lan truyền FIB đến các Router kế cận
  - Tạo thành **từng lớp**: *Router có thay đổi, Router kế cận, Router kế kế cận, ...*
- Bản tin FIB được dùng một lần
  - Mỗi FIB của một Router chỉ được dùng ở lớp kế tiếp
- Các Router cập nhật thông tin **dựa vào các Router khác**
  - *Thông tin về thay đổi được biết đến gián tiếp thông qua thay đổi của Router kế cận*

### Phân tích ưu nhược điểm:

- Ưu điểm:
  - Tốc độ tính toán nhanh (Tính toán phân tán)
    - *Cả thuật toán được phân ra thành các bước*
    - *Mỗi một lớp Router tương ứng tính một bước*
  - Triển khai dễ dàng
    - *Thuật toán khá đơn giản, 3 quy luật cập nhật dễ triển khai*

- Nhược điểm:
  - Tốc độ lan truyền thông tin **chậm**
    - Việc tiếp cận thông tin có độ trễ cao: Phải đợi các Router ở các lớp trước tính toán và đưa ra FIB mới thì các Router ở lớp sau mới biết được có thay đổi. Do đó chậm thích nghi với những thay đổi diễn ra thường xuyên
  - Tiêu **tốn** tài nguyên lớn
    - Thông tin định tuyến chính là bảng FIB; Hơn thế nữa, mọi Router đều lan truyền FIB của nó. Điều này dẫn tới tốn tài nguyên như băng thông, bộ nhớ, ... để phục vụ quá trình định tuyến
  - Tính tin cậy **kém**
    - Các Router sẽ dùng kết quả (FIB) của Router trước đó để định tuyến nên nếu 1 Router ở một lớp phía trước tính toán sai thì tất cả các Router ở tất cả các lớp sau tính sai.

## Các vấn đề trong giao thức định tuyến

- Routing dựa trên một thao tác quan trọng là **trao đổi thông tin giữa các Router**
  - Thường sử dụng phương pháp truyền tin không kết nối (connectionless)
    - Trước đây là IP còn hiện tại là UDP
  - Kế thừa các nhược điểm: *lost, delayed, duplicated, corrupted, out of order*
  - Để khắc phục các nhược điểm đó, nhiều kỹ thuật kèm theo được sử dụng
    - Một số đã biết như là: *checksum, soft-state hoặc acknowledgment, sequence number, ...*
    - Một số kỹ thuật khác trong từng giao thức cụ thể sẽ được giới thiệu trong các chương sau
- Distance-Vector có các nhược điểm riêng của nó: **Chậm, Tốn, Kém tin cậy**
  - để khắc phục các nhược điểm này, chúng ta sẽ đưa ra 1 chiến lược mới. Đó là sử dụng *link-state*.

## Lớp thuật toán Link-State (SPF)

### Link-State (Trạng thái liên kết):

- Là **thông tin kết nối** giữa 2 Router kề nhau
- Có được thông tin qua thao tác **kiểm tra kết nối**:
  - Một Router sẽ định kỳ gửi gói tin (đơn giản nhất có thể là ping) tới Router kế cận để kiểm tra.
  - Nếu Router đó phản hồi lại, Link-State được coi là thông; Nếu không, Link-State được coi là ngắt.

### Cơ chế hoạt động:

- Mỗi Router sẽ **lưu thông tin** về “bản đồ” mạng (cấu trúc, hình trạng ...)
  - Giống như một đồ thị, trong đó chứa tất cả các Router và kết nối giữa chúng.
  - Theo ngôn ngữ đồ thị: mỗi Router sẽ là 1 đỉnh, và kết nối giữa 2 Router kề nhau là 1 cạnh
- Mỗi Router sẽ định kỳ **broadcast Link-State** của mình:
  - Bản tin Link-State của các Router có dạng: “Kết nối giữa tôi với Router X còn thông” hoặc “Kết nối giữa tôi với Router X đã bị ngắt”.
  - Thông báo trạng thái này không chỉ định tuyến đường - nó chỉ đơn giản thông báo về việc có thể kết nối hay không giữa 1 cặp router.

- Khi nhận được bản tin Link-State, mỗi Router **cập nhật lại** bản đồ mạng của nó
  - Đầu tiên, nó lấy thông tin về cặp router đó và đảm bảo rằng 2 Router này đã tồn tại và đã có cạnh liên kết giữa chúng, nếu chưa thì phải thêm vào.
  - Sau đó, nó để update lại trạng thái của cạnh đó là Up hay Down.
- Sau đó, Router **định tuyến lại** dựa trên bản đồ mới bằng thuật toán SPF (Dijkstra)
  - Kết quả về cơ bản là một FIB, và nếu có quyền, sẽ được dùng để thay đổi bảng chuyển tiếp.

#### Đặc điểm:

- Trao đổi thông tin định tuyến qua **bản tin Link-State**
- Bản tin link-state truyền dưới dạng **broadcast**
  - Router sử dụng broadcast nếu hạ tầng mạng hỗ trợ broadcast
  - Nếu không có hỗ trợ broadcast, sẽ phải lần lượt được chuyển tiếp Point-to-Point đến tất cả các Router khác.
- Mỗi Router tính toán định tuyến dựa trên **dữ liệu của mình**

#### Phân tích ưu nhược điểm:

- Ưu điểm: Khắc phục những hạn chế của lớp thuật toán Distance-Vector
  - Thông tin về thay đổi trong mạng được broadcast làm tăng tốc độ cập nhật của các Router về thay đổi đó
  - Bản tin Link-State nhỏ hơn rất nhiều so với bản tin chứa FIB
  - Các Router không dùng kết quả của Router khác nên nếu 1 Router tính toán lỗi, sẽ không ảnh hưởng tới các Router khác.
- Nhược điểm:
  - Nhược điểm chung về truyền tin không kết nối
    - Nếu thông tin về bản đồ ban đầu, hoặc thông tin về một thay đổi nào đó của 2 hoặc nhiều Router có sự khác biệt sẽ dẫn tới kết quả sai về định tuyến trong mạng
  - Yêu cầu cao hơn về hiệu năng của Router và lập trình:
    - Phải chứa thông tin đồ thị về toàn bộ mạng
    - Phải triển khai được thuật toán định tuyến (Dijkstra)

## Tổng kết

- Truyền tin liên mạng thông qua mạng lưới nhiều Router
- Bài toán về định tuyến: Tìm đường và chọn đường tốt nhất
- Có 2 chiến lược định tuyến:
  - Sử dụng thông tin đầy đủ: Có đầy đủ thông tin về cấu trúc, hình thái, kết nối ... của toàn bộ mạng
  - Sử dụng thông tin không đầy đủ (dùng Default Gate): Có thông tin về một phần của mạng
- Kiến trúc Internet chia làm 2 phần:
  - Local Network: Sử dụng Tier-2 Router, sử dụng thông tin không đầy đủ
  - Backbone Network: Sử dụng Tier-1 (Core) Router, sử dụng thông tin đầy đủ
- Backbone Network có nhiệm vụ kết nối các Local Network
- Bài toán định tuyến chủ yếu xét trong Backbone Network
- Có nhiều Backbone Network tồn tại song song tạo thành hệ thống Peer Backbone
- 2 lớp thuật toán cơ bản trong định tuyến:
  - Distance-Vector
  - Link-State