



Lớp con MAC (Media Access Control Sublayer)

Bởi:

unknown

Lớp con MAC (Media Access Control Sublayer)

Như đã trình bày ở trên, chương này trình bày về mạng LAN – mạng dạng truyền quảng bá và các giao thức truyền quảng bá của nó.

Trong bất kỳ mạng dạng quảng bá nào, vấn đề then chốt luôn là cách thức người ta quyết định ai có quyền truy cập kênh truyền tại một thời điểm. Để làm rõ vấn đề hơn, hãy xem xét ví dụ sau: Có sáu người đang họp thông qua hệ thống điện thoại, mọi người đều được nối kết để có thể nghe và nói với những người khác. Khi một người ngừng nói mà có hai người hoặc nhiều hơn cùng phát biểu tiếp sẽ tạo ra tình trạng lộn xộn. Trong các cuộc họp dạng gặp mặt trực tiếp, tình trạng lộn xộn này có thể được giải quyết bằng cách đưa tay xin phát biểu. Nhưng trong hệ thống hội thảo thông qua điện thoại này, khi mà đường truyền rảnh, việc quyết định ai sẽ nói tiếp có vẻ khó làm hơn. Đã có nhiều giao thức dùng giải quyết vấn đề trên. Và chúng chính là nội dung trình bày của phần này. Nói một cách khác, các kênh truyền dạng quảng bá thỉnh thoảng còn được gọi là các kênh đa truy cập (multiaccess channels) hay là các kênh truy cập ngẫu nhiên (random access channels).

Các giao thức được sử dụng để quyết định ai có quyền truy cập đường truyền quảng bá trước được gom vào trong một lớp con của tầng liên kết dữ liệu gọi là lớp con MAC. Lớp con MAC là đặc biệt quan trọng trong mạng LAN, do nhiều mạng LAN sử dụng đường truyền dạng quảng bá như là phương tiện truyền thông nền tảng. Các mạng WAN, theo xu hướng ngược lại, lại dùng các nối kết dạng điểm-điểm (ngoại trừ các mạng dùng vệ tinh).

Về cơ bản, có ba phương pháp điều khiển truy cập đường truyền: Chia kênh, truy cập ngẫu nhiên (Random Access) và phân lượt (“Taking-turns”). Giải thích cụ thể về ba phương pháp điều khiển truy cập đường truyền trên sẽ được trình bày ngay sau đây.

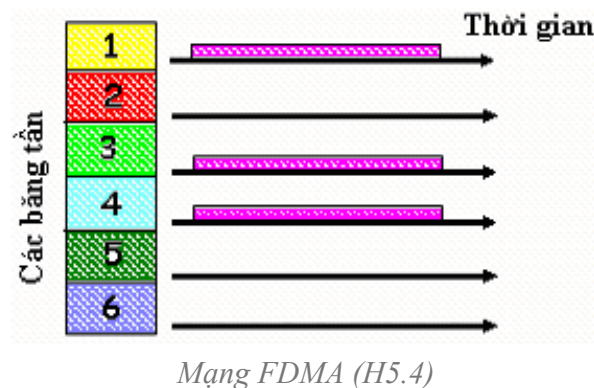
Phương pháp chia kênh

Ý tưởng chung của phương pháp này là: đường truyền sẽ được chia thành nhiều kênh truyền, mỗi kênh truyền sẽ được cấp phát riêng cho một trạm. Có ba phương pháp chia kênh chính: FDMA, TDMA, CDMA.

Chia tần số (FDMA – Frequency Division Multiple Access)

Một phương thức truyền thông để chia sẻ một kênh truyền đơn cho nhiều người dùng cạnh tranh là Chia tần số (FDMA). Phổ của kênh truyền được chia thành nhiều băng tần (frequency bands) khác nhau. Mỗi trạm được gán cho một băng tần cố định. Những trạm nào được cấp băng tần mà không có dữ liệu để truyền thì ở trong trạng thái nhàn rỗi (idle).

Ví dụ: Một mạng LAN có sáu trạm, các trạm 1, 3, 4 có dữ liệu cần truyền, các trạm 2, 5, 6 nhàn rỗi.



Nhận xét:

- Do mỗi người dùng được cấp một băng tần riêng, nên không có sự đụng độ xảy ra. Khi chỉ có số lượng người dùng nhỏ và ổn định, mỗi người dùng cần giao tiếp nhiều thì FDMA chính là cơ chế điều khiển truy cập đường truyền hiệu quả.
- Tuy nhiên, khi mà lượng người gửi dữ liệu là lớn và liên tục thay đổi hoặc đường truyền vượt quá khả năng phục vụ thì FDMA bộc lộ một số vấn đề. Nếu phổ đường truyền được chia làm N vùng và có ít hơn N người dùng cần truy cập đường truyền, thì một phần lớn phổ đường truyền bị lãng phí. Ngược lại, có nhiều hơn N người dùng có nhu cầu truyền dữ liệu thì một số người dùng sẽ phải bị từ chối không có truy cập đường truyền vì thiếu băng thông. Tuy nhiên, nếu lại giả sử rằng số lượng người dùng bằng cách nào đó luôn được giữ ổn định ở con số N, thì việc chia kênh truyền thành những kênh truyền con như thể tự thân là không hiệu quả. Lý do cơ bản ở đây là: nếu có vài người dùng rỗi,

không truyền dữ liệu thì những kênh truyền con cấp cho những người dùng này bị lãng phí.

- Có thể dễ dàng thấy được hiệu năng nghèo nàn của FDMA từ một phép tính theo lý thuyết xếp hàng đơn giản. Bắt đầu là thời gian trì hoãn trung bình T trong một kênh truyền có dung lượng C bps, với tỉ lệ đến trung bình là λ khung/giây, mỗi khung có chiều dài được chỉ ra từ hàm phân phối mũ với giá trị trung bình là $1/\mu$ bit/khung. Với các tham số trên ta có được tỉ lệ phục vụ là μC khung/giây. Từ lý thuyết xếp hàng ta có:

$$T = \frac{1}{\mu C - \lambda}$$

Ví dụ: nếu $C = 100$ Mbps, $1/\mu = 10000$ bits và $\lambda = 5000$ khung/giây thì $T = 200$ μs .

Bây giờ nếu ta chia kênh lớn này thành N kênh truyền nhỏ độc lập, mỗi kênh truyền nhỏ có dung lượng C/N bps. Tỉ lệ trung bình các khung đến các kênh truyền nhỏ bây giờ là λ/N . Tính toán lại T chúng ta có:

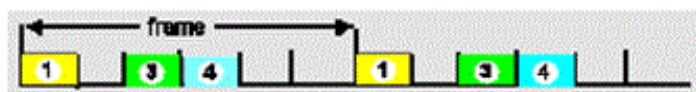
$$T_{FDMA} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT$$

Thời gian chờ đợi trung bình trong các kênh truyền con sử dụng FDMA là xấu hơn gấp N lần so với trường hợp ta sắp xếp cho các khung được truyền tuần tự trong một kênh lớn.

Chia thời gian (TDMA – Time Division Multiple Access)

Trong phương pháp này, các trạm sẽ xoay vòng (round) để truy cập đường truyền. Vòng ở đây có thể hiểu là vòng thời gian. Một vòng thời gian là khoảng thời gian đủ để cho tất cả các trạm trong LAN đều được quyền truyền dữ liệu. Qui tắc xoay vòng như sau: một vòng thời gian sẽ được chia đều thành các khe (slot) thời gian bằng nhau, mỗi trạm sẽ được cấp một khe thời gian – đủ để nó có thể truyền hết một gói tin. Những trạm nào tới lượt được cấp cho khe thời gian của mình mà không có dữ liệu để truyền thì vẫn chiếm lấy khe thời gian đó, và khoảng thời gian bị chiếm này được gọi là thời gian nhàn rỗi (idle time). Tập hợp tất cả các khe thời gian trong một vòng được gọi lại là khung (frame).

Ví dụ:



Mạng TDMA (H5.5)

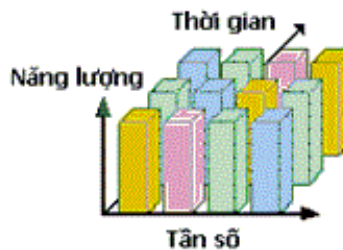
Mạng LAN dùng cơ chế truy cập đường truyền TDMA trên có sáu trạm. Các trạm 1, 3, 4 có dữ liệu cần truyền. Các trạm 2, 5, 6 nhàn rỗi.

Chúng ta cũng áp dụng cùng một nhận xét về mạng TDMA như mạng FDMA. Mỗi người dùng được cấp phát một khe thời gian. Và nếu người dùng không sử dụng khe thời gian này để truyền dữ liệu thì thời gian sẽ bị lãng phí.

Kết hợp giữa FDMA và TDMA

Trong thực tế, hai kỹ thuật TDMA và FDMA thường được kết hợp sử dụng với nhau, ví dụ như trong các mạng điện thoại di động.

- Các điện thoại di động TDMA sử dụng các kênh 30 KHz, mỗi kênh lại được chia thành ba khe thời gian. Một thiết bị cầm tay sử dụng một khe thời gian cho việc gửi và một khe khác cho việc nhận dữ liệu. Chẳng hạn như các hệ thống: Cingular (Nokia 8265, TDMA 800/ 1900 MHz, AMPS 800 MHz), AT&T Wireless.
- Hệ thống GSM sử dụng các kênh 200 KHz được chia thành 8 khe thời gian. Một thiết bị cầm tay sẽ sử dụng một khe thời gian trong hai kênh khác nhau để gửi và nhận thông tin. Các hệ thống Cingular, T-Mobile, AT&T đang chuyển sang dùng kỹ thuật này.



Kết hợp giữa TDMA và FDMA (H5.6)

Phân chia mã (CDMA – Code Division Multiple Access)

CDMA hoàn toàn khác với FDMA và TDMA. Thay vì chia một dải tần số thành nhiều kênh truyền băng thông hẹp, CDMA cho phép mỗi trạm có quyền phát dữ liệu lên toàn bộ phổ tần của đường truyền lớn tại mọi thời điểm. Các cuộc truy cập đường truyền xảy ra đồng thời sẽ được tách biệt với nhau bởi kỹ thuật mã hóa. CDMA cũng xóa tan lo lắng cho rằng những khung dữ liệu bị đùng độ trên đường truyền sẽ bị biến dạng. Thay vào đó CDMA chỉ ra rằng nhiều tín hiệu đồng thời sẽ được cộng lại một cách tuyến tính! Kỹ thuật CDMA thường được sử dụng trong các kênh truyền quảng bá không dây (mạng điện thoại di động, vệ tinh ...).

Trước khi đi vào mô tả giải thuật CDMA, hãy xem xét một ví dụ gần giống như sau: tại một phòng đợi trong sân bay có nhiều cặp hành khách đang chuyện trò. TDM có thể

được so sánh với cảnh tượng: tất cả mọi người đều đứng giữa phòng, chờ đến lượt mình được phát biểu. FDM thì giống như cảnh tượng: mỗi một cặp được sắp vào một ô nói chuyện riêng. Còn CDMA lại giống như cảnh: mọi người đều đứng ngay trong phòng đợi, nói chuyện đồng thời, nhưng mỗi cặp chuyện trò sẽ sử dụng một ngôn ngữ riêng. Cặp nói tiếng Pháp chỉ líu lo với nhau bằng tiếng Pháp, bỏ qua mọi tiếng động không phải là tiếng Pháp và coi đó như là tiếng ồn. Vì thế, vấn đề then chốt trong CDMA là khả năng rút trích ra được tín hiệu mong muốn trong khi từ chối mọi thứ khác và coi đó là tiếng ồn ngẫu nhiên.

Trong CDMA, thời gian gửi một bit (bit time) lại được chia thành m khoảng nhỏ hơn, gọi là chip. Thông thường, có 64 hay 128 chip trên một bit, nhưng trong ví dụ phía dưới, chúng ta dùng 8 chip cho đơn giản.

Nhiều người dùng đều chia sẻ chung một băng tần, nhưng mỗi người dùng được cấp cho một mã duy nhất dài m bit gọi là dãy chip (chip sequence). Dãy chip này sẽ được dùng để mã hóa và giải mã dữ liệu của riêng người dùng này trong một kênh truyền chung đa người dùng. Ví dụ, sau đây là một dãy chip: (11110011). Để gửi bit 1, người dùng sẽ gửi đi dãy chip của mình. Còn để gửi bit 0, người dùng sẽ gửi đi phần bù của dãy chip của mình. Ví dụ với dãy chip trên, khi gửi bit 1, người dùng sẽ gửi 11110011; khi gửi bit 0 thì người dùng sẽ gửi 00001100.

Để tiện cho việc minh họa, chúng ta sẽ sử dụng các ký hiệu lưỡng cực sau: bit 0 được ký hiệu là -1, bit 1 được ký hiệu là +1.

Cũng cần phải đưa ra một định nghĩa mới: tích trong (inner product): Tích trong của hai mã S và T , ký hiệu là $S \cdot T$, được tính bằng trung bình tổng của tích các bit nội tại tương ứng của hai mã này.

$$S \cdot T = \frac{1}{m} \sum_{i=1}^m S_i T_i$$

Ví dụ:

$$S = +1+1+1-1-1+1+1-1$$

$$T = +1+1+1+1-1-1+1-1$$

$$S \cdot T = \frac{+1+1+1+(-1)+1+(-1)+1+1}{8} = \frac{1}{2}$$

Bây giờ ta xem xét cách thức cấp phát chuỗi chip cho các trạm, sao cho không gây ra lẫn lộn thông tin giữa các trạm với nhau.

Định nghĩa: Hai mã S và T có cùng chiều dài m bits được gọi là trực giao khi:

$$S \bullet T = 0.$$

Ví dụ]

$$S = +1 + 1 - 1 - 1 - 1 - 1 - 1 + 1$$

$$T = -1 - 1 + 1 - 1 - 1 - 1 + 1 + 1$$

$$S \bullet T = \frac{(-1) + (-1) + (-1) + 1 + 1 + 1 + (-1) + 1}{8} = 0$$

Nếu các người dùng trong hệ thống có các mã trực giao với nhau thì họ có thể cùng tồn tại và truyền dữ liệu một cách đồng thời với khả năng bị giao thoa dữ liệu là ít nhất.

Qui ước:

- Gọi D_i là bit dữ liệu mà người dùng i muốn mã hóa để truyền trên mạng.
- C_i là chuỗi chip (mã số) của người dùng i .

Sau đây là cách thức mã hóa tín hiệu để gửi lên đường truyền và giải mã để lấy dữ liệu đó ra:

- Tín hiệu được mã của người dùng i :

$$Z_i = D_i \times C_i$$

- Tín hiệu tổng hợp được gửi trên đường truyền:

$$Z = \sum_{i=1}^n Z_i$$

với n là tổng số người dùng gửi tín hiệu lên đường truyền tại cùng thời điểm

- Giải mã:

Dữ liệu mà người dùng i lấy về từ tín hiệu tổng hợp chung:

$$D_i = Z \bullet C_i$$

Nếu $D_i > \text{“ngưỡng”}$, coi nó là 1, ngược lại coi nó là -1

Ví dụ:

Hệ thống có 4 người dùng A, B, C, D. Các mã số tương ứng của họ như sau:

A: 0 0 0 1 1 0 1 1
B: 0 0 1 0 1 1 1 0
C: 0 1 0 1 1 1 0 0
D: 0 1 0 0 0 0 1 0

Nếu ký hiệu theo kiểu lưỡng cực thì:

Lớp con MAC (Media Access Control Sublayer)

$$\begin{aligned} A: & (-1 -1 -1 +1 +1 -1 +1 +1) \\ B: & (-1 -1 +1 -1 +1 +1 +1 -1) \\ C: & (-1 +1 -1 +1 +1 +1 -1 -1) \\ D: & (-1 +1 -1 -1 -1 -1 +1 -1) \end{aligned}$$

Đề ý các mã số A, B, C, D là trục giao!

Có sáu ví dụ:

1. Chỉ có người dùng C gửi bit 1:
2. B gửi bit 1, C gửi bit 1
3. A gửi bit 1, B gửi bit 0
4. A, C đều gửi bit 1, B gửi bit 0
5. A, B, C, D đều gửi bit 1
6. A, B, D gửi bit 1, C gửi bit 0

Ta tính toán được các mã tổng hợp gửi lên đường truyền như sau:

1) -- 1--	C	$Z = (-1 +1 -1 +1 +1 +1 -1 -1)$
2) - 1 1--	B + C	$Z = (-2 \ 0 \ 0 \ 0 +2 +2 \ 0 -2)$
3) 1 0 --	A + \bar{B}	$Z = (0 \ 0 -2 +2 \ 0 -2 \ 0 +2)$
4) 1 0 1--	A + \bar{B} + C	$Z = (-1 +1 -3 +3 +1 -1 -1 +1)$
5) 1 1 1 1	A + B + C + D	$Z = (-4 \ 0 -2 \ 0 +2 \ 0 +2 -2)$
6) 1 1 0 1	A + B + \bar{C} + D	$Z = (-2 -2 \ 0 -2 \ 0 -2 +4 \ 0)$

Bây giờ, ta tính được dữ liệu nguyên thủy của người dùng ở trạm C, sau khi đã rút trích ra từ mã tổng hợp như sau:

$$\begin{aligned} 1) Z \bullet C &= (1 +1 +1 +1 +1 +1 +1 +1)/8 = 1 \\ 2) Z \bullet C &= (2 +0 +0 +0 +2 +2 +0 +2)/8 = 1 \\ 3) Z \bullet C &= (0 +0 +2 +2 +0 -2 +0 -2)/8 = 0 \\ 4) Z \bullet C &= (1 +1 +3 +3 +1 -1 +1 -1)/8 = 1 \\ 5) Z \bullet C &= (4 +0 +2 +0 +2 +0 -2 +2)/8 = 1 \\ 6) Z \bullet C &= (2 -2 +0 -2 +0 -2 -4 +0)/8 = -1 \end{aligned}$$

Nhận xét:

- Đầu tiên, chúng ta phải giả sử rằng tất cả các dây chip được đồng bộ hóa để được gửi nhận cùng thời điểm. Nhưng trong thực tế, kiểu đồng bộ hóa như vậy là không thể có được. Những gì người ta có thể làm được để đồng bộ hóa là: người gửi và người nhận đồng bộ hóa với nhau bằng cách cho người gửi gửi một dây chip được định nghĩa trước, dây này phải đủ dài để cho bên nhận có thể theo kịp bên gửi. Tất cả các cuộc truyền nhận khác được xem như là nhiễu ngẫu nhiên. Người ta chứng minh được rằng, chuỗi chip càng dài thì xác suất

phát hiện ra chuỗi này một cách chính xác là càng cao với sự hiện diện của nhiễu.

- Cũng cần phải giả thiết rằng: bên nhận biết chính xác bên gửi là ai. Tuy trong thực tế, cần phải trung thực mà nói rằng: đặt giả thiết thì dễ hơn là làm. Nhưng hãy tin tưởng là CDMA có nhiều chi tiết phức tạp hơn và thông minh hơn để làm được chuyện đó.

Phương pháp truy cập đường truyền ngẫu nhiên (Random Access)

Trong phương pháp này, người ta để cho các trạm tự do tranh chấp đường truyền chung để truyền từng khung dữ liệu một. Nếu một trạm cần gửi một khung, nó sẽ gửi khung đó trên toàn bộ dải thông của kênh truyền. Sẽ không có sự phối hợp trình tự giữa các trạm. Nếu có hơn hai trạm phát cùng một lúc, “đụng độ” (collision) sẽ xảy ra, các khung bị đụng độ sẽ bị hư hại.

Giao thức truy cập đường truyền ngẫu nhiên được dùng để xác định:

- Làm thế nào để phát hiện đụng độ.
- Làm thế nào để phục hồi sau đụng độ.

Ví dụ về các giao thức truy cập ngẫu nhiên: slotted ALOHA và pure ALOHA, CSMA và CSMA/CD, CSMA/CA.

ALOHA

Vào những năm 1970, Norman Abramson cùng các đồng sự tại Đại học Hawaii đã phát minh ra một phương pháp mới ưu hạng dùng để giải quyết bài toán về cấp phát kênh truyền. Sau đó công việc của họ tiếp tục được mở rộng bởi nhiều nhà nghiên cứu khác. Mặc dù công trình của Abramson, được gọi là hệ thống ALOHA, sử dụng hệ thống truyền quảng bá trên sóng radio mặt đất, nhưng ý tưởng cơ sở của nó có thể áp dụng cho bất kỳ hệ thống nào trong đó những người dùng không có phối hợp với nhau sẽ tranh chấp sử dụng đường truyền chung duy nhất.

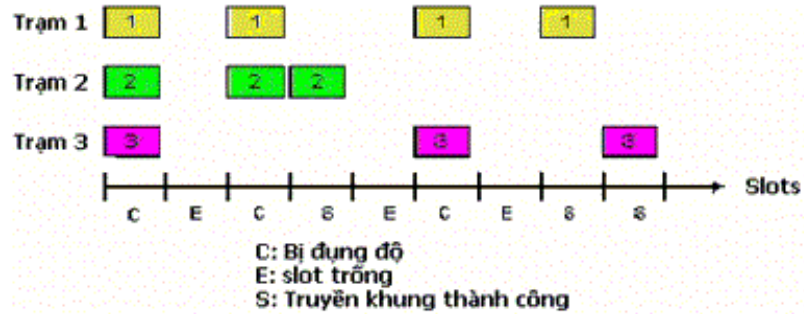
Ở đây, chúng ta sẽ thảo luận về hai phiên bản của ALOHA: pure (thuần túy) và slotted (được chia khe).

Slotted ALOHA

Thời gian được chia thành nhiều slot có kích cỡ bằng nhau (bằng thời gian truyền một khung). Một trạm muốn truyền một khung thì phải đợi đến đầu slot thời gian kế tiếp mới được truyền. Dĩ nhiên là sẽ xảy ra đụng độ và khung bị đụng độ sẽ bị hư. Tuy nhiên, dựa trên tính phản hồi của việc truyền quảng bá, trạm phát luôn có thể theo dõi xem khung của nó phát đi có bị hủy hoại hay không bằng cách lắng nghe kênh truyền. Những trạm khác cũng làm theo cách tương tự. Trong trường hợp vì lý do nào đó mà trạm không thể

dùng cơ chế lắng nghe đường truyền, hệ thống cần yêu cầu bên nhận trả lời một khung báo nhận (acknowledgement) cho bên phát. Nếu phát sinh đụng độ, trạm phát sẽ gửi lại khung tại đầu slot kế tiếp với xác suất p cho đến khi thành công.

Ví dụ minh họa: Có 3 trạm đều muốn truyền một khung thông tin.



Minh họa giao thức Slotted ALOHA (H5.7)

Do sẽ có đụng độ mà mất khung thông tin, một câu hỏi đặt ra là: đâu là tỉ suất truyền khung thành công của các trạm trong mạng?

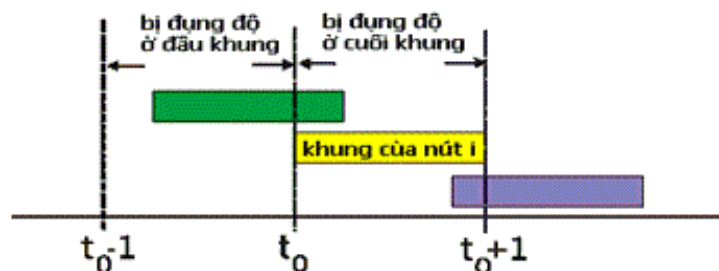
Giả sử có N trạm muốn truyền dữ liệu, mỗi trạm truyền khung thông tin của mình trong một slot với xác suất p . Xác suất để một trạm trong N trạm truyền thành công $S(p)$ được tính như sau:

$$S(p) = Np(1-p)^{N-1}$$

Khi $p = 1/N$, $S(p)$ đạt giá trị cực đại $(1 - 1/N)^{N-1}$

Pure ALOHA

Kỹ thuật Pure ALOHA đơn giản hơn Slotted ALOHA do không có sự đồng bộ hóa giữa các trạm. Mỗi khi muốn truyền một khung thông tin, trạm sẽ truyền nó ngay mà không cần đợi đến đầu của slot thời gian kế tiếp. Vì thế xác suất bị đụng độ tăng thêm! Nghĩa là khung thông tin được gửi tại thời điểm t_0 sẽ đụng độ với những khung được gửi trong khoảng thời gian $[t_0-1, t_0+1]$.



Minh họa giao thức Pure ALOHA (H5.8)

Gọi P là xác suất của một sự kiện nào đó, ta có những phân tích sau:

$$P(\text{nút } i \text{ truyền thành công}) = P(\text{đề nút } i \text{ truyền})$$

$$\times P(\text{không có nút nào khác truyền trong khoảng } [t_0-1, t_0]) .$$

$$\times P(\text{không có nút nào khác truyền trong khoảng } [t_0, t_0+1])$$

$$= p(1-p)^{N-1}(1-p)^{N-1}$$

$$S(p) = P(\text{một nút bất kỳ trong } N \text{ nút truyền thành công}) = Np(1-p)^{N-1}(1-p)^{N-1}$$

Những phân tích vừa nêu giả sử rằng luôn có thường trực N trạm trong mạng. Và trong trường hợp tối ưu, mỗi trạm thử truyền với xác suất 1/N.

Trong thực tế, số lượng các trạm thường trực trong mạng luôn thay đổi. Giả sử chúng ta có tổng cộng m trạm làm việc. n trạm là thường trực trên mạng, mỗi trạm thường trực trên mạng sẽ cố gởi khung thông tin với xác suất cố định p. m-n trạm còn lại là không thường trực, và chúng có thể gởi khung thông tin với xác suất p_a , với p_a có thể nhỏ hơn p.

Nhận xét chung về ALOHA:

- Hiệu năng thấp do không có thăm dò đường truyền trước khi gởi khung, dẫn đến việc mất nhiều thời gian cho việc phát hiện đụng độ và phục hồi sau đụng độ.
- Hoạt động theo kiểu ALOHA có khả năng dẫn đến việc hệ thống bị “chết đứng” do mọi nỗ lực gởi gói tin của tất cả các trạm đều bị đụng độ.

Slotted ALOHA trở nên quan trọng với lý do không mấy rành mạch lắm. Nó ra đời vào những năm 1970, được sử dụng trong một số hệ thống thí nghiệm thời đó, và rồi hầu như bị lãng quên. Và khi công nghệ truy cập Internet không qua cable được phát minh, đột nhiên lại phát sinh vấn đề làm sao để cấp phát đường truyền được chia sẻ cho nhiều người dùng cạnh tranh, Slotted ALOHA lại được lôi ra từ thùng rác để cứu rỗi cuộc đời. Vẫn thường có chuyện là các giao thức hợp lý một cách hoàn hảo lại không được sử dụng vì những lý do về chính trị, nhưng nhiều năm sau đó, một số người thông thái lại nhận ra rằng những giao thức bỏ đi từ lâu rồi đó lại có thể giúp họ giải quyết được vấn đề. Với lý do như vậy, trong chương này, chúng ta sẽ nghiên cứu một số giao thức ưu hạng hiện tại không được sử dụng rộng rãi lắm nhưng biết đâu có thể được sử dụng để

dùng trong các ứng dụng ở tương lai. Dĩ nhiên là chúng ta cũng sẽ nghiên cứu nhiều giao thức đang được sử dụng rộng rãi hiện nay.

CSMA – Carrier Sense Multiple Access

Giao thức ALOHA mặc dù đã chạy được, nhưng một điều đáng ngạc nhiên là người ta lại để cho các trạm làm việc tự do gửi thông tin lên đường truyền mà chẳng cần quan tâm đến việc tìm hiểu xem những trạm khác đang làm gì. Và điều đó dẫn đến rất nhiều vụ đụng độ tín hiệu. Tuy nhiên, trong mạng LAN, người ta có thể thiết kế các trạm làm việc sao cho chúng có thể điều tra xem các trạm khác đang làm gì và tự điều chỉnh hành vi của mình một cách tương ứng. Làm như vậy sẽ giúp cho hiệu năng mạng đạt được cao hơn. CSMA là một giao thức như vậy!

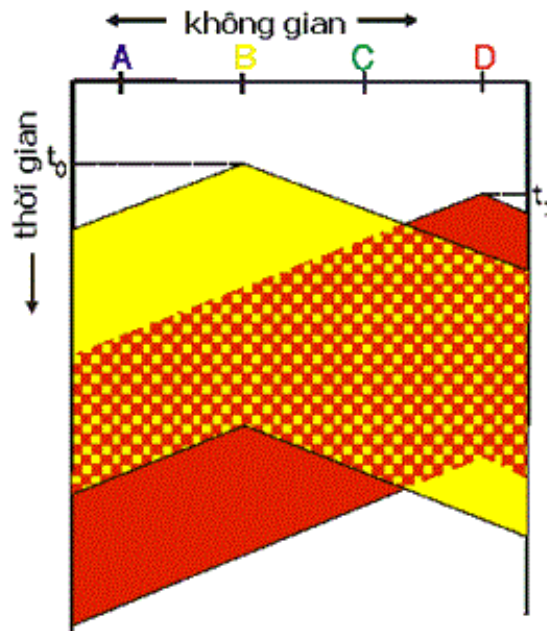
Các giao thức mà trong đó các trạm làm việc lắng nghe đường truyền trước khi đưa ra quyết định mình phải làm gì tương ứng với trạng thái đường truyền đó được gọi là các giao thức có “cảm nhận” đường truyền (carrier sense protocol). Cách thức hoạt động của CSMA như sau: lắng nghe kênh truyền, nếu thấy kênh truyền rỗi thì bắt đầu truyền khung, nếu thấy đường truyền bận thì trì hoãn lại việc gửi khung.

Thế nhưng trì hoãn việc gửi khung cho đến khi nào?

Có ba giải pháp:

- Theo dõi không kiên trì (Non-persistent CSMA): Nếu đường truyền bận, đợi trong một khoảng thời gian ngẫu nhiên rồi tiếp tục nghe lại đường truyền.
- Theo dõi kiên trì (persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng 1.
- Theo dõi kiên trì với xác suất p (P-persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng p .

Dễ thấy rằng giao thức CSMA cho dù là theo dõi đường truyền kiên trì hay không kiên trì thì khả năng tránh đụng độ vẫn tốt hơn là ALOHA. Tuy thế, đụng độ vẫn có thể xảy ra trong CSMA! Tình huống phát sinh như sau: khi một trạm vừa phát xong thì một trạm khác cũng phát sinh yêu cầu phát khung và bắt đầu nghe đường truyền. Nếu tín hiệu của trạm thứ nhất chưa đến trạm thứ hai, trạm thứ hai sẽ cho rằng đường truyền đang rảnh và bắt đầu phát khung. Như vậy đụng độ sẽ xảy ra.



Mô tả không gian và thời gian diễn ra đụng độ (H5.9)

Hậu quả của đụng độ là: khung bị mất và toàn bộ thời gian từ lúc đụng độ xảy ra cho đến khi phát xong khung là lãng phí!

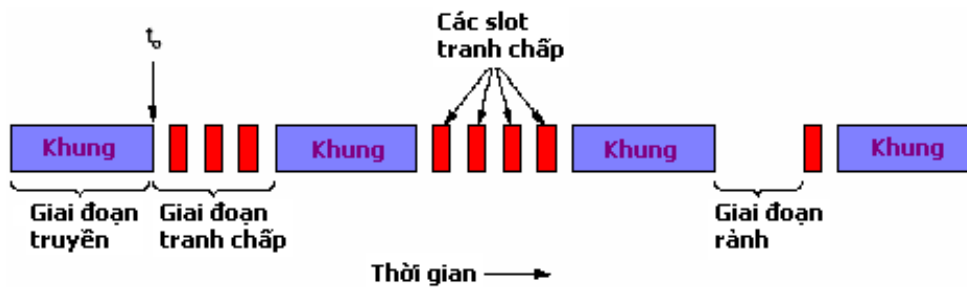
Bây giờ phát sinh vấn đề mới: các trạm có quan tâm theo dõi xem có đụng độ xảy ra không và khi đụng độ xảy ra thì các trạm sẽ làm gì?

CSMA với cơ chế theo dõi đụng độ (CSMA/CD – CSMA with Collision Detection)

CSMA/CD về cơ bản là giống như CSMA: lắng nghe trước khi truyền. Tuy nhiên CSMA/CD có hai cải tiến quan trọng là: phát hiện đụng độ và làm lại sau đụng độ.

Phát hiện đụng độ: Trạm vừa truyền vừa tiếp tục dò xét đường truyền. Ngay sau khi đụng độ được phát hiện thì trạm ngưng truyền, phát thêm một dãy nhồi (dãy nhồi này có tác dụng làm tăng cường thêm sự va chạm tín hiệu, giúp cho tất cả các trạm khác trong mạng thấy được sự đụng độ), và bắt đầu làm lại sau đụng độ.

CSMA/CD, cũng giống như các giao thức trong LAN khác, sử dụng mô hình quan niệm như trong hình sau:

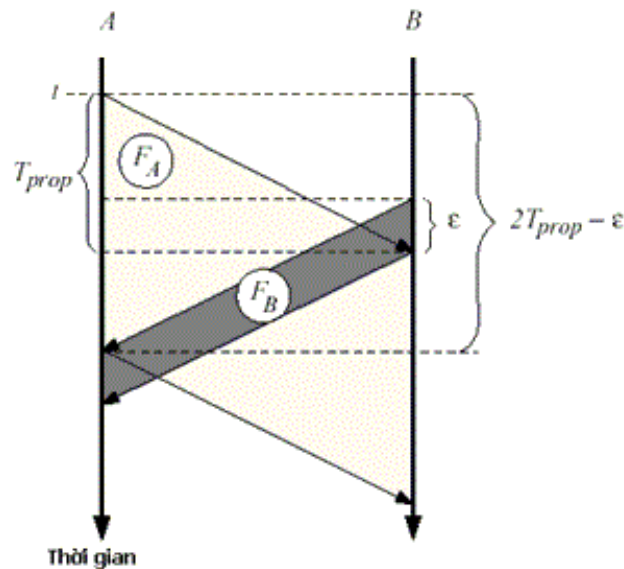


CSMA/CD có thể ở một trong ba trạng thái: tranh chấp, truyền, rảnh (H5.10)

Tại thời điểm t_0 , một trạm đã phát xong khung của nó. Bất kỳ trạm nào khác có khung cần truyền bây giờ có thể cố truyền thử. Nếu hai hoặc nhiều hơn các trạm làm như vậy cùng một lúc thì sẽ xảy ra đụng độ. Đụng độ có thể được phát hiện bằng cách theo dõi năng lượng hay độ rộng của xung của tín hiệu nhận được và đem so sánh với độ rộng của xung vừa truyền đi.

Bây giờ ta đặt ra câu hỏi: Sau khi truyền xong khung (hết giai đoạn truyền), trạm sẽ bỏ ra thời gian tối đa là bao lâu để biết được là khung của nó đã bị đụng độ hoặc nó đã truyền thành công? Gọi thời gian này là “cửa sổ va chạm” và ký hiệu nó là T_w . Phân tích sau đây sẽ cho ra câu trả lời.

Hình sau sẽ mô phỏng chi tiết về thời gian phát khung giữa hai trạm A và B ở hai đầu mút xa nhất trên đường truyền tải.



Thời gian cần thiết để truyền một khung (H5.11)

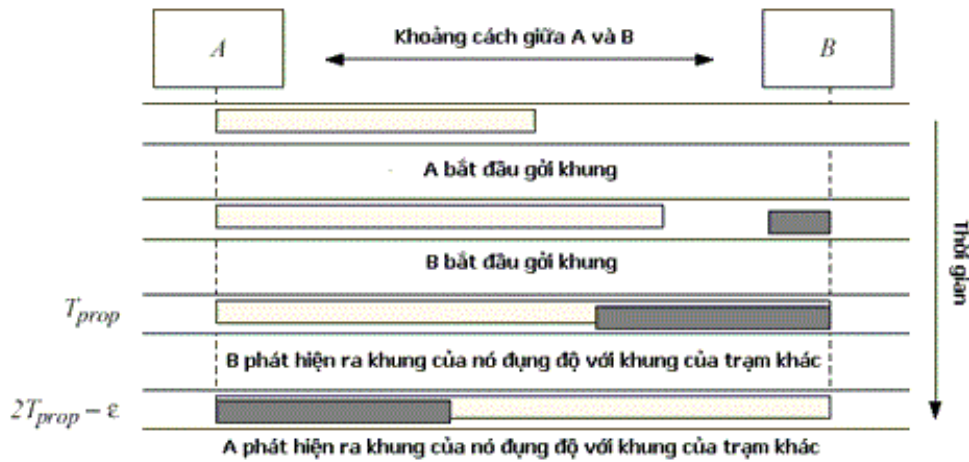
Đặt T_{prop} là thời gian lan truyền tín hiệu giữa hai đầu mút xa nhau nhất trên đường truyền tải.

- Tại thời điểm t , A bắt đầu phát đi khung dữ liệu của nó.

Lớp con MAC (Media Access Control Sublayer)

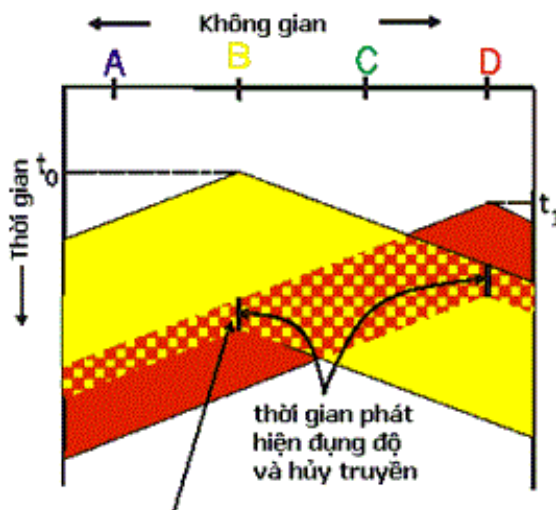
- Tại $t + T_{prop} - \epsilon$, B phát hiện kênh truyền rảnh và phát đi khung dữ liệu của nó.
- Tại $t + T_{prop}$, B phát hiện sự đụng độ.
- Tại $t + 2T_{prop} - \epsilon$, A phát hiện sự đụng độ.

Theo phân tích trên, thì $T_w = 2T_{prop}$



Phát hiện đụng độ khi truyền tin (H5.12)

Việc hủy bỏ truyền khung ngay khi phát hiện có đụng độ giúp tiết kiệm thời gian và băng thông, vì nếu cứ tiếp tục truyền khung đi nữa, khung đó vẫn hư và vẫn phải bị hủy bỏ.



thay vì lãng phí thời gian để truyền hết khung bị đụng độ, hủy bỏ việc truyền ngay sau khi đụng độ xảy ra

Xử lý khu đụng độ (H5.13)

Thuật toán back-off hoạt động như sau:

Lớp con MAC (Media Access Control Sublayer)

- Rút ngẫu nhiên ra một con số nguyên M thỏa: $0 \leq M \leq 2^k$. Trong đó $k = \min(n, 10)$, với n là tổng số lần đụng độ mà trạm đã gánh chịu.
- Kỳ hạn mà trạm phải chờ trước khi thử lại một lần truyền mới là $M \cdot T_w$.
- Khi mà n đạt đến giá trị 16 thì hủy bỏ việc truyền khung. (Trạm đã chịu đựng quá nhiều vụ đụng độ rồi, và không thể chịu đựng hơn được nữa!)

ĐÁNH GIÁ HIỆU SUẤT CỦA GIAO THỨC CSMA/CD:

Gọi:

P là kích thước của khung, ví dụ như 1000 bits.

C là dung lượng của đường truyền, ví dụ như 10 Mbps.

Ta có thời gian phát hết một khung thông tin là P/C giây.

Trung bình, chúng ta sẽ thử e lần trước khi truyền thành công một khung. Vì vậy, với mỗi lần phát thành công một khung (tốn P/C giây), ta đã mất tổng cộng $2eT_{prop}$ ($\gg 5T_{prop}$) vì đụng độ.

Thành thử hiệu năng của giao thức (tỉ lệ giữa thời gian hoạt động hữu ích trên tổng thời gian hoạt động) là:

$$\frac{\frac{P}{C}}{\frac{P}{C} + 5T_{prop}} = \frac{1}{1 + \frac{5T_{prop}}{\frac{P}{C}}} = \frac{1}{1 + 5a}, \text{ với } a = \frac{T_{prop} C}{P}$$

Giá trị của a đóng vai trò rất quan trọng đến hiệu suất hoạt động của mạng kiểu CSMA/CD.

Phương pháp phân lượt truy cập đường truyền

Bây giờ thử nhìn lại hai phương pháp điều khiển truy cập đường truyền “chia kênh” và “truy cập ngẫu nhiên”, ta sẽ thấy chúng đều có những điểm hay và hạn chế:

- Trong các giao thức dạng chia kênh, kênh truyền được phân chia một cách hiệu quả và công bằng khi tải trọng đường truyền là lớn. Tuy nhiên chúng không hiệu quả khi tải trọng của đường truyền là nhỏ: có độ trì hoãn khi truy cập kênh truyền, chỉ $1/N$ băng thông được cấp cho người dùng ngay cả khi chỉ có duy nhất người dùng đó hiện diện trong hệ thống.

- Các giao thức dạng truy cập ngẫu nhiên thì lại hoạt động hiệu quả khi tải trọng của đường truyền thấp. Nhưng khi tải trọng đường truyền cao thì phải tốn nhiều chi phí cho việc xử lý ñụng ñộ.

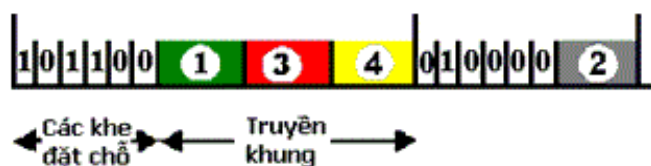
Các giao thức dạng “phân lượt” sẽ ñề ý ñến việc tận dụng những mặt mạnh của hai dạng nói trên. Ý tưởng chính của các giao thức dạng “phân lượt” là không ñể cho ñụng ñộ xảy ra bằng cách cho các trạm truy cập đường truyền một cách tuần tự.

Về cơ bản, có hai cách thức ñể “phân lượt” sử dụng đường truyền:

- **Thăm dò (polling):** Trạm chủ (master) sẽ mời các trạm tớ (slave) truyền khi ñến lượt. Lượt truyền ñược cấp phát cho trạm tớ có thể bằng cách: trạm chủ dành phần cho trạm tớ hoặc trạm tớ yêu cầu và ñược trạm chủ ñáp ứng. Tuy nhiên có thể thấy những vấn ñề sẽ gặp phải của giải pháp này là: chi phí cho việc thăm dò, ñộ trễ do phải chờ ñược phân lượt truyền, hệ thống rối loạn khi trạm chủ gặp sự cố.
- **Chuyển thẻ bài (token passing):** Thẻ bài ñiều khiển sẽ ñược chuyển lần lượt từ trạm này qua trạm kia. Trạm nào có trong tay thẻ bài sẽ ñược quyền truyền, truyền xong phải chuyển thẻ bài qua trạm kế tiếp. Những vấn ñề cần phải quan tâm: chi phí quản lý thẻ bài, ñộ trễ khi phải chờ thẻ bài, khó khăn khi thẻ bài bị mất.

Ví dụ về phương pháp thăm dò: Thăm dò phân tán (Distributed Polling)

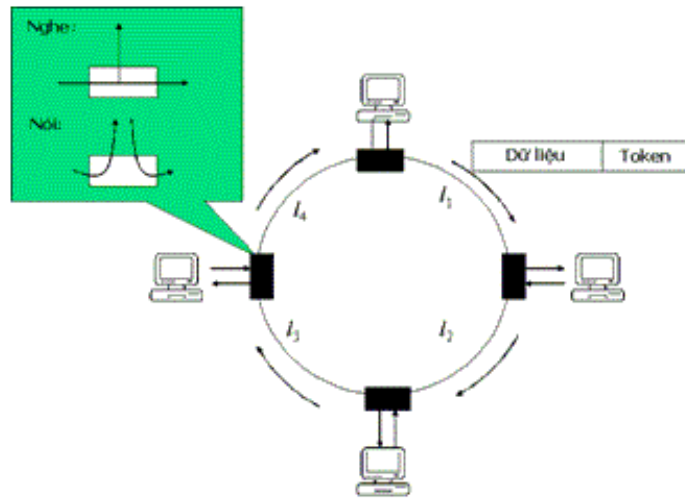
Thời gian ñược chia thành những “khe” (slot). Giả sử hệ thống hiện có N trạm làm việc. Một chu kỳ hoạt ñộng của hệ thống bắt ñầu bằng N khe thời gian ngắn ñùng ñể ñặt chỗ (reservation slot). Khe thời gian ñùng ñể ñặt chỗ bằng với thời gian lan truyền tín hiệu giữa hai ñầu mút xa nhất trên đường truyền. Tới khe ñặt chỗ thứ i, trạm thứ i nếu muốn truyền dữ liệu sẽ phát tín hiệu ñặt chỗ của mình lên kênh truyền, và tín hiệu này sẽ ñược nhìn thấy bởi tất cả các trạm khác trong mạng. Sau thời gian ñặt chỗ, các trạm bắt ñầu việc truyền dữ liệu của mình theo ñúng trình tự ñã ñăng ký.



Mô tả các chu kỳ hoạt ñộng của hệ thống Thăm dò phân tán (H5.14)

Ví dụ về phương pháp chuyển thẻ bài: Token Ring

Giao thức này sử dụng mạng kiểu hình vòng, ñùng thẻ bài ñể cấp quyền sử dụng đường truyền. Mạng token ring bao gồm một tập hợp các trạm ñược nối với nhau thành một vòng.



Mô hình hoạt động của mạng Token Ring (H5.15)

Dữ liệu luôn chạy theo một hướng vòng quanh vòng. Mỗi trạm nhận khung từ trạm phía trên của nó và rồi chuyển khung đến trạm phía dưới. Thẻ bài là công cụ để quyết định ai có quyền truyền tại một thời điểm.

Cách thức hoạt động của mạng token ring như sau: một thẻ bài, thực chất chỉ là một dãy bit, sẽ chạy vòng quanh vòng; mỗi nút sẽ nhận thẻ bài rồi lại chuyển tiếp thẻ bài này đi. Khi một trạm có khung cần truyền và đúng lúc nó thấy có thẻ bài tới, nó liền lấy thẻ bài này ra khỏi vòng (nghĩa là không có chuyển tiếp chuỗi bit đặc biệt này lên vòng nữa), và thay vào đó, nó sẽ truyền khung dữ liệu của mình đi. Khi khung dữ liệu đi một vòng và quay lại, trạm phát sẽ rút khung của mình ra và chèn lại thẻ bài vào vòng. Hoạt động cứ xoay vòng như thế.

Card mạng dùng cho token ring sẽ có trên đó một bộ nhận, một bộ phát và một bộ đệm dùng chứa dữ liệu. Khi không có trạm nào trong vòng có dữ liệu để truyền, thẻ bài sẽ lưu chuyển vòng quanh. Nếu một trạm có dữ liệu cần truyền và có thẻ bài, nó có quyền truyền một hoặc nhiều khung dữ liệu tùy theo qui định của hệ thống.

Mỗi khung dữ liệu được phát đi sẽ có một phần thông tin chứa địa chỉ đích của trạm bên nhận; ngoài ra nó còn có thể chứa địa chỉ multicast hoặc broadcast tùy theo việc bên gửi muốn gửi khung cho một nhóm người nhận hay tất cả mọi người trong vòng. Khi khung thông tin chạy qua mỗi trạm trong vòng, trạm này sẽ nhìn vào địa chỉ đích trong khung đó để biết xem có phải nó là đích đến của khung không. Nếu phải, trạm sẽ chép nội dung của khung vào trong bộ đệm của nó, chỉ chép thôi chứ không được xóa khung ra khỏi vòng.

Một vấn đề cần phải quan tâm đến là một trạm đang giữ thẻ bài thì nó có quyền truyền bao nhiêu dữ liệu, hay nói cách khác là trạm được cho bao nhiêu thời gian để truyền dữ liệu? Chúng ta gọi thời gian này là thời gian giữ thẻ bài – THT (Token Holding Time). Trong trường hợp trong vòng chỉ có một trạm cần truyền dữ liệu và các trạm khác không

có nhu cầu truyền, thì ta có thể cấp THT cho trạm có nhu cầu càng lâu càng tốt. Điều này sẽ làm tăng hiệu suất sử dụng hệ thống một cách đáng kể. Bởi vì sẽ thật là ngớ ngẩn nếu bắt trạm ngừng, chờ thẻ bài chạy hết một vòng, rồi lại truyền tiếp. Tuy nhiên, giải pháp trên sẽ không hoạt động tốt nếu có nhiều trạm trong vòng cần gửi dữ liệu. THT dài chỉ thích hợp cho những trạm cần truyền nhiều dữ liệu, nhưng lại không phù hợp với những trạm chỉ có ít thông điệp cần gửi đi ngay cả khi thông điệp này là tối quan trọng. Điều này cũng giống như tình huống mà bạn xếp hàng để sử dụng máy ATM ngay sau một anh chàng định rút ra 10 triệu đồng, trong khi bạn chỉ cần vào đây để kiểm tra tài khoản của mình còn bao nhiêu tiền! Trong các mạng 802.5, THT mặc định là 10 ms.

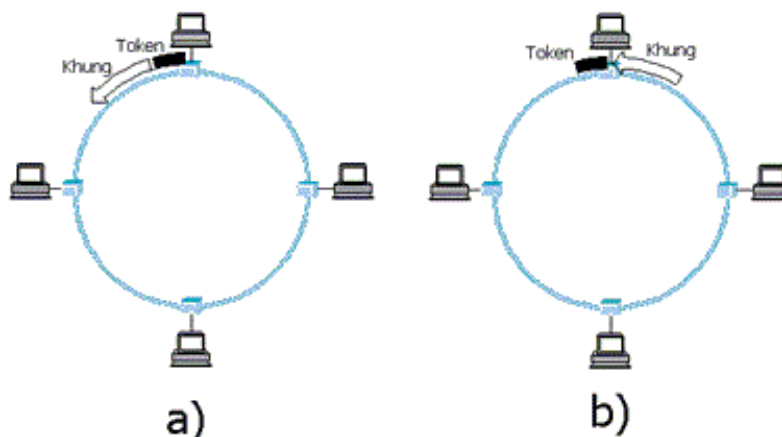
Từ thời gian giữ thẻ bài, chúng ta lại nghĩ ra một số đo quan trọng khác: Thời gian xoay vòng của thẻ bài – TRT (Token rotation time), nghĩa là lượng thời gian bỏ ra để thẻ bài đi hết đúng một vòng. Dễ nhận thấy rằng:

$$TRT \leq \text{Số nút hoạt động} \times THT + \text{Độ trễ của vòng}$$

Với “Độ trễ của vòng” là tổng thời gian để thẻ bài đi hết một vòng khi trong vòng không có trạm nào cần truyền dữ liệu, “Số nút hoạt động” ám chỉ số trạm có dữ liệu cần truyền.

Giao thức 802.5 cung cấp một phương thức truyền dữ liệu tin cậy bằng cách sử dụng hai bit A và C ở đuôi của khung dữ liệu. Hai bit này ban đầu nhận giá trị 0. Khi một trạm nhận ra nó là đích đến của một khung dữ liệu, nó sẽ đặt bit A trong khung này lên. Khi trạm chép khung vào trong bộ nhớ đệm của nó, nó sẽ đặt bit C lên. Khi trạm gửi thấy khung của nó quay lại với bit A vẫn là 0, nó biết là trạm đích bị hư hỏng hoặc không có mặt. Nếu bit A là 1, nhưng bit C là 0, điều này ám chỉ trạm đích có mặt nhưng vì lý do nào đó trạm đích không thể nhận khung (ví dụ như thiếu bộ đệm chẳng hạn). Vì thế khung này có thể sẽ được truyền lại sau đó với hy vọng là trạm đích có thể tiếp nhận nó.

Chi tiết cuối cùng cần phải xem xét là: chính xác khi nào thì trạm sẽ nhả thẻ bài ra? Có hai đề nghị: a) nhả thẻ bài ra ngay sau khi trạm vừa truyền khung xong (RAT); b) nhả thẻ bài ra ngay sau khi trạm nhận lại khung vừa phát ra (RAR).



Quản lý hoạt động của mạng Token Ring

Cần thiết phải đề cử ra một trạm làm nhiệm vụ quản lý mạng token ring gọi là monitor. Công việc của monitor là đảm bảo sức khỏe cho toàn bộ vòng. Bất kỳ trạm nào cũng có thể trở thành monitor. Thủ tục bầu chọn monitor diễn ra khi vòng vừa được tạo ra hoặc khi monitor của vòng bị sự cố. Một monitor mạnh khỏe sẽ định kỳ thông báo sự hiện diện của nó cho toàn vòng biết bằng một thông điệp đặc biệt. Nếu một trạm không nhận được thông báo hiện diện của monitor trong một khoảng thời gian nào đó, nó sẽ coi như monitor bị hỏng và sẽ cố trở thành monitor mới.

Khi một trạm quyết định rằng cần phải có một monitor mới, nó sẽ gửi một thông điệp thỉnh cầu, thông báo ý định trở thành monitor của mình. Nếu thông điệp này chạy một vòng và về lại được trạm, trạm sẽ cho rằng mọi người đồng ý vị trí monitor của nó. Còn nếu đồng thời có nhiều trạm cùng gửi thông điệp thỉnh cầu, chúng sẽ phải áp dụng một luật lựa chọn nào đó, chẳng hạn như “ai có địa chỉ cao nhất sẽ thắng cử”.

Nhiệm vụ đáng chú ý của monitor là phải đảm bảo rằng luôn luôn có sự hiện diện của thẻ bài ở đâu đó trên vòng, có thể là đang di chuyển hay đang bị giữ bởi một trạm nào đó. Rõ ràng là thẻ bài có thể bị biến mất vì lý do nào đó chẳng hạn như lỗi bit, trạm đang giữ nó bị hư hỏng. Để phát hiện ra việc thẻ bài bị mất, khi thẻ bài chạy ngang qua monitor, nó sẽ bật một bộ đếm thời gian để tính giờ. Bộ đếm này có giá trị tối đa là:

Số lượng trạm \times THT + Độ trễ của vòng

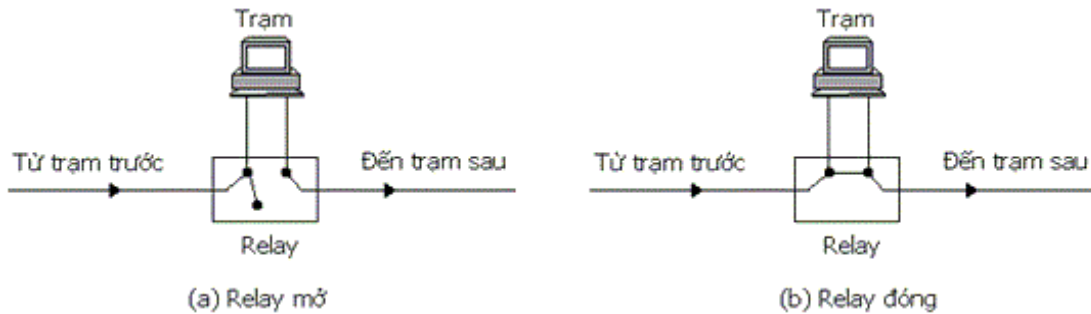
Trong đó “Số lượng trạm” là số các trạm làm việc đang hiện diện trên vòng, “độ trễ của vòng” là tổng thời gian lan truyền tín hiệu trên vòng. Nếu bộ đếm đạt đến giá trị tối đa mà monitor vẫn không thấy thẻ bài chạy qua nó nữa thì nó sẽ tạo ra thẻ bài mới.

Monitor cũng phải kiểm tra xem có khung nào bị hỏng hoặc vô thừa nhận hay không. Một khung nếu có lỗi checksum hoặc khuôn dạng không hợp lệ sẽ chạy một cách vô định trên vòng. Monitor sẽ thu khung này lại trước khi chèn lại thẻ bài vào vòng. Một khung vô thừa nhận là khung mà đã được chèn thành công vào vòng, nhưng cha của nó bị chết, nghĩa là trạm gửi nó chỉ gửi nó lên vòng, nhưng chưa kịp thu nó lại thì đã bị chết (down). Những khung như vậy sẽ bị phát hiện bằng cách thêm vào một bit điều khiển gọi là monitor bit. Khi được phát lần đầu tiên, monitor bit trên khung sẽ nhận giá trị 0. Khi khung đi ngang qua monitor, monitor sẽ đặt monitor bit lên 1. Nếu monitor thấy khung này lại chạy qua nó với monitor bit là 1, nó sẽ rút khung này ra khỏi vòng.

Một chức năng quản lý vòng khác là phát hiện ra một trạm bị chết. Nếu một trạm trong vòng bị chết, nó sẽ làm đứt vòng. Để tránh tình trạng này người ta thêm vào trạm một rờ-le điện tử (relay). Khi trạm còn mạnh khỏe, rờ-le sẽ mở và trạm được nối với vành,

Lớp con MAC (Media Access Control Sublayer)

khi trạm bị chết và ngưng không cung cấp năng lượng cho rờ-le, rờ-le sẽ tự động đóng mạch và bỏ qua trạm này.



Sử dụng relay để tránh đứt vòng (H5.17)

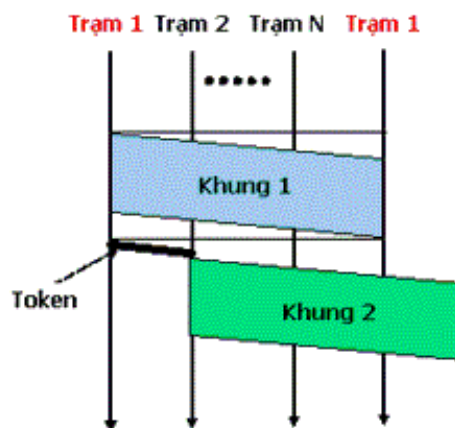
Khi monitor nghi ngờ một trạm bị chết, nó sẽ gửi đến trạm đó một khung đặc biệt gọi là khung beacon. Nếu không nhận được trả lời thích đáng, monitor sẽ coi trạm đó đã chết.

ĐÁNH GIÁ VỀ MẠNG TOKEN RING:

Ta sẽ khảo sát hai kiểu chuyển thẻ bài: Release After Reception (RAR) và Release After Transmissions (RAT).

RAR: Nhà thẻ bài sau khi nhận lại dữ liệu

Sau khi một trạm phát đi khung dữ liệu của nó, trạm sẽ chờ đến khi khung này quay trở lại mới chuyển thẻ bài cho trạm kế tiếp. Mạng IEEE 802.5 Token Ring (16Mbps) sử dụng cơ chế này.



Mô phỏng cơ chế chuyển thẻ bài trong RAR (H5.18)

Ta gọi hiệu suất truyền khung là h_{RAR} . Mạng kiểu RAR sẽ đạt được hiệu suất tối đa nếu một trạm phát liên tục.

Lớp con MAC (Media Access Control Sublayer)

Đặt:

T_{prop} là thời gian lan truyền tín hiệu giữa hai đầu mút xa nhau nhất trên đường truyền tải.

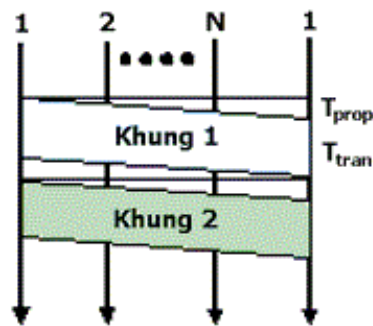
T_{tran} là thời gian để phát hết một khung dữ liệu lên đường truyền.

P là kích thước của khung dữ liệu, ví dụ 1000 bits.

C là dung lượng của đường truyền, ví dụ 10 Mbps.

$$\text{Có } T_{tran} = \frac{P}{C}$$

Sau đây là biểu đồ mô phỏng mối liên quan giữa thời gian phát khung và thời gian truyền tín hiệu:



Thời gian lan truyền và phát khung với RAR (H5.19)

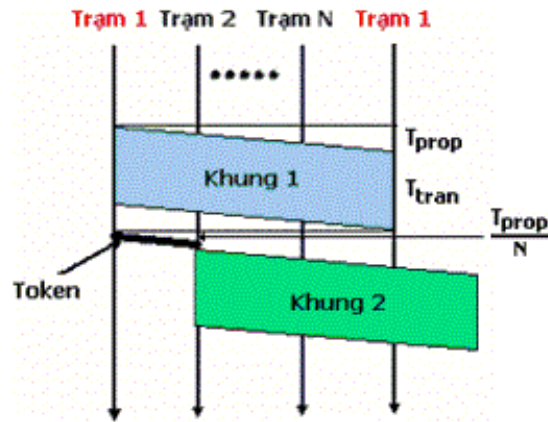
Có thể thấy:

$$\eta_{RAR} = \frac{1}{1+a}$$

Với

$$a = \frac{T_{prop}}{T_{tran}}$$

Trong trường hợp một trạm *luôn phải nhường token* ngay sau khi token đi hết một vòng và quay trở lại nó thì hiện trạng phân bố thời gian được vẽ ra dưới đây:



Các khoảng thời gian một trạm phải trải qua khi gửi dữ liệu và chuyển thẻ bài (H5.20)

Giả sử token có kích thước quá nhỏ để coi như thời gian phát nó bằng không, mạng có N trạm làm việc và khoảng cách giữa các trạm là bằng nhau. Vì vậy, thời gian lan truyền tín hiệu từ một trạm đến một trạm liền kề nó là T_{prop}/N .

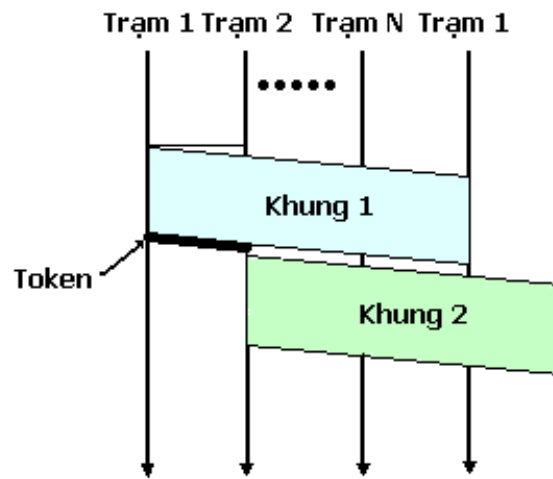
Có thể nhận thấy thời gian để token chuyển từ một trạm sang một trạm kề nó là T_{prop}/N .

Vì vậy:

$$\eta_{RAR} = \frac{T_{tran}}{T_{prop} + T_{tran} + \frac{T_{prop}}{N}} = \frac{1}{1 + \left(\frac{N+1}{N}\right) \frac{T_{prop}}{T_{tran}}} \approx \frac{1}{1+a}$$

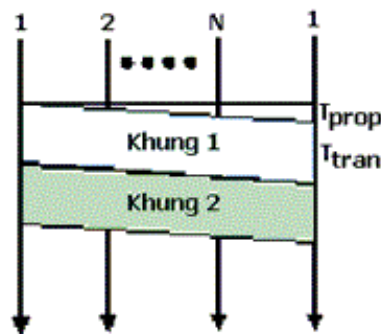
RAT: Nhả thẻ bài ngay sau khi truyền dữ liệu

Với kỹ thuật RAT, trạm sẽ chuyển thẻ bài điều khiển cho trạm kế tiếp ngay sau khi nó vừa phát xong khung dữ liệu. Ví dụ mạng FDDI (Fiber Distributed Data Interface - 100Mbps) sử dụng kỹ thuật này.



Mô phỏng kỹ thuật RAT (H5.21)

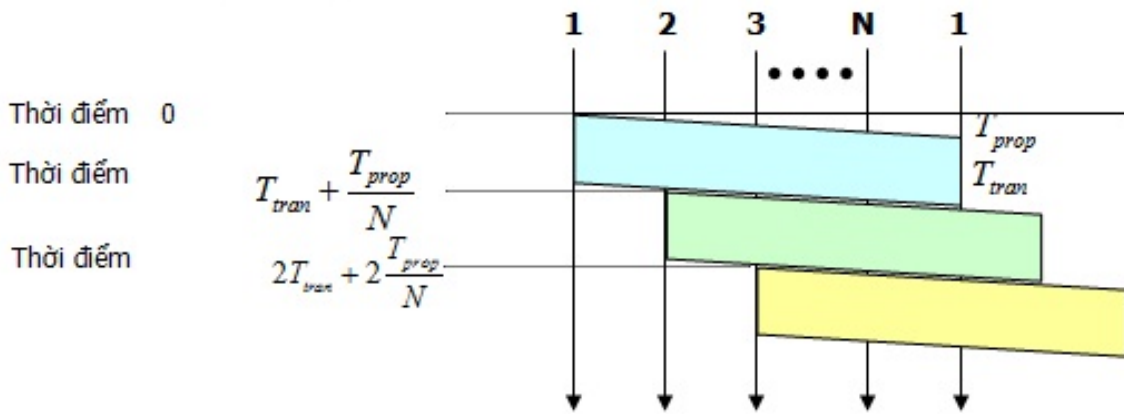
Gọi hiệu suất truyền khung là *****SORRY, THIS MEDIA TYPE IS NOT SUPPORTED.***** . Với kỹ thuật RAT, hiệu suất đạt tối đa khi một trạm *liên tục truyền khung*. Sau đây là biểu đồ thời gian mô phỏng:



Thời gian phát và lan truyền khung (H5.22)

Có thể thấy khi một trạm phát khung liên tục thì $\eta_{RAT} = 1$

Tuy nhiên khi một trạm buộc *phải nhả token ngay sau khi nó vừa phát dữ liệu xong*, thì biểu đồ thời gian có khác:



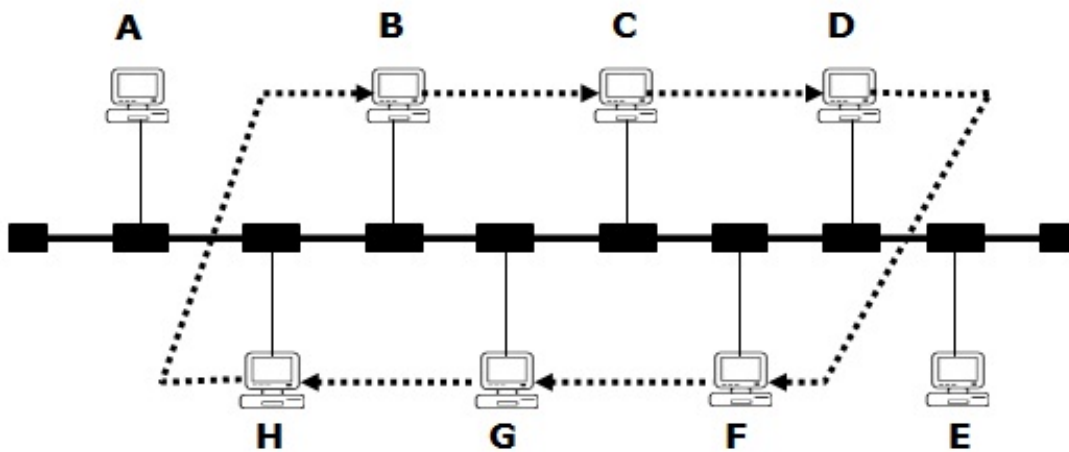
Khi một trạm phải nhả token ngay sau khi nó vừa phát dữ liệu (H5.23)

Ta tính lại hiệu suất như sau:

$$\eta_{RAT} = \frac{T_{tran}}{T_{tran} + \frac{T_{prop}}{N}} = \frac{1}{1 + \frac{1}{N} \frac{T_{prop}}{T_{tran}}} = \frac{1}{1 + \frac{a}{N}}$$

Với
$$a = \frac{T_{prop}}{T_{tran}}$$

Ví dụ về phương pháp chuyển thẻ bài: Token Bus



Sơ đồ mạng Token Bus (H5.24)

Kỹ thuật Token Bus về bản chất là sử dụng mạng hình bus. Tuy nhiên người ta muốn thiết lập một vòng ảo trên đó để nó hoạt động giống như Token Ring. Nguyên tắc hoạt

động như sau: trạm có nhu cầu truyền dữ liệu thì sẽ tham gia vào vòng ảo, ngược lại thì sẽ nằm ngoài và chỉ nghe thôi!

Giải thuật bổ sung một trạm vào vòng:

- Mỗi trạm trong vòng có trách nhiệm định kỳ tạo điều kiện cho các trạm khác tham gia vào vòng.
- Trước khi chuyển thẻ bài đi, trạm sẽ gửi thông báo “tìm trạm đứng sau” (có địa chỉ giữa nó và trạm đứng liền kề hiện tại).
- Nếu sau một thời gian xác định mà vẫn không có yêu cầu gia nhập nào, trạm sẽ chuyển thẻ bài đến trạm kế tiếp như thường lệ.
- Nếu có yêu cầu gia nhập vòng, thì trạm sẽ ghi nhận trạm mới yêu cầu là trạm kế tiếp của nó và sẽ chuyển thẻ bài tới trạm kế mới này.

Giải thuật rút lui ra khỏi vòng:

- Khi muốn rút ra khỏi vòng, trạm sẽ chờ đến khi nó có token, sau đó sẽ gửi yêu cầu “nổi trạm đứng sau” tới trạm đứng trước nó, yêu cầu trạm đứng trước nổi trực tiếp với trạm đứng liền sau nó.

Ngoài ra còn phải quan tâm đến tình trạng mất thẻ bài, các trạm thành viên trong vòng bị hư hỏng.