

Điều khiển luồng và tránh tắc nghẽn theo phương pháp cửa sổ



1

Giới thiệu

- Cơ chế điều khiển luồng và chống tắc nghẽn dựa trên phương pháp cửa sổ được thực hiện bởi việc **giới hạn số lượng gói tin được truyền ở phía phát** nhằm **đảm bảo thông tin** này không vượt quá khả năng xử lý của phía thu.



2

Giới thiệu(2)

- Theo cơ chế này, **phía phát sẽ không thực hiện phát tin** chừng nào phía thu còn chưa xử lý xong gói tin (hoặc một số gói tin) trước đó.
- Khi phía thu xử lý xong thông tin do phía phát gửi đến thì nó sẽ báo cho phía phát biết và lúc này, phía phát sẽ tiếp tục gửi các gói tin tiếp theo. Cơ chế này **đảm bảo việc truyền tin không bao giờ vượt quá khả năng xử lý của phía thu.**



3

Giới thiệu(3)

- Với việc kết hợp hoạt động nhịp nhàng giữa phía phát và phía thu (có sử dụng báo nhận), số lượng gói tin đồng thời tồn tại trên đường truyền nằm trong giới hạn nhất định.
- Nếu phía thu có bộ đệm với **dung lượng lớn** hơn tổng kích thước các gói tin này thì bộ đệm phía thu sẽ không bao giờ **bị tràn.**
- .



4

Điều khiển luồng theo cửa sổ (Window Flow Control)



5

Giới thiệu

- Phương pháp điều khiển luồng theo cửa sổ trượt là phương pháp được sử dụng phổ biến nhất ở thời điểm hiện tại.
 - end-to-end (điều khiển luồng giữa điểm phát và điểm thu trong mạng)
 - hop-by-hop (điều khiển luồng giữa hai nút mạng liên tiếp).



6

Cửa sổ End-to-End

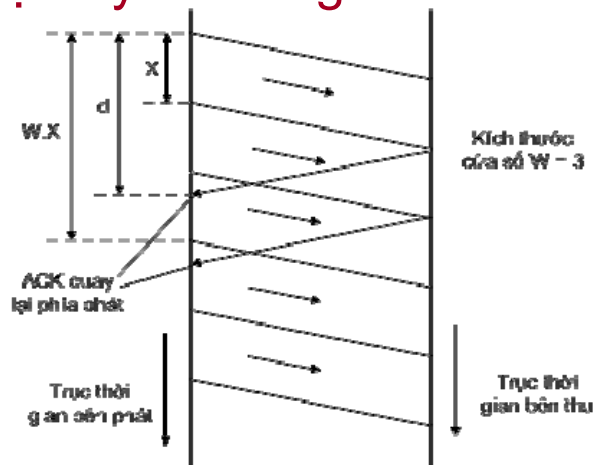
- Phương pháp điều khiển luồng theo cửa sổ dựa trên cơ sở phương pháp **cửa sổ trượt** làm việc tại lớp liên kết dữ liệu. Các khung thông tin từ phát sang thu và khung báo nhận, báo lỗi truyền từ thu sang phát được đánh số thứ tự để phân biệt, kích thước cửa sổ $W < 2^k$ với k là số bit dùng đánh số phân biệt các khung



7

Quan hệ giữa kích thước cửa sổ và tốc độ truyền thông tin

X là thời gian phát một khung thông tin,
 W là kích thước cửa sổ
 d là tổng trễ từ phát đến thu (dùng cho khung thông tin) và từ thu đến phát (dùng cho báo nhận), round-trip delay.

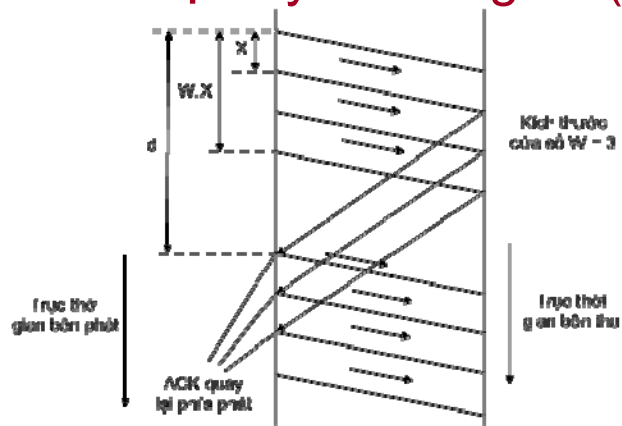


- Khi $W.x > d$ thì phía phát truyền liên tục



8

Quan hệ giữa kích thước cửa sổ và tốc độ truyền thông tin (2)



- Khi $W.x \leq d$



9

Nhận xét

- Khi $d \leq W.X$: Phía phát có thể truyền thông tin liên tục mà không cần phải dừng lại đợi. Tốc độ phát thông tin $r = 1/X$ và trong trường hợp này, điều khiển luồng không có ý nghĩa (vì phía phát có thể phát tin với tốc độ cao nhất mà không bị hạn chế)



10

Nhận xét (2)

- trường hợp $d > W.X$,
 - →vai trò của điều khiển luồng. Phía phát thực hiện phát W khung thông tin sau đó dừng lại chờ báo nhận ở phía thu, rồi mới được phát tiếp.
 - Nói một cách khác, lượng thông tin đến phía thu (hay lượng thông tin đi vào mạng) đã bị hạn chế nhỏ hơn khả năng phát cực đại của phía phát.



11

Nhận xét (3)

- trường hợp $d > W.X$, (cont.)
 - Điều này xảy ra khi round-trip delay lớn nên khi phía phát thực hiện phát xong W gói tin rồi nhưng báo nhận đầu tiên vẫn chưa quay trở lại. Lúc này phía phát phải ngừng phát và chờ báo nhận.
 - Nếu phía phát luôn có thông tin để phát thì tốc độ phát tin trung bình sẽ là W/d gói/s



12

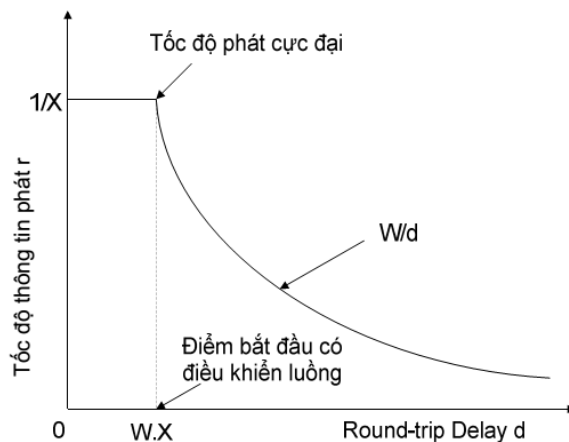
Nhận xét (4)

- Kết hợp cả hai trường hợp ta tính được tốc độ phát tin cực đại khi kể đến round-trip delay sẽ là $r = \left\{ \frac{1}{X}, \frac{W}{d} \right\}$
- Khi d tăng (có tắc nghẽn), điều khiển luồng sẽ thực hiện vai trò của nó và giới hạn tốc độ truyền tin
- Khi không có tắc nghẽn xảy ra, d giảm và r tăng lên



13

Nhận xét (5)



14

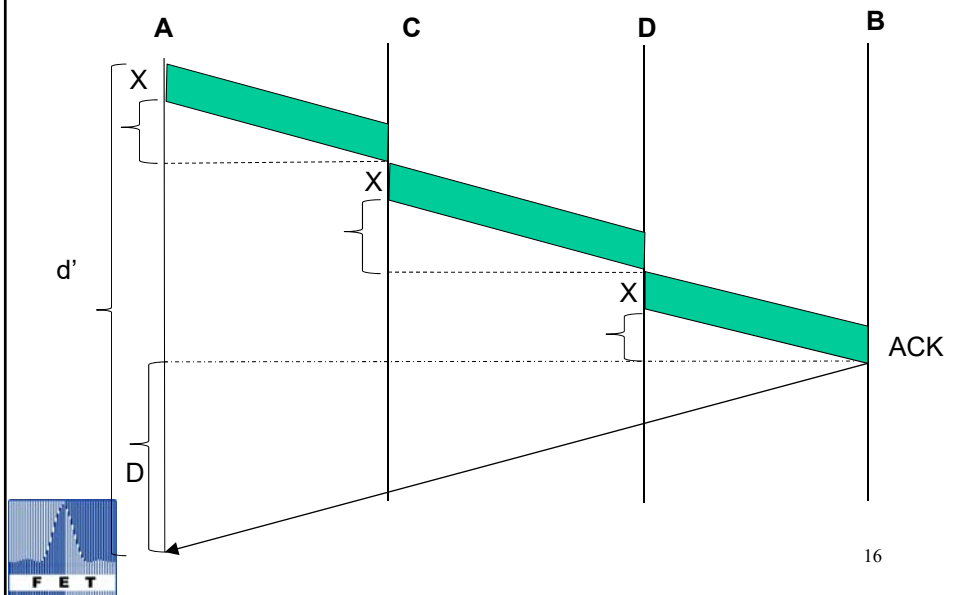
Kích thước cửa sổ

- Trong trường hợp không có tắc nghẽn xảy ra, kích thước cửa sổ được chọn đủ lớn để đảm bảo tốc độ truyền thông tin đạt $r = 1/X$ gói/s.
- d' = round-trip delay khi trễ hàng đợi xấp xỉ 0 (không có tắc nghẽn) – đây là trễ tính từ lúc phát gói thông tin ở bên phát và nhận ACK từ phía thu
- N = số nút trung gian dọc theo đường truyền từ phát đến thu
- D = trễ truyền sóng dọc theo đường truyền



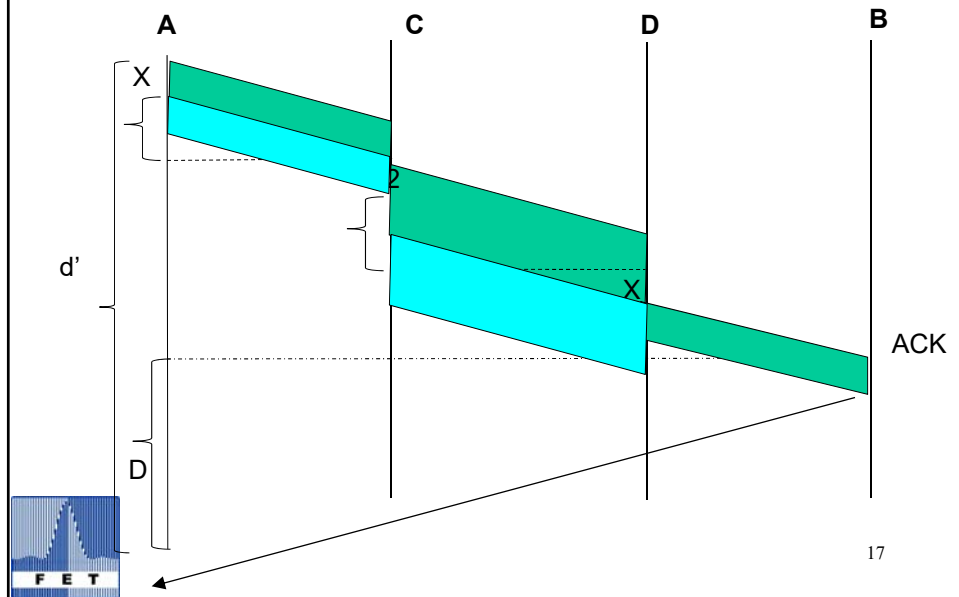
15

Kích thước cửa sổ (2)



16

Kích thước cửa sổ (2)



Kích thước cửa sổ (3)

- $d' = X + N.X + 2.D$
- Để đảm bảo tốc độ truyền tin tối đa cần đảm bảo $W.X \geq d'$ hay $W \geq (N+1) + 2.D/X$. Ta nhận thấy:
- Khi $D < X$ thì $W \approx N+1$ - kích thước cửa sổ không phụ thuộc vào trễ truyền sóng.
- Khi $D \gg X$ thì $W \approx 2.D/X$ – kích thước cửa sổ không phụ thuộc vào chiều dài đường đi.



18

Kích thước cửa sổ (4)

- Trong trường hợp có tắc nghẽn xảy ra, thì trễ round-trip $d > d'$ (d bao gồm cả trễ hàng đợi do tắc nghẽn)



19

Hạn chế end-to-end window

- Khó đảm bảo trễ nằm trong giới hạn cho phép khi lưu lượng vào mạng tăng

Giả sử trong mạng có n tiến trình điều khiển luồng với kích thước cửa sổ tương ứng là W_1, W_2, \dots, W_n . Lúc này, tổng số gói tin trong mạng sẽ là $\sum_{i=1}^n \beta_i \cdot W_i$ trong đó β_i là một hệ số trong khoảng 0 đến 1 phụ thuộc vào thời gian trễ của ACK. Theo định luật Little's thì trễ trung bình của

gói tin trong mạng sẽ là $T = \frac{\sum_{i=1}^n \beta_i \cdot W_i}{\lambda}$ trong đó λ là thông lượng.



20

Hạn chế end-to-end window(2)

- Khó đảm bảo trễ nằm trong giới hạn cho phép khi lưu lượng vào mạng tăng (cont.)
 - Khi số lượng các tiến trình cần điều khiển luồng tăng lên (n tăng) thì λ tiến đến giá trị cực đại là tốc độ của các đường liên kết, là giá trị không đổi (giá trị này phụ thuộc vào mạng, vị trí của điểm phát và thu cũng như giải thuật định tuyến).
 - Như vậy giá trị trễ T sẽ tăng tỷ lệ với số lượng tiến trình được điều khiển luồng (chính xác ra là kích thước cửa sổ của chúng). Nếu số lượng tiến trình là rất lớn thì hệ thống mạng không đảm bảo giữ giá trị T nằm trong một giới hạn nhất định và



21

Hạn chế end-to-end window(3)

- Khó đảm bảo trễ nằm trong giới hạn cho phép khi lưu lượng vào mạng tăng (cont.)
 - Một giải pháp có thể sử dụng là giảm kích thước cửa sổ để có thể giảm trễ khi mạng hoạt động ở tình trạng nặng tải (có thể xảy ra tắc nghẽn). Giải pháp này có thể áp dụng ở một mức độ nào đó tuy nhiên nó nếu giá trị này quá nhỏ thì việc truyền thông tin lại không hiệu quả.



22

Hạn chế end-to-end window (4)

- Khó đảm bảo trễ nằm trong giới hạn cho phép khi lưu lượng vào mạng tăng (cont.)
 - Trên thực tế, người ta sử dụng phương pháp cửa sổ thích ứng (adaptive window) để thực hiện truyền tin. Trong phương pháp này, kích thước cửa sổ có thể thay đổi tùy thuộc tình trạng của mạng. Trong trường hợp mạng ít tải, kích thước cửa sổ có thể lớn để cho phép truyền thông tin với tốc độ cao. Khi tải trên mạng tăng, kích thước cửa sổ được giảm đi nhằm tránh tắc nghẽn.



23

Hạn chế end-to-end window(5)

- Khó đảm bảo tính công bằng cho tất cả người dùng.
 - Để đảm bảo truyền tin tốt nhất cho một kết nối, kích thước cửa sổ tỷ lệ với số nút mạng trên đường đi từ nguồn đến đích cũng như tỷ lệ với trễ truyền sóng dọc theo đường truyền (cũng phụ thuộc vào khoảng cách).
 - Như vậy, trong trường hợp có tắc nghẽn, nếu trên một đường truyền có nhiều kết nối cùng hoạt động thì kết nối nào có khoảng cách nguồn – đích lớn sẽ được sử dụng tài nguyên nhiều hơn (do kích thước cửa sổ lớn hơn và số lượng gói tin đến nút đó và được chấp nhận sẽ nhiều hơn).



Hạn chế end-to-end window(5)

- Khó đảm bảo tính công bằng cho tất cả người dùng (cont.)
 - Để đảm bảo được tính công bằng, người ta dùng cơ chế round-robin (xử lý vòng) cho tất cả các kết nối cùng sử dụng tài nguyên của một nút mạng. Lúc này, các kết nối được coi như có độ ưu tiên như nhau và được xử lý luân phiên dựa theo kết nối chứ không dựa trên tỷ lệ gói tin đến.



25

Cửa sổ Hop-by-Hop

- Việc điều khiển luồng được thực hiện giữa hai nút mạng kế tiếp trên đường truyền.
- Mỗi nút mạng có các cửa sổ độc lập dùng cho các kênh làm việc khác nhau (kênh ảo). Nguyên tắc hoạt động của cơ chế này tương tự như điều khiển luồng kiểu end-to-end nhưng chỉ áp dụng cho một chặng.



26

Cửa sổ Hop-by-Hop (2)

- Trong trường hợp truyền thông tin cục bộ không quá xa (với đa phần các cơ chế truyền tin, trừ thông tin vệ tinh) kích thước cửa sổ thường là 2 hoặc 3 (do số nút mạng thông tin phải đi qua là 1, trễ truyền sóng không đáng kể).



27

Cửa sổ Hop-by-Hop (3)

- Gọi nút có thông tin cần truyền là nút nguồn, nút có nhận thông tin là nút đích (các nút dọc trên đường truyền, và có thể bao gồm cả phía phát và phía thu).
- Mục đích chính của điều khiển luồng hop-by-hop là đảm bảo bộ đệm của nút đích không bị quá tải bởi quá nhiều gói tin đến (như trong trường hợp end-to-end). Điều này được thực hiện với việc nút đích giảm tốc độ gửi ACK về cho nút nguồn.



28

Cửa sổ Hop-by-Hop (4)

- Trong trường hợp tổng quát, nút đích có bộ đệm với dung lượng W gói cho mỗi liên kết và nó sẽ gửi ACK cho nút nguồn nếu trong bộ đệm còn chỗ trống. Nút đích sẽ xóa gói tin trong bộ đệm nếu nó đã được truyền thành công đến nút kế tiếp trên đường truyền hay đã đi ra khỏi mạng



29

Cửa sổ Hop-by-Hop (5)

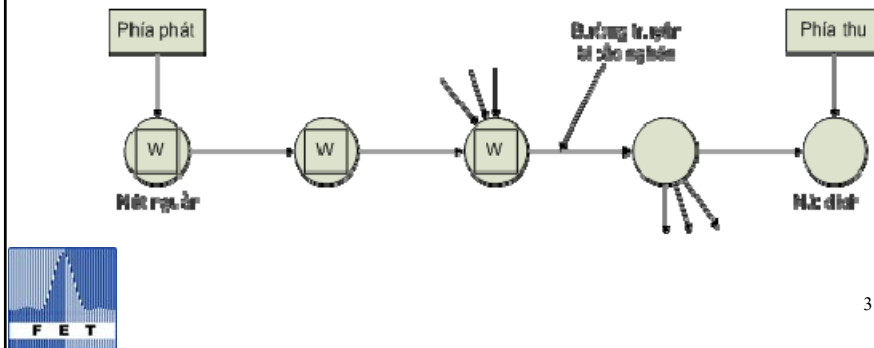
- Giả sử có ba nút liên tiếp trên mạng là $(i-1, i, i+1)$. Giả sử bộ đệm của i đã bị đầy với W gói tin. Nút i sẽ gửi ACK cho nút $i-1$ nếu nó đã gửi thành công một gói tin cho nút $i+1$ (lúc đó bộ đệm của nút i mới được giải phóng và có chỗ cho một gói tin). Nút i thực hiện được điều này nếu nó nhận được một ACK từ nút $i+1$.



30

Cửa sổ Hop-by-Hop (6)

- Trong trường hợp có tắc nghẽn xảy ra tại một nút nào đó, bộ đệm của nút này bị đầy bởi W gói tin và theo hệ quả, bộ đệm của các nút phía trước nút đó cũng sẽ dần dần bị đầy. Hiện tượng này được gọi là **backpressure**



31

Ưu điểm của phương pháp hop-by-hop

- Dung lượng bộ đệm cần thiết ở mỗi nút sẽ nhỏ hơn trường hợp end-to-end rất nhiều.
 - Trong trường hợp xấu nhất, giả sử tắc nghẽn xảy ra tại đường nối cuối cùng của tuyến truyền (đường nối thứ n) thì tổng số gói tin nằm trong mạng sẽ là $n.W$ (bộ đệm của mỗi nút sẽ bị điền đầy bởi W gói tin).
 - Trong trường hợp này, số lượng gói tin sẽ được phân bố đều ở bộ đệm của các nút và do đó dung lượng bộ đệm cần thiết ở mỗi nút sẽ nhỏ hơn trường hợp end-to-end rất nhiều (trong trường hợp end-to-end, nếu tổng số gói tin vào mạng, hay kích thước cửa sổ, là $n.W$ thì dung lượng bộ đệm tương ứng ở mỗi nút cũng phải là $n.W$).



32

Ưu điểm của phương pháp hop-by-hop (2)

- Cho phép thực hiện tính công bằng.
 - Với việc phân các gói tin của một kết nối dọc theo các nút mạng mà kết nối phải đi qua, ta có thể tránh được tình trạng ở tại một nút, kết nối với khoảng cách nguồn – đích lớn sẽ chiếm hết tài nguyên của các kết nối khác.



33

Ưu điểm của phương pháp hop-by-hop (3)

- Cho phép thực hiện tính công bằng (cont.).
 - Trong trường hợp hop-by-hop, kích thước cửa sổ của các kết nối là xấp xỉ bằng nhau do đó tốc độ thông tin đến là không chênh lệch và việc sử dụng tài nguyên được đảm bảo công bằng.
 - Điều này không đúng trong trường hợp kết nối giữa hai nút dùng cho truyền vệ tinh. Trong trường hợp này, do trễ truyền dẫn khá lớn nên kích thước cửa sổ của kết nối vệ tinh có thể lớn hơn kích thước cửa sổ của các kết nối khác dẫn đến tình trạng không công bằng.



34

Phương thức Isarithmic

- Phương thức này cũng được coi là một **biến thể** của cơ chế điều khiển luồng theo cửa sổ với một **cửa sổ duy nhất** được dùng cho toàn mạng.
- Việc điều khiển luồng được thực hiện bởi việc **giới hạn số lượng gói tin đi vào mạng** thông qua việc **cấp phát một số lượng hạn chế thẻ bài**.



35

Phương thức Isarithmic (2)

- Mỗi một gói tin muốn đi vào mạng cần phải **nhận được một thẻ bài ở nút mà gói tin đó vào và trả lại thẻ bài ở nút mà gói tin đó ra khỏi mạng**. Như vậy, tổng số gói tin tồn tại đồng thời trong mạng luôn nhỏ hơn hoặc bằng tổng số lượng thẻ bài, và việc điều khiển luồng được thực hiện.



36

Hạn chế của phương thức Isarithmic

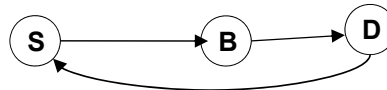
- Không đảm bảo tính công bằng cho tất cả người dùng:
 - Không có những cơ chế nhất định để quản lý và phân phối thẻ bài.
 - Ngoài ra, các thẻ bài có thể bị mất vì những lý do nhất định mà hiện tại chưa có cơ chế để quản lý số lượng thẻ bài tồn tại trong mạng.
 - Vì những lý do đó, phương thức Isarithmic ít được sử dụng trong thực tế.



37

Ví dụ 1

- Nút **S** gửi các gói tin đến D sử dụng cơ chế điều khiển luồng cửa sổ trượt *Go Back N*, kích thước mỗi gói là 1000byte. Trễ truyền lan trên các liên kết (**S, B**), (**B, D**) và (**D, S**) lần lượt là lần lượt là 3ms, 4ms, 5ms và tốc độ các đường truyền là 10Mbps. Giả sử kích thước gói tin ACK bằng 0
 - Tính kích thước cửa sổ nhỏ nhất để tốc độ truyền đạt lớn nhất khi truyền không lỗi
 - Với kích thước cửa sổ bằng 3. Tỷ lệ lỗi bit BER= 10^{-5} . Hãy tính tốc độ đường truyền đạt được trên liên kết. Giả sử gói phản hồi không bị lỗi.

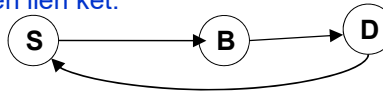


38



Ví dụ 1(2)

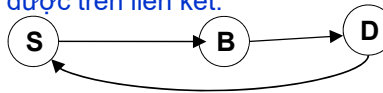
- Nút **S** gửi các gói tin đến D, Go Back N, kích thước mỗi gói là 1000byte. Trễ truyền lan trên các liên kết (**S, D**), (**D, B**) và (**B,S**) lần lượt là lần lượt là 3ms, 4ms, 5ms và tốc độ các đường truyền là 10Mbps
- Tính kích thước cửa sổ để tốc độ truyền đạt lớn nhất. Tính tốc độ đường truyền đạt được trên liên kết.
- $T_F = \frac{1000 \cdot 8}{10 \cdot 10^6} = 0.8(\text{ms})$
- Để tốc độ đạt cực đại thì $W \cdot T_F \geq T_\Sigma$.
- $T_\Sigma = ???$
- Tỷ lệ lỗi bit $\text{BER} = 10^{-5} \rightarrow q = (1 - \text{BER})^L \rightarrow p = 1 - q$



39

Ví dụ 2(2)

- Nút **S** gửi các gói tin đến D, Go Back N, kích thước mỗi gói là 1000byte. Trễ truyền lan trên các liên kết (**S, D**), (**D, B**) và (**B,S**) lần lượt là lần lượt là 3ms, 4ms, 5ms và tốc độ các đường truyền là (S,B) 10 Mbps, (B,D) 5 Mbps, (D,S) 5 Mbps, ACK = 500 byte
- Tính kích thước cửa sổ nhỏ nhất để tốc độ truyền đạt lớn nhất. Tính tốc độ đường truyền đạt được trên liên kết.
- $T_{F1} = \frac{1000 \cdot 8}{10 \cdot 10^6} = 0.8(\text{ms})$
- Để tốc độ đạt cực đại thì $W \cdot T_F \geq T_\Sigma$.
- $T_\Sigma = ???$



40

Kích thước cửa sổ (2)

The diagram illustrates the window size (Kích thước cửa sổ) in a communication protocol. It shows a sequence of frames (green and cyan) being transmitted from a sender (S) to a receiver (D) through a buffer (B). The receiver's buffer size is labeled d' . The frames are labeled X, X2, and X. The receiver's buffer is divided into segments labeled S, B, and D. The frames are shown in the buffer at different stages of reception. The frames are labeled X, X2, and X. The receiver's buffer is labeled d' . The frames are shown in the buffer at different stages of reception. The frames are labeled X, X2, and X. The receiver's buffer is labeled d' .

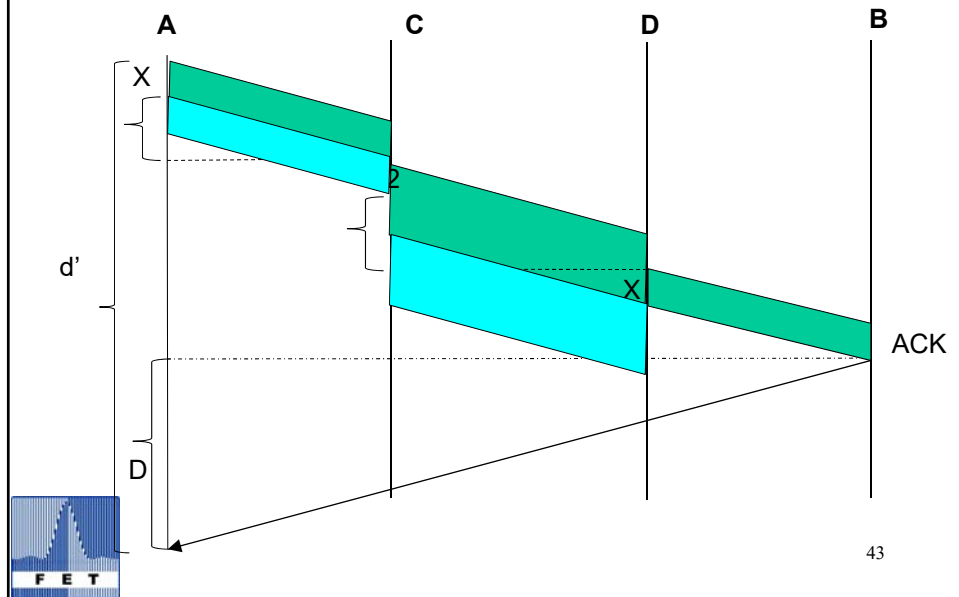
ACK

F E T

41

Điều khiển tắc nghẽn sử dụng cửa sổ thích ứng (adaptive window)

Kích thước cửa sổ (2)



Giới thiệu

- Bên cạnh việc sử dụng cơ chế cửa sổ để thực hiện điều khiển luồng, người ta có thể sử dụng cơ chế cửa sổ để thực hiện điều khiển và tránh tắc nghẽn ở trong mạng.
- Khi mạng có khả năng mang thông tin của người dùng, kích thước cửa sổ sẽ được đặt ở một mức nào đó.



Giới thiệu (2)

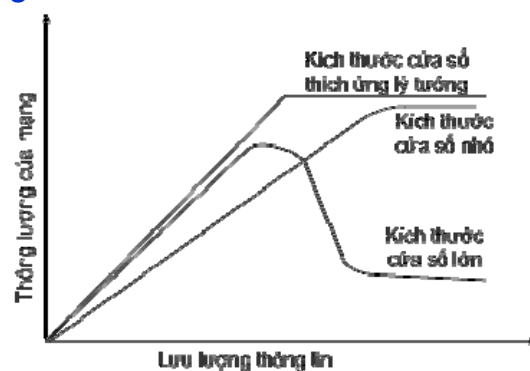
- Khi mạng nặng tải và có tắc nghẽn xảy ra, phía phát sẽ giảm kích thước cửa sổ để giảm số lượng gói tin đi vào mạng, do đó, thực hiện chức năng điều khiển tắc nghẽn cho mạng. **Kích thước cửa sổ chính** là nhân tố **quyết định tốc độ thông tin** từ phía phát đi vào mạng.



45

Giới thiệu (3)

- *Quan hệ giữa kích thước cửa sổ và lưu lượng mạng*

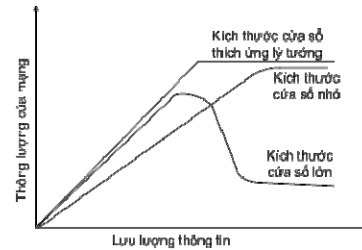


46

Giới thiệu (4)

Khi lưu lượng vào mạng nhỏ, kích thước cửa sổ lớn tỏ ra tối ưu do tận dụng được thời gian truyền gói tin.

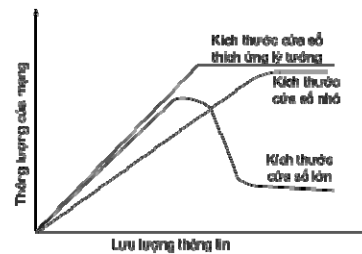
Tuy nhiên, khi lưu lượng vào mạng tăng lên, việc sử dụng kích thước cửa sổ lớn sẽ gây ra **tắc nghẽn** do có quá nhiều gói tin có thể **được gửi cùng lúc vào mạng**. Trong trường hợp này, người ta sử dụng các cửa sổ có kích thước nhỏ để đáp ứng với tình trạng của mạng.



47

Giới thiệu (5)

Việc thay đổi kích thước cửa sổ một cách **mềm dẻo** cho phù hợp với tình trạng lưu lượng của mạng chính là cách thức điều khiển **tắc nghẽn** của các thiết bị đầu cuối (phía phát và phía thu). Cơ chế thay đổi kích thước cửa sổ theo trình trạng lưu lượng mạng được gọi là **cơ chế cửa sổ thích ứng** (adaptive window).



48

Điều kiện quyết định thay đổi kích thước cửa sổ

- Phía phát dựa trên các thông tin phản hồi từ phía thu hoặc các thiết bị trên đường truyền từ phát đến thu để thực hiện điều chỉnh kích thước cửa sổ.



49

Phân loại thiết bị mạng trung gian

- Thiết bị mạng thông minh (active intermediate system)
 - Có khả năng phát hiện tắc nghẽn đang xảy ra hoặc có thể xảy ra và có khả năng thông báo cho phía phát
- Thiết bị mạng không thông minh (passive intermediate system)
 - Không có khả năng phát hiện tắc nghẽn, việc xác định tình trạng tắc nghẽn hoàn toàn được thực hiện bởi phía phát.



50

Thiết bị mạng thông minh

- Kỹ thuật điều khiển tắc nghẽn sử dụng thiết bị mạng thông minh hoạt động như sau:
 - Thiết bị mạng phát hiện tình trạng tắc nghẽn xảy ra hoặc sắp xảy ra (ví dụ: dung lượng bộ đệm vượt quá một ngưỡng nào đó)
 - Khi phát hiện tắc nghẽn, thiết bị mạng thông báo cho tất cả các nút nguồn (phía phát) thực hiện phát thông tin qua thiết bị mạng này



51

Thiết bị mạng thông minh (2)

- Kỹ thuật điều khiển tắc nghẽn sử dụng thiết bị mạng thông minh hoạt động như sau (cont.):
 - Các nút nguồn thực hiện giảm kích thước cửa sổ để giảm tắc nghẽn (với việc giảm kích thước cửa sổ, phía phát giảm số lượng gói tin có thể đi vào mạng)
 - Các nút nguồn có thể tăng kích thước cửa sổ nếu chúng xác định được rằng tình trạng tắc nghẽn đã được giải quyết.



52

Thiết bị mạng thông minh (3)

- Để có thể cảnh báo cho phía phát, nút mạng có thể sử dụng một trong hai cơ chế:
 - Sử dụng một gói tin cảnh báo độc lập – phương pháp này cho phép phía phát nhanh chóng nhận được thông tin tắc nghẽn và phản ứng kịp thời. Tuy nhiên, hạn chế của phương pháp này là phải sử dụng gói tin độc lập gây lãng phí băng thông và phức tạp hóa việc quản lý



53

Thiết bị mạng thông minh (4)

- Để có thể cảnh báo cho phía phát, nút mạng có thể sử dụng một trong hai cơ chế(cont.)
 - Sử dụng một bit chỉ thị tắc nghẽn nằm trong trường điều khiển của gói tin mang dữ liệu từ phía thu sang phía phát. Bit chỉ thị tắc nghẽn bằng 0 thể hiện tắc nghẽn không xảy ra và bit này bằng 1 khi tắc nghẽn xảy ra.
- Phía phát sẽ dựa trên thông tin cảnh báo này để quyết định việc tăng giảm kích thước cửa sổ.



54

Thiết bị mạng thông minh (5)

- Nếu việc thay đổi kích thước cửa sổ chỉ được dựa trên một gói tin phản hồi thì có thể xảy ra tình trạng hệ thống hoạt động không hiệu quả. Vì vậy, trên thực tế, phía phát sẽ dựa trên một số lượng thông báo nhất định từ phía nút mạng rồi mới kết luận về tình trạng tắc nghẽn.



55

Thiết bị mạng thông minh (6)

- Thông thường, với một số lượng thông báo nhận được, nếu số gói tin cảnh báo tắc nghẽn vượt quá một giới hạn nào đó thì phía phát sẽ coi là có tắc nghẽn xảy ra và giảm kích thước cửa sổ. Nếu số lượng cảnh báo này nhỏ hơn giới hạn cho phép thì phía phát sẽ coi là không có tắc nghẽn và tăng kích thước cửa sổ.



56

Thiết bị mạng thông minh (7)

- Việc tăng và giảm kích thước cửa sổ có thể tuân theo một trong hai quy tắc: phép cộng và phép nhân.
- Phép cộng: $W_{new} = W_{old} + I$.
 - W_{new} và W_{old} là kích thước cửa sổ mới và cũ, I là hệ số tăng giảm.
 - Khi $I > 0$ là tăng kích thước cửa sổ và $I < 0$ là giảm kích thước cửa sổ



57

Thiết bị mạng thông minh (8)

- Phép nhân: $W_{new} = W_{old} * \alpha$.
 - W_{new} và W_{old} là kích thước cửa sổ mới và cũ
 - Khi $\alpha > 1$ là tăng kích thước cửa sổ và $\alpha < 1$ là giảm kích thước cửa sổ.
 - Trong trường hợp kích thước cửa sổ không phải số nguyên thì kích thước đó sẽ được quy về số nguyên gần nhất.
- Trong ứng dụng cụ thể, người ta thường dùng phép cộng khi tăng và dùng phép nhân khi giảm.



58

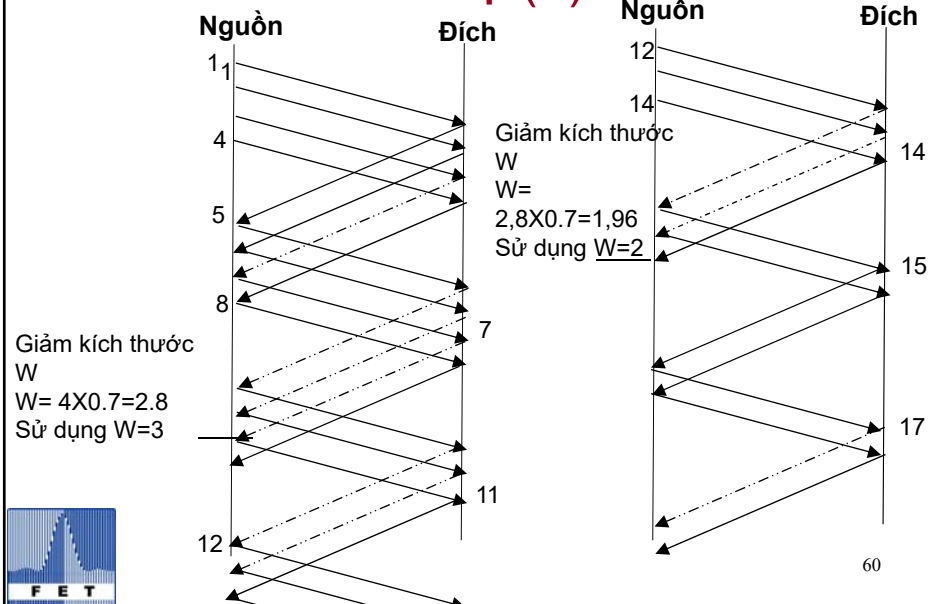
Ví dụ

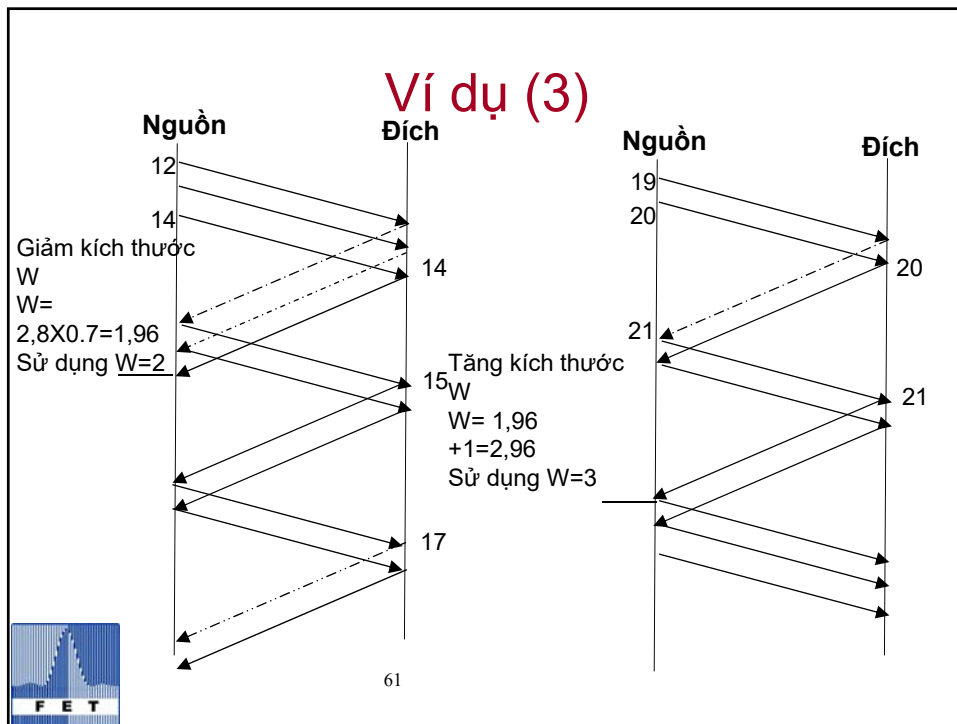
- Kích thước cửa sổ ban đầu là $W = 4$,
- Việc kết luận về tình trạng tắc nghẽn được dựa trên các nhóm 7 báo nhận gửi về.
 - Trong 7 báo nhận đó, nếu có lớn hơn hoặc bằng 4 báo nhận có bit chỉ thị tắc nghẽn bằng 1 thì nút nguồn coi là có tắc nghẽn và giảm kích thước cửa sổ, ngược lại thì nút nguồn coi là không có tắc nghẽn và tăng kích thước cửa sổ.
- Việc giảm được thực hiện theo phép nhân với $\alpha = 0,7$ và việc tăng được thực hiện theo phép cộng với $I = 1$.



59

Ví dụ (2)





Thiết bị mạng không thông minh

- Các thiết bị mạng không có khả năng cảnh báo cho phía phát về tình trạng tắc nghẽn .
- Việc xác định tắc nghẽn trong mạng hoàn toàn dựa trên việc suy đoán của nút nguồn.
- Thiết bị mạng không thông minh là các thiết bị mạng đơn giản, không có khả năng xác định trạng thái bộ đệm, trạng thái CPU hay trạng thái sử dụng đường truyền.



62

Thiết bị mạng không thông minh(2)

- Trong một số trường hợp khác, do yêu cầu hoạt động với **tốc độ cao** nên các thiết bị mạng có thể cũng không kiểm tra về tình trạng tắc nghẽn có thể xảy ra mỗi khi gói tin đi qua thiết bị.



63

Thiết bị mạng không thông minh(3)

- Khi không có sự hỗ trợ của thiết bị mạng, nút nguồn kết luận về trạng thái tắc nghẽn hoàn toàn dựa trên báo nhận được gửi về.
- Trong trường hợp mạng bị tắc nghẽn, báo nhận có thể bị trễ lớn (trễ báo nhận hoặc trễ gói đến phía thu) hoặc có thể bị mất (mất báo nhận hoặc mất gói nên không có báo nhận).
- Trong trường hợp mất báo nhận hoặc báo nhận đến quá trễ, nút nguồn sẽ phải phát lại gói và việc phát lại này có thể coi là **một tín hiệu** để kết luận về tình trạng tắc nghẽn.



64

Thiết bị mạng không thông minh(4)

- Cơ chế tắc nghẽn này gọi là cơ chế điều khiển tắc nghẽn dùng cửa sổ thích ứng dựa trên time-out và hoạt động như sau:
- Tại thời điểm ban đầu, nút nguồn đặt kích thước cửa sổ bằng W_{max}
- Mỗi khi có time-out xảy ra và phía phát phải thực hiện phát lại gói tin thì nút nguồn sẽ đặt $W = 1$



65

Thiết bị mạng không thông minh(5)

- Mỗi khi nhận được n báo nhận từ nút đích, phía phát lại tăng kích thước cửa sổ lên 1.
- Kích thước cửa sổ sẽ không bao giờ vượt quá kích thước cửa sổ W_{max} . Với việc thay đổi giá trị n , người ta có thể thực hiện điều khiển tắc nghẽn ở nhiều mức độ khác nhau.
- Trong trường hợp này, chúng ta giả thiết tỷ lệ lỗi bit là khá nhỏ và time-out xảy ra hoàn toàn là do trễ chứ không phải do mất gói vì lỗi bit.



66

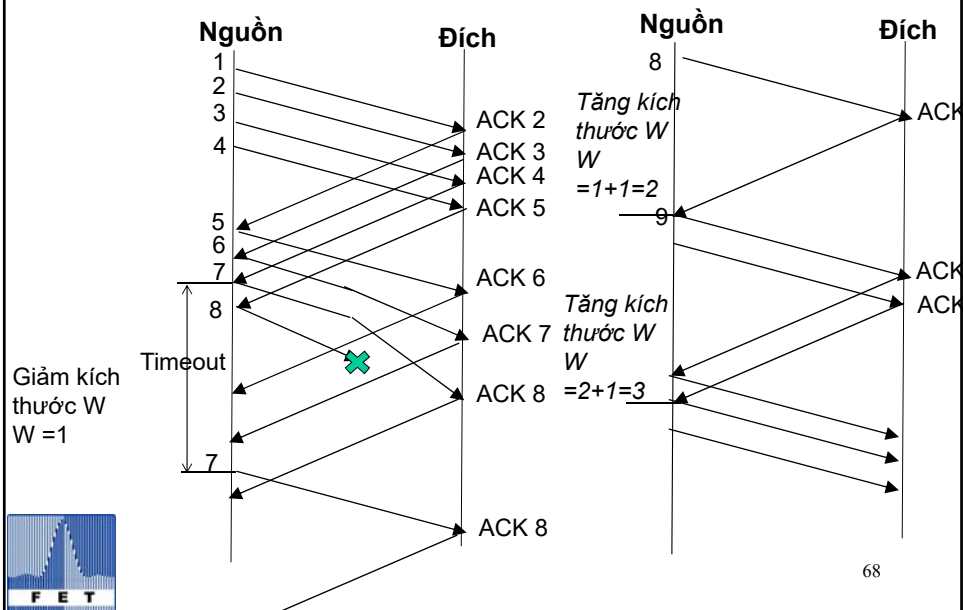
Ví dụ

- Kích thước cửa sổ ban đầu $W_{max} = 4$, và giá trị $n = 2$. Giả thiết rằng các nút mạng trung gian có thể gây ra trễ hoặc hủy gói tin hoặc báo nhận nếu tắc nghẽn xảy ra. Điều này dẫn đến hệ quả là có time-out xảy ra tại nút nguồn cho các gói tin đó.



67

Ví dụ (2)



68