

SO SÁNH HIỆU SUẤT CỦA CÁC KẾ HOẠCH KIỂM SOÁT LỖI TRONG MẠNG TRUYỀN THÔNG MÁY TÍNH

NGUYỄN ĐÌNH VIỆT

Abstract. This paper analyzes quantitatively link-by-link and end-to-end error control schemes in computer communication networks. Two measures used for performance evaluations are mean packet delay time and throughput of the networks. With two specific examples, performance comparisons are made for two cases: traditional computer networks (low-speed) and high-speed (gigabit/sec) computer networks. The results is that: link-by-link error control scheme is superior than end-to-end error control schemes for low-speed computer networks, by contrast, for high-speed computer networks end-to-end scheme is superior than link-by-link scheme.

Tóm tắt. Bài báo này phân tích một cách định tính các kế hoạch kiểm soát lỗi link-by-link và end-by-end trong mạng truyền thông máy tính. Hai tiêu chuẩn được dùng cho sự so sánh là thời gian trễ của gói tin và thông lượng của mạng. Với hai ví dụ đặc trưng, việc so sánh hiệu suất của các kế hoạch được thực hiện cho hai trường hợp: mạng truyền thông máy tính tốc độ thấp và mạng truyền thông máy tính tốc độ cao. Kết quả là: kế hoạch kiểm soát lỗi link-by-link tốt hơn kế hoạch end-by-end đối với các mạng tốc độ thấp, ngược lại, đối với các mạng tốc độ cao thì kế hoạch end-by-end lại tốt hơn so với kế hoạch link-by-link.

1. GIỚI THIỆU

Trong các mạng chuyển mạch gói, nói chung có hai cách tiếp cận nhằm giải quyết vấn đề các bản tin bị mất hoặc bị hỏng khi chúng được truyền giữa hai người dùng cuối (end-user) trên mạng.

Cách tiếp cận link-by-link: việc kiểm soát lỗi chủ yếu được giải quyết bởi các giao thức tầng liên kết dữ liệu (data-link), trong đó hai nút lân cận trên đường truyền phát hiện và khắc phục việc bản tin bị mất hoặc hỏng trên đường truyền nối hai nút đó. Theo cách tiếp cận này vẫn cần phải có các giao thức end-to-end ở mức cao hơn, thí dụ để khắc phục lỗi gây ra do một nút mạng nào đó trên con đường truyền bị hỏng.

Cách tiếp cận end-to-end: việc kiểm soát lỗi chỉ dựa trên cơ sở một giao thức end-to-end, trong đó hai nút ở hai đầu mút của kênh truyền phát hiện và khắc phục việc bản tin bị mất hoặc hỏng trên các đường truyền nối hai nút đó. Các chức năng truyền thống của giao thức tầng liên kết dữ liệu bị loại bỏ gần hết, mặc dầu có thể vẫn giữ lại chức năng phát hiện lỗi chứ không giữ lại chức năng khắc phục lỗi.

Trong các mạng máy tính đầu tiên, đường truyền có tốc độ rất thấp và tỉ suất lỗi bit rất cao so với các đường truyền ngày nay, thí dụ mạng ARPANET thời kỳ đầu sử dụng các đường truyền 50 Kbps, tỉ suất lỗi bit khoảng 10^{-5} . Nhiều nhà khoa học đã nghiên cứu và họ đã chứng minh được rằng, trong các mạng này cách tiếp cận link-by-link cho hiệu suất cao hơn so với cách tiếp cận end-to-end.

Ngày nay, nhờ các tiến bộ trong công nghệ phần cứng và công nghệ quang sợi, chúng ta có thể sử dụng các đường truyền tốc độ gigabit trong các mạng truyền thông máy tính chuyển mạch gói. Các mạng tốc độ cao này có các đặc tính khác hẳn các mạng tốc độ thấp trước đây, cụ thể là:

- Thứ nhất, tỉ số của tốc độ truyền trên tốc độ xử lí đã tăng lên rất nhiều, thúc đẩy nhu cầu chuyển các giao thức nặng về tính toán đã được đặt trong tầng bên dưới lên các tầng bên trên của mạng. (Trong khoảng 30 năm qua, tốc độ truyền tăng cỡ 10^5 lần, tốc độ xử lí cỡ 10^3 lần).

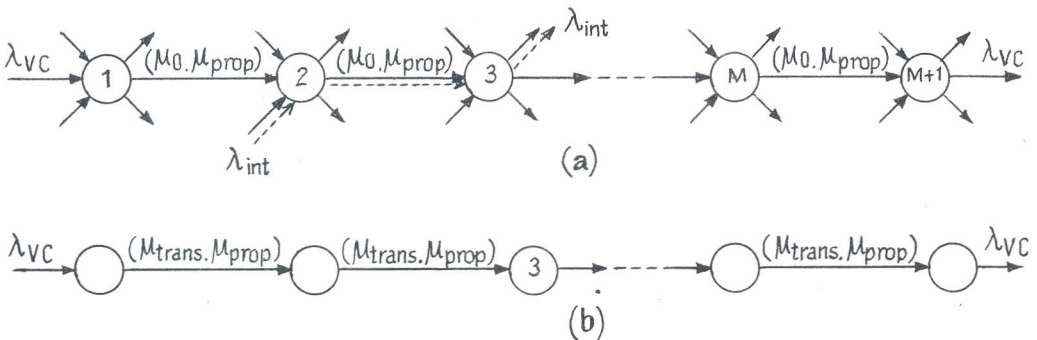
- Thứ hai, tỉ số của độ trễ lan truyền bản tin trên thời gian truyền bản tin cũng tăng lên, nhất là trong trường hợp mạng diện rộng. (Trong khoảng 30 năm qua, tỉ số này tăng cỡ 10^6 lần).
- Cuối cùng là việc sử dụng cáp quang làm môi trường truyền thông sẽ làm cho tỉ suất lỗi bit của kênh giảm đi rất nhiều (từ 10^3 đến 10^4 lần).

Những thay đổi trên làm thay đổi các tiền đề cơ sở mà người ta đã dựa vào đó để phát triển công nghệ chuyển mạch gói của các mạng WAN truyền thống, đòi hỏi chúng ta phải xem xét lại cẩn thận việc thiết kế các giao thức nằm giữa các thiết bị truyền và các thiết bị chuyển mạch tốc độ cao và các giao thức tầng ứng dụng ở lớp trên của mạng tốc độ cao.

2. MÔ HÌNH MẠNG

Mô hình mạng được dùng để so sánh việc kiểm soát lỗi link-by-link và end-to-end được trình bày trên hình 1a. Mạch ảo (VC - Virtual Circuit) có M đường truyền, bắt đầu từ nút nguồn được ghi nhãn 1 và kết thúc tại nút đích được ghi nhãn $M + 1$, λ_{VC} là lưu lượng đi vào mạch ảo, μ'_0 là tốc độ phát bản tin của đường truyền i , $1/\mu'_{prop}$ là độ trễ lan truyền trung bình giữa nút i và $i + 1$.

Trên thực tế một số VC có thể chia sẻ một đường truyền vật lý, thí dụ trên hình 1(a) một VC có tốc độ λ_{int} chia sẻ đường truyền giữa nút 2 và 3 với λ_{VC} . Chúng ta lập mô hình ảnh hưởng của lưu lượng đi vào này bằng cách giảm năng lực truyền mà λ_{VC} có thể dùng dọc theo mỗi đường truyền của VC đi một lượng thích hợp. Mô hình cho một mạch ảo M chặng vì thế được thể hiện trên hình 1(b), với μ^i_{trans} biểu diễn cho năng lực phục vụ hiệu dụng của đường truyền i đối với mạch ảo λ_{VC} .



Hình 1. Mô hình mạch ảo

3. KIỂM SOÁT LỖI END-TO-END

Trong mục này chúng ta sẽ phát triển một mô hình giải tích để nghiên cứu hiệu suất của kế hoạch kiểm soát lỗi end-to-end. Mô hình của chúng ta xét một cách cụ thể ảnh hưởng của độ trễ lan truyền của bản tin, dung lượng bộ nhớ đệm hữu hạn, lỗi của kênh truyền và việc sử dụng cơ chế hết giờ (time-out) trong kế hoạch kiểm soát lỗi.

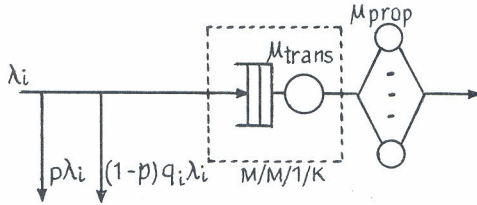
3.1. Mô tả việc kiểm soát lỗi end-to-end

Khi nút nguồn VC truyền một bản tin, nó sẽ nhớ trong bộ nhớ đệm bản copy của bản tin đó cho đến khi nhận được biên nhận (ack) đến từ trạm đích. Trạm đích chỉ sinh ra bản tin biên nhận khi bản tin mà nó nhận được là đúng. Nếu nút nguồn không nhận được ack trong một khoảng thời gian time-out định trước thì nó sẽ coi bản tin là bị mất hoặc hỏng và sẽ truyền lại bản copy của bản tin đó. Chúng ta giả thiết rằng time-out chỉ xảy ra khi bản tin thực sự bị mất hoặc bị hỏng. Chú ý rằng của nút trung gian trong cách tiếp cận này là đơn giản: kiểm tra các bản tin đi đến xem có lỗi hay không, nếu có lỗi thì loại bỏ, nếu không lỗi thì xếp vào hàng đợi để truyền đi tiếp. Các bản tin

đi đến dù không có lỗi, nhưng nếu tất cả các bộ đệm đã đầy thì cũng sẽ bị loại.

3.2. Mô hình mức liên kết dữ liệu cho việc kiểm soát lỗi end-to-end

Mô hình hàng đợi được thể hiện trên hình 2. Để cho đơn giản chúng ta giả thiết rằng tất cả các nút dọc theo VC là giống nhau, nghĩa là chúng có cùng một tốc độ truyền bản tin, độ trễ lan truyền và một số cố định các bộ đệm. Nút thứ i được mô hình bằng hệ thống hàng đợi $M/M/1/K$. Độ trễ lan truyền $1/\mu_{prop}$ của đường truyền được mô hình bởi một hàng đợi có vô hạn server, mỗi server có tốc độ phục vụ bằng μ_{prop} .



Hình 2. Mô hình hàng đợi cho một nút

Khi bản tin đến nút i , một trong 3 sự kiện sau có thể xảy ra:

- Bản tin được nhận nhưng có lỗi với xác suất p , nó sẽ bị loại bỏ.
- Bản tin không bị lỗi nhưng không còn bộ đệm rồi, nó cũng sẽ bị loại bỏ.
- Bản tin được nhận đúng và được cấp bộ đệm.

Chúng ta giả thiết rằng mỗi gói tin đi đến nút i một cách độc lập sẽ thấy tất cả bộ đệm đầy với xác suất là q_i , là xác suất trạng thái dừng, q_i khác nhau đối với mỗi nút của VC. Chúng ta kí hiệu: $f_i = q_i + p - q_i p$ là xác suất mà một gói tin bị loại bỏ bởi nút i .

Người ta đã có lời giải cho hàng đợi $M/M/1/K$ như sau [1, 2]:

- $E[L_{1,i}]$ là số bản tin trung bình ở nút trung gian i , cũng chính là chiều dài trung bình của hàng đợi $M/M/1/K$ tính được như sau:

$$E[L_{1,i}] = \begin{cases} \rho_{1,i}/(1 - \rho_{1,i}), & i = 1 \\ \rho_{1,i}/(1 - \rho_{1,i}) - (K + 1)\rho_{1,i}^{K+1}/(1 - \rho_{1,i}^{K+1}), & i = 2, \dots, M \end{cases}$$

(với $\rho_{1,i} = \lambda_i(1 - p)/\mu_{trans}$).

- $E[W_{1,i}]$ là thời gian trung bình mà một gói tin đứng xếp hàng đợi truyền cộng với thời gian truyền; sử dụng công thức Little chúng ta sẽ có:

$$E[W_{1,i}] = E[L_{1,i}]/\lambda_i(1 - f_i). \tag{1}$$

- $E[W_i]$ là độ trễ trung bình (xếp hàng + truyền + lan truyền) mà một gói tin phải chịu khi đi qua đường truyền i :

$$E[W_i] = E[W_{1,i}] + 1/\mu_{prop}. \tag{2}$$

- Xác suất blocking q_i , là xác suất tất cả các bộ đệm tại nút i đầy, (và vì vậy nút $i - 1$ bị chặn - blocked):

$$q_i = \Pr[L_{1,i} = K] = \frac{(1 - \rho_{1,i})\rho_{1,i}^{K+1}}{1 - \rho_{1,i}^{K+1}}, \quad i = 2, \dots, M. \tag{3}$$

Các phương trình (2) và (3) cho ta độ trễ trung bình và xác suất blocking tại nút i . Mỗi phương trình này là một hàm của các giá trị còn chưa được xác định $\{\lambda_i\}$; các giá trị này sẽ nhận được trong mục sau, trong đó chúng ta kết hợp các mô hình mức liên kết dữ liệu lại thành một mô hình mức VC.

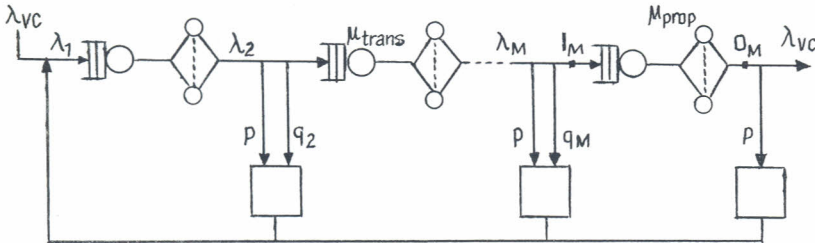
3.3. Mô hình mức VC cho việc kiểm soát lỗi end-to-end

Hình 3 kết hợp M mô hình mức liên kết dữ liệu riêng biệt thành một mô hình duy nhất cho VC. Để thực sự giải được các xác suất blocking và độ trễ trung bình theo các đường truyền, chúng ta vẫn phải xác định $\{\lambda_i\}$. Khó khăn ở đây là λ_i và q_i là phụ thuộc hàm của nhau. Hơn nữa $\lambda_i \neq \lambda_j \neq \lambda_{VC}$, do xác suất blocking với các nút là khác nhau và thực tế là một bản tin sẽ bị loại bỏ tại nút đầu tiên phát hiện ra nó bị lỗi. Cách tiếp cận của chúng ta là trước hết tìm tốc độ đi đến và xác suất blocking cho chặng cuối cùng của VC (đường truyền M), rồi sử dụng các giá trị này để tìm các giá trị tương ứng cho đường truyền $M - 1$, và sau đó tiếp tục giải quyết mô hình mức VC từ phía cuối ngược trở lại đầu của nó.

Rõ ràng là tốc độ truyền bản tin ra đường truyền M tại điểm O_M trên hình 3 được cho bởi $\lambda_{VC}/(1-p)$. Do tính bảo toàn thông lượng, tốc độ đi vào đường truyền M tại điểm I_M cũng phải bằng giá trị này, và vì thế:

$$\lambda_M = \frac{\lambda_{VC}}{(1-p)^2(1-q_M)}. \quad (4)$$

Thay biểu thức này cho λ_M vào (3) sẽ cho ta một phương trình duy nhất với một biến q_M và như vậy chúng ta có thể tìm ngay được q_M . Khi đã biết một giá trị của q_M , (4) sẽ cho ta giá trị của λ_M . Cuối cùng, khi đã biết λ_M , giải phương trình (2) chúng ta sẽ nhận được độ trễ trung bình qua đường truyền M . Khi đã biết được λ_M thì q_{M-1} , λ_{M-1} , và $E[W_{M-1}]$ có thể nhận được bằng cách sử dụng các lập luận tương tự. Bằng cách làm việc theo chiều ngược trên VC chúng ta sẽ nhận được tập các giá trị chưa biết $\{q_i\}$, $\{\lambda_i\}$ và $\{E[W_j]\}$.



Hình 3. Mô hình mức VC của sự kiểm soát lỗi end-to-end

Cuối cùng, để tính độ trễ trung bình dọc theo VC, phải định nghĩa thêm một số đại lượng bổ sung như sau:

- p_{fail} - xác suất việc truyền một gói tin từ nút nguồn đến nút đích không thành công.

$$p_{fail} = \sum_{i=1}^M f_i \prod_{j=1}^{i-1} (1 - f_j).$$

- N_t - số lần phải phát lại một gói tin trước khi nó được nhận thành công ở trạm đích. Giả sử rằng xác suất của việc phát lại mỗi gói tin là độc lập, N_t có phân bố $P[N_t = k] = p_{fail}^k (1 - p_{fail})$, $k = 0, 1, \dots$ và giá trị trung bình của N_t (mean):

$$E[N_t] = \sum_{k=1}^{\infty} k P[N_t = k] = \frac{p_{fail}}{1 - p_{fail}}. \quad (5)$$

- T_{ee} - khoảng thời gian time-out đối với VC.
- $E[W_{ee}]$ - thời gian trễ end-to-end trung bình mà một gói tin phải trải qua.

Sự trễ của gói tin từ khi nó đi đến nút nguồn cho đến khi được nhận thành công tại đích sinh ra do N_t khoảng thời gian time-out, trong suốt khoảng thời gian đó nó được phát đi tới đích nhưng không thành công, cộng với những độ trễ mà nó phải chịu dọc theo VC trong lần nó được truyền thành công tới đích. Vì vậy:

$$E[W_{ee}] = E[N_t](E[W_{1,1}] + T_{ee}) + \sum_{k=1}^M E[W_k]. \tag{6}$$

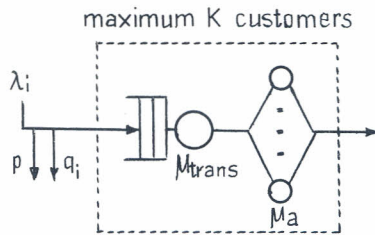
4. KIỂM SOÁT LỖI LINK-BY-LINK

4.1. Mô tả việc kiểm soát lỗi link-by-link

Trong cách tiếp cận link-by-link, mỗi nút dọc theo VC nhớ đệm bản copy của bản tin được phát đi cho đến khi nhận được bản tin ack cho bản tin đó từ nút tiếp theo trong VC gửi trở về. Nút sẽ sinh ra một bản tin ack cho mỗi bản tin mà nó nhận được đúng. Nếu không nhận được ack trong khoảng thời gian time-out, thì bên gửi sẽ truyền lại bản copy của bản tin cho nút tiếp theo trong VC.

4.2. Mô hình mức liên kết dữ liệu cho sự kiểm soát lỗi link-by-link

Để cho đơn giản, chúng ta giả thiết rằng các nút là đồng nhất. Mô hình của một nút mạng được thể hiện trên hình 4; trong đó chúng ta phải xét đến việc bản tin sẽ tiếp tục chiếm bộ đệm trong khi nó và ack của nó (nếu có) lan truyền trên kênh. Mô hình là một hàng đợi $M/M/1$, tốc độ phục vụ $1/\mu_{trans}$, chính sách phục vụ FCFS, độ trễ truyền được mô hình bằng thời gian phục vụ của một server thuộc hàng đợi có vô hạn server (hàng đợi IS - Infinites Server queue) là $1/\mu_2 = 2/\mu_{prop}$. Các khách hàng (customer) trong hàng đợi IS là các bản tin mà thời gian time-out cho chúng đã được bắt đầu tính, mỗi khách hàng này phải chịu một độ trễ cố định là $1/\mu_2 = 2/\mu_{prop}$. Như vậy tổng số bản tin thực sự được nhớ đệm tại nút i là tổng của số khách hàng trong các hàng đợi FCFS và IS. Dung lượng bộ đệm cố định là K , nút sẽ không cho phép các gói tin đi vào mô hình hàng đợi node/link (là phần được bao bởi đường gạch gạch trên hình 4) khi tổng này bằng K .



Hình 4. Mô hình hàng đợi mức thấp cho giao thức link-by-link

Chúng ta quan tâm đến phân bố dừng của số gói tin trong hàng đợi truyền $L_{1,i}$ và số gói tin mà thời gian time-out đối với chúng đã được bắt đầu tính $L_{2,i}$. Cho $\pi_{i,m,n} = P[L_{1,i} = m, L_{2,i} = n]$, $0 \leq m, n \leq K, m + n \leq K$. Mạng hàng đợi này có phân bố chiều dài hàng đợi dạng tích, các xác suất đã được người ta tính như sau [1, 3]:

$$\pi_{i,m,n} = G^{-1} \rho_{1,i}^m \frac{\rho_{2,i}^n}{n!}, \quad 0 \leq m, n \leq K, m + n \leq K. \tag{7}$$

Trong đó

$$\rho_{1,i} = \lambda_i(1 - p)/\mu_{trans}, \quad \rho_{2,i} = \lambda_i(1 - p)/\mu_2.$$

G^{-1} là hằng số chuẩn hóa, được cho như sau:

$$G = \frac{1 - \rho_{1,i}}{S_K(\rho_{2,i}) - \rho_{1,i}^{K+1} S_K(\rho_{2,i}/\rho_{1,i})}, \quad \text{với } S_K(x) = \sum_{j=0}^K \frac{x^j}{j!}. \tag{8}$$

Xác suất blocking

$$q_i = \sum_{m=0}^K \pi_{i,m,K-m}. \quad (9)$$

Số bản tin trung bình tại nút i của VC đang đợi để được truyền đi được cho bởi:

$$E[L_{1,i}] = \sum_{m=0}^K m \sum_{n=0}^{K-m} \pi_{i,m,n}. \quad (10)$$

Theo định luật của Little, thời lượng một bản tin phải xếp hàng đứng đợi được truyền đi một cách tích cực (nghĩa là trái với thời gian đợi cho time-out hết) được cho bởi:

$$E[W_{1,i}] = E[L_{1,i}] / \lambda_i (1 - f_i), \quad (11)$$

trong đó $f_i = q_i + p - q_i p$ là xác suất mà một bản tin hoặc là được nhận nhưng có lỗi hoặc là khi nó đến nút i thì thấy bộ đệm đầy.

Giống như trước đây, chúng ta định nghĩa $E[W_i]$ là độ trễ trung bình mà bản tin phải chịu khi nó đi qua đường truyền i . Chú ý rằng nói chung $E[W_i] \neq [W_{1,i}] + 1/\mu_{\text{prop}}$, bởi vì trong trường hợp link-by-link, bản tin có thể được phát lại một vài lần trước khi nó được nhận thành công ở nút $i + 1$. Tuy nhiên, chúng ta có thể tiếp tục làm theo cách đã sử dụng trong Mục 3.3 và tính $E[W_i]$ như dưới đây.

Để tính $E[W_i]$ chúng ta cần định nghĩa một số đại lượng:

- N_i^k - số lần phải phát lại một gói tin tại nút i trước khi nó nhận được thành công tại trạm $i + 1$. Theo giả thiết xác suất phát lại mỗi gói tin là độc lập, phân bố của N_i^k là $P[N_i^k = k] = f_{i+1}^k (1 - f_{i+1})$, $k = 1, 2, \dots$, vì thế chúng ta sẽ có:

$$E[N_i^k] = \frac{f_{i+1}}{1 - f_{i+1}}. \quad (12)$$

- T_{li} - thời gian time-out của đường truyền. Trong các thí dụ chúng ta lấy T_{li} bằng hai lần độ trễ lan truyền, nghĩa là $T_{li} = 2/\mu_{\text{prop}}$.

Độ trễ của gói tin tính từ khi nó đến nút i cho đến khi nó nhận thành công ở nút $i + 1$ sinh ra do N_i^k khoảng thời gian time-out cộng với các độ trễ mà nó phải chịu trong lần được truyền thành công. Vì vậy:

$$E[W_i] = E[N_i^k] (E[W_{1,i}] + T_{li}) + E[W_{1,i}] + 1/\mu_{\text{prop}}. \quad (13)$$

Các phương trình (13) và (9) cho ta độ trễ trung bình và xác suất blocking tại nút i . Giống như trong trường hợp end-to-end, các phương trình này đều là hàm của các giá trị còn chưa được xác định $\{\lambda_i\}$.

4.3. Mô hình mức VC cho việc kiểm soát lỗi link-by-link

Mô hình mức VC cho link-by-link giống mô hình end-to-end; sự khác nhau duy nhất là các xác suất blocking được tính bằng cách sử dụng (9) chứ không phải là (3). Cách tính toán tương tự như đã trình bày ở trên.

Độ trễ của gói tin từ khi nó mới bắt đầu đến nút nguồn của VC cho đến khi nó được nhận thành công tại nút đích của VC là tổng của các độ trễ riêng biệt mà nó phải chịu trên mỗi đường truyền:

$$E[W_{li}] = \sum_{i=1}^M E[W_i]. \quad (14)$$

5. SO SÁNH CÁC KẾ HOẠCH KIỂM SOÁT LỖI LINK-BY-LINK VỚI END-TO-END

Sau khi đã phát triển các mô hình giải tích về giao thức kiểm soát lỗi link-by-link và end-to-end, bây giờ chúng ta sẽ sử dụng các mô hình này để so sánh hiệu suất của các giao thức. Dưới đây chúng ta sẽ xét hai thí dụ về mạng tốc độ thấp và mạng tốc độ cao, qua đó sẽ rút ra được các kết luận cần thiết.

5.1. So sánh link-by-link với end-to-end ở mạng tốc độ thấp

Thí dụ thứ nhất, cho một mạng WAN, kích thước 400 km, sử dụng các đường truyền tốc độ thấp, tỉ suất lỗi bit cao, tương tự với các tham số đường truyền của hệ thống điện thoại kiểu cũ. Các tham số cụ thể như sau:

Số chặng (hop) trong mạng:	$M = 8$	Độ dài mỗi chặng:	50 km
Dung lượng đường truyền:	$C = 50$ Kbps	Số lượng bộ đệm tại mỗi nút:	2, 4, 10
Độ dài gói tin (packet):	$l = 1000$ bit	Xác suất lỗi bit:	$P_b = 10^{-5}$

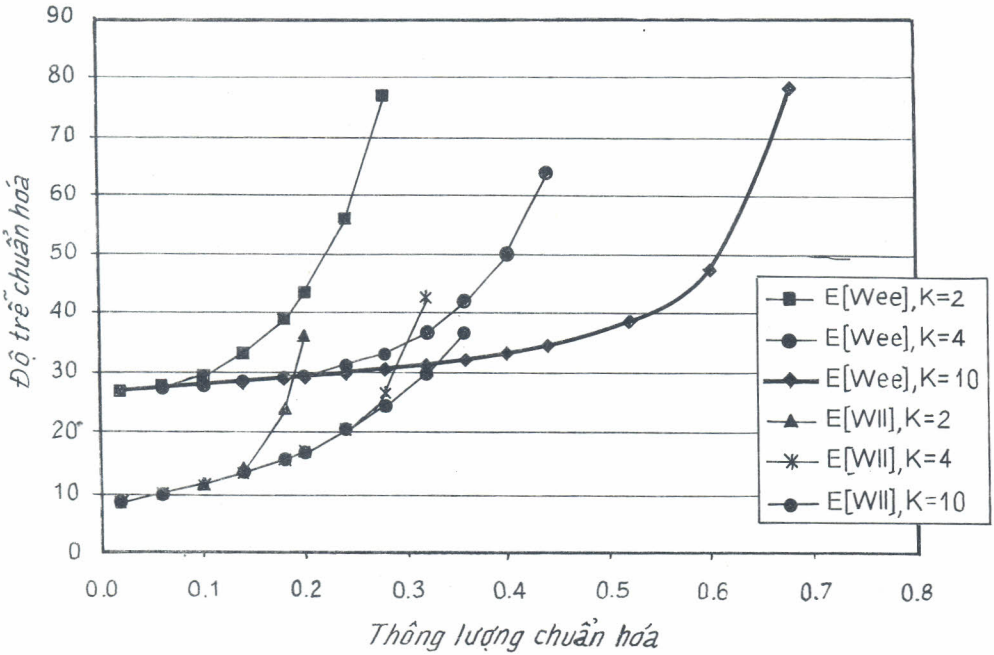
Từ đó tính ra được: $1/\mu_{trans} = l/C = 0,02$; $1/\mu_{prop} = 0,00025$; $p = 1 * P_b = 10^{-2}$. Thay các giá trị cụ thể này vào các biểu thức thu được ở trên, chúng ta sẽ nhận được các kết quả trong bảng 1, và được biểu diễn bằng đồ thị trên hình 5, trong đó D là thông lượng chuẩn hóa, đó là tỉ số của thông lượng đi vào mạng trên dung lượng của đường truyền. Tất cả các đại lượng thời gian tính được đều được chuẩn hóa theo thời gian truyền một gói tin, tức là đều được chia cho $1/\mu_{trans}$. Chúng tôi sử dụng phần mềm Mathcad, phiên bản 8.0 (Mathcad 2000) để tính toán. Các phép tính được thực hiện với 15 chữ số sau dấu phẩy thập phân.

Bảng 1. Sự phụ thuộc của độ trễ chuẩn hóa vào thông lượng chuẩn hóa

D	0,02	0,06	0,10	0,14	0,18	0,20	0,24	0,28	0,32	0,36	0,40	0,44	0,52	0,60	0,68
$E[W_{ee}]$ $K=2$	27,10	27,82	29,59	33,11	39,21	43,60	56,24	77,18							
$E[W_{ee}]$ $K=4$	27,09	27,48	27,91	28,45	29,20	29,71	31,11	33,30	36,70	42,00	50,39	64,28			
$E[W_{ee}]$ $K=10$	27,09	27,47	27,89	28,36	28,87	29,15	29,76	30,44	31,22	32,12	33,20	34,53	38,63	47,60	78,52
$E[W_{ll}]$ $K=2$	8,74	10,02	11,73	14,48	24,41	36,04									
$E[W_{ll}]$ $K=4$	8,74	10,02	11,57	13,46	15,81	17,23	20,84	26,45	42,60						
$E[W_{ll}]$ $K=10$	8,74	10,02	11,57	13,45	15,77	17,13	20,37	24,47	29,76	36,72					

Nhận xét:

- Nếu số bộ đệm lớn hơn 4 và thông lượng không vượt quá 30% năng lực vận chuyển của đường truyền thì sử dụng giao thức kiểm soát lỗi link-by-link cho thời gian trễ nhỏ hơn hẳn so với giao thức kiểm soát lỗi end-to-end. Trong thực tế các mạng tốc độ thấp thường hoạt động trong miền thông lượng này, do đó việc sử dụng giao thức kiểm soát lỗi link-by-link rõ ràng là tốt hơn giao thức end-to-end.
- Trong cả hai giao thức kiểm soát lỗi, ở miền thông lượng nhỏ không vượt quá 10% năng lực vận chuyển của đường truyền, việc tăng thêm số bộ đệm ở các nút mạng không đem lại lợi ích gì. Trong miền thông lượng lớn hơn, việc tăng số bộ đệm ở các nút mạng sẽ có ích, nó làm giảm thời gian trễ. Tuy nhiên việc tăng số bộ đệm lên cao không làm cho thời gian trễ giảm đi theo cùng một tỉ lệ.



Hình 5. Độ trễ chuẩn hóa thay đổi theo thông lượng chuẩn hóa

5.2. So sánh link-by-link với end-to-end ở mạng tốc độ cao

Thí dụ thứ hai, cho một mạng WAN, kích thước 400 km, sử dụng các đường truyền tốc độ cao, tỉ suất lỗi bit thấp, tương tự với các tham số đường truyền sử dụng cáp quang. Các tham số cụ thể như sau:

Số chặng (hop) trong mạng:	$M = 8$	Độ dài mỗi chặng:	50 km
Dung lượng đường truyền:	$C = 100$ Mbps	Số lượng bộ đệm tại mỗi nút:	10, 15, 20
Độ dài gói tin (packet):	$l = 1000$ bit	Xác suất lỗi bit:	$P_b = 10^{-9}$

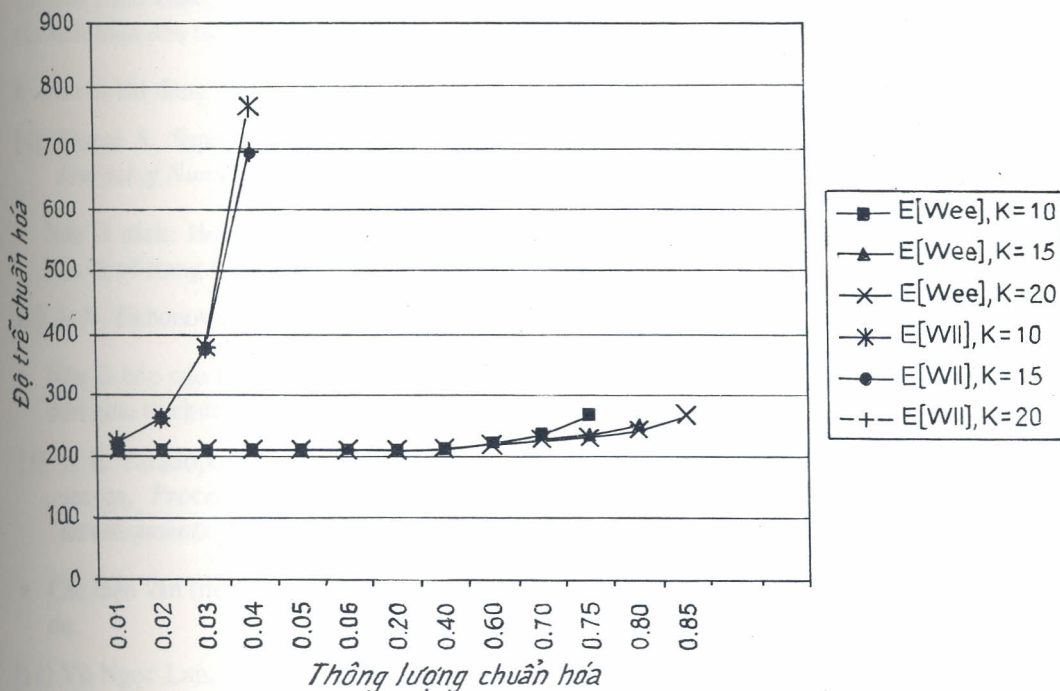
Từ đó tính ra được: $1/\mu_{trans} = l/C = 0,00001$; $1/\mu_{prop} = 0,00025$; $p = 1 * P_b = 10^{-6}$. Thay các giá trị cụ thể này vào các biểu thức thu được ở trên, chúng ta sẽ nhận được các kết quả trong bảng 2 và được biểu diễn bằng đồ thị trên hình 6, trong đó thông lượng và thời gian trễ chuẩn hóa được định nghĩa giống như trên.

Nhận xét:

- Trong toàn bộ miền thông lượng được khảo sát, sử dụng giao thức kiểm soát lỗi end-to-end cho thời gian trễ nhỏ hơn hẳn so với kiểm soát lỗi link-by-link. Thậm chí có thể nói rằng không thể sử dụng các giao thức link-by-link, bởi vì nếu sử dụng chúng, khi tải tăng từ không lên khoảng 4% năng lực vận chuyển của đường truyền thì thời gian trễ tăng gấp hơn ba lần và sẽ tăng lên rất nhanh chóng.
- Nếu sử dụng giao thức kiểm soát lỗi end-to-end, trong phần lớn miền thông lượng thường được sử dụng (dưới 70% năng lực vận chuyển của đường truyền) số bộ đệm hầu như không ảnh hưởng tới độ trễ, chỉ trong miền thông lượng lớn (trên 70% năng lực vận chuyển của đường truyền) việc tăng số bộ đệm mới đem lại lợi ích, nhưng không nhiều. Nếu số bộ đệm tăng lên đến 20 thì có thể sử dụng đến 85% năng lực vận chuyển của đường truyền với độ trễ lớn hơn độ trễ khi tải nhẹ không quá 1,3 lần. Việc tăng số bộ đệm vượt quá 20 có thể coi là vô ích bởi vì trong thực tế khi thiết kế mạng người ta không sử dụng thông lượng ở miền xấp xỉ năng lực vận chuyển của đường truyền (miền bão hòa).

Bảng 2. Sự phụ thuộc của độ trễ chuẩn hóa vào thông lượng chuẩn hóa

D	0,01	0,02	0,03	0,04	0,05	0,06	0,20	0,40	0,60	0,70	0,75	0,80	0,85
$E[W_{ee}]$ K=10	208,08	208,17	208,25	208,34	208,38	208,47	210,00	213,37	222,20	239,13	269,58		
$E[W_{ee}]$ K=15	208,08	208,17	208,25	208,34	208,38	208,47	210,00	213,33	220,15	228,11	236,30	254,82	
$E[W_{ee}]$ K=20	208,08	208,17	208,25	208,34	208,38	208,47	210,00	213,34	220,01	226,87	232,79	243,07	269,60
$E[W_{ll}]$ K=10	222,41	262,81	376,35	767,24									
$E[W_{ll}]$ K=15	222,41	262,81	376,10	693,71									
$E[W_{ll}]$ K=20	222,41	262,81	376,06	693,71									



Hình 6. Độ trễ chuẩn hóa thay đổi theo thông lượng chuẩn hóa

6. KẾT LUẬN

Trong bài báo này chúng tôi đã nghiên cứu một cách định hướng và so sánh hiệu suất của các tiếp cận link-by-link với cách tiếp cận end-to-end để xử lý các bản tin bị mất hoặc bị hỏng khi chúng được truyền giữa 2 người dùng cuối trong mạng. Kết quả đã nhận được biểu thức giải tích để tính thời gian trễ của gói tin khi đi qua mạng.

Chúng tôi đã đưa ra hai thí dụ với các con số cụ thể, thí dụ thứ nhất là một mạng WAN sử dụng đường truyền tốc độ thấp, tỉ suất lỗi bit cao: thí dụ thứ hai là một mạng WAN sử dụng đường truyền tốc độ cao, tỉ suất lỗi bit thấp. Kết quả nghiên cứu đã chỉ ra rằng trong các mạng tốc độ cao ngày nay, với miền tham số mạng mà chúng ta quan tâm, cách tiếp cận end-to-end để kiểm soát lỗi

cho ta hiệu suất bằng hoặc lớn hơn, trong khi đó lại đòi hỏi ít tài nguyên mạng hơn (tức là các b
đệm và thời gian tính toán) so với cách tiếp cận link-by-link.

TÀI LIỆU THAM KHẢO

- [1] Amit Bhargava, James F. Kurose, Don Towsley, and Guy Vanleemput, Performance comparison of error control schemes in high-speed computer communication networks, *IEEE Journal of Selected Area in Communications* 6 (9) (1988) 1565-1575.
- [2] Mischa Schwartz, *Telecommunication Networks*, Addison Wesley, 1987.
- [3] P. Schweitzer and S. Lam, Buffer overflow in a store and forward network node, *IBM J. Res. Develop.* (1976) 542-550.

Nhận bài ngày 27-6-200

Khoa Công nghệ, Đại học Quốc gia Hà Nội.