

THUẬT TOÁN ĐIỀU KHIỂN TƯƠNG THÍCH CHẤT LƯỢNG DỊCH VỤ TRONG MIDDLEWARE TƯƠNG THÍCH THEO ỨNG DỤNG

VŨ HOÀNG HIẾU, HOÀNG MINH, NGUYỄN NGỌC SAN

I. GIỚI THIỆU

Trong môi trường phân bố, nhiều ứng dụng đa phương tiện như: video theo yêu cầu (VoD), hội nghị truyền hình, giám sát bằng hình ảnh ... cùng chia sẻ nguồn tài nguyên hệ thống hữu hạn và cùng chịu sự biến đổi về độ sẵn sàng của tài nguyên, làm ảnh hưởng đến chất lượng dịch vụ. Điều này đang là thách thức trong môi trường mạng có chia sẻ tài nguyên khi không có sự hỗ trợ QoS trong hệ thống mạng và hệ điều hành ở lớp dưới, nhất là khi không dự báo được biến đổi về độ sẵn sàng của tài nguyên như: bộ xử lý, băng thông mạng và bộ nhớ. Để giải quyết những khó khăn trên các hệ thống middleware phân bố được sử dụng, nhằm hỗ trợ việc phân phối QoS theo yêu cầu của ứng dụng. Trong [1], chúng tôi đã đề xuất một kiến trúc middleware tổng quát để thực hiện tương thích QoS theo ứng dụng và phương pháp điều khiển tác vụ. Kiến trúc và phương pháp này cho phép xử lý tương thích ứng dụng để đạt được mức QoS yêu cầu trong điều kiện tài nguyên biến đổi. Bài viết này trình bày thuật toán điều khiển tương thích QoS cho kiến trúc trên, đánh giá độ ổn định và cân bằng của thuật toán dựa trên những nguyên tắc toán học trong lý thuyết điều khiển truyền thống. Đồng thời chúng tôi trình bày thực nghiệm kiểm chứng tính hợp lý của thuật toán điều khiển.

Nhiều công trình nghiên cứu trong lĩnh vực tương thích QoS như: mô hình tương thích QoS chung [2], tương thích buffering, tương thích smoothing [5], tương thích ưu tiên ... giải quyết vấn đề về cơ chế hệ thống, giao thức truyền dẫn, chính sách tương thích ... Tuy nhiên các công trình này không đưa ra mô hình hay khung mẫu cho quá trình tương thích QoS để có thể diễn giải các đặc tính tương thích như độ ổn định và cân bằng trong việc cung cấp QoS giữa các ứng dụng với nhau. Ngoài ra các giải pháp này cũng không cho phép phát triển các thuật toán tương thích mới trong điều kiện mạng có kích thước lớn và độ tích hợp cao.

Bài viết này gồm các phần: phần II nhắc lại kiến trúc middleware tương thích nhận biết ứng dụng đã đề xuất trong [1]; phần III đề xuất thuật toán điều khiển tương thích QoS trong kiến trúc middleware; phần IV trình bày độ ổn định và cân bằng của thuật toán điều khiển tương thích dựa trên các nguyên tắc toán học trong lý thuyết điều khiển truyền thống; phần V mô tả kết quả thực nghiệm kiểm chứng mô hình lý thuyết; cuối cùng phần VI là kết luận và định hướng nghiên cứu.

II. KIẾN TRÚC MIDDLEWARE TƯƠNG THÍCH NHẬN BIẾT ỨNG DỤNG

Kiến trúc middleware tổng quát cho phép thực hiện tương thích QoS theo ứng dụng trong các hệ thống đa phương tiện phân bố được đề xuất trong [1] gồm hai lớp.

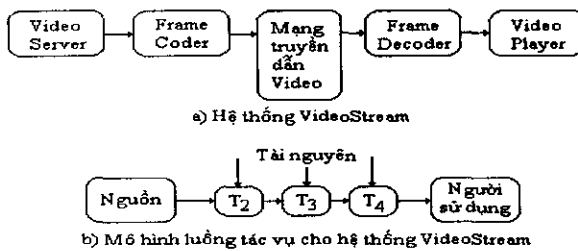
Trong lớp thứ nhất, bộ điều khiển tương thích ứng dụng chung và bộ đánh giá tài nguyên hỗ trợ tương thích tài nguyên lớp thấp bằng cách phát hiện và phản ứng với những thay đổi về độ sẵn sàng của tài nguyên. Trong lớp thứ hai, bộ thao tác ứng dụng cụ thể, các profiler của ứng dụng và bộ đàm phán chịu trách nhiệm thực hiện tương thích chức năng ở mức cao đối với mỗi

ứng dụng cụ thể, bao gồm thời điểm quyết định và kiểu chức năng ứng dụng được phép tham gia vào cung cấp dịch vụ trên cơ sở các yêu cầu của ứng dụng đang chạy. Tương tác giữa các thành phần khác nhau của middleware và ứng dụng được thực hiện thông qua một platform cho phép thực hiện các dịch vụ truyền thông, ví dụ như CORBA.

Với kiến trúc hai lớp như trên bao gồm bộ điều khiển tương thích chung và bộ thao tác ứng dụng cụ thể, hoạt động điều khiển do các thành phần này tạo ra sẽ thực hiện điều khiển việc tương thích chất lượng dịch vụ đối với tất cả các ứng dụng chia sẻ cùng một nguồn tài nguyên ở hệ thống đầu cuối, đồng thời điều khiển tương thích chất lượng dịch vụ đối với từng dịch vụ cụ thể dựa trên những tham số cấu hình được thiết lập trong profiler. Các thành phần của kiến trúc middleware như trên phù hợp với các thành phần trong mô hình chung cho tương thích QoS đã được đề xuất trong [2].

III. THUẬT TOÁN ĐIỀU KHIỂN TƯƠNG THÍCH QOS TRONG KIẾN TRÚC MIDDLEWARE

Trong mô hình thành phần chức năng [3], ứng dụng là tập hợp thành phần chức năng. Tác vụ là một thành phần chức năng thực hiện nhiệm vụ cụ thể để tạo ra kết quả cho tác vụ tiếp. Hình 1a mô tả thành phần chức năng của ứng dụng videostream như mã hóa khung, truyền tín hiệu ... các thành phần chức năng này yêu cầu tài nguyên để thực hiện chức năng xử lý và tạo ra một giá trị điều khiển chức năng tiếp theo.



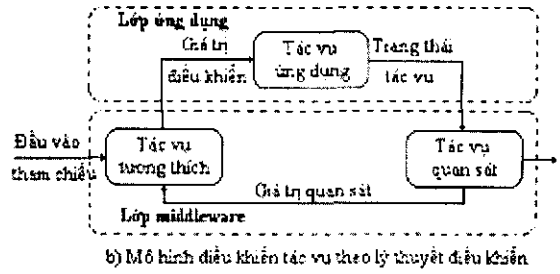
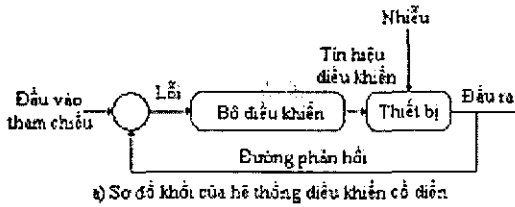
Hình 1. Mô hình luồng tác vụ

Với thành phần chức năng như trên, hình 1b mô tả mô hình luồng tác vụ cho hệ thống videostream, mỗi tác vụ là một thành phần chức năng và mỗi ứng dụng được chuyển đổi thành một biểu đồ luồng tác vụ có hướng không khép kín của nhiều tác vụ. Liên kết có hướng từ tác vụ T_i sang tác vụ T_j chỉ ra tác vụ T_j sử dụng kết quả của tác vụ T_i . Để thực hiện tác vụ T_i hệ thống phải cung cấp lượng tài nguyên nhất định.

1. Mô hình tổng quát cho tác vụ ứng dụng

Mô hình điều khiển truyền thống [4] được mô tả như hình 2a. Trong đó “Thiết bị” được điều khiển bởi một “Bộ điều khiển”, trạng thái bên trong của thiết bị này được xác định bởi bộ điều khiển theo một thuật toán điều khiển xác định. Thuật toán điều khiển tạo ra tín hiệu điều khiển bằng cách so sánh trạng thái bên trong của thiết bị mà nó quan sát được với giá trị “Đầu vào tham chiếu”.

Mô hình điều khiển tác vụ tập trung vào một tác vụ cụ thể (T_i). Khi thực hiện tác vụ này, hệ thống phải cung cấp lượng tài nguyên nhất định. Với nguồn tài nguyên hữu hạn, hệ thống có thể đáp ứng hoặc không đáp ứng được yêu cầu tài nguyên này. Vì vậy số yêu cầu tài nguyên của một tác vụ cần phải được điều khiển tương thích với độ sẵn sàng của tài nguyên hệ thống.



Hình 2. Mô hình điều khiển tác vụ

Áp dụng mô hình điều khiển truyền thống cho một tác vụ ứng dụng cụ thể để nhận được mô hình điều khiển tác vụ, như hình 2b, gồm các thành phần sau:

(1) *Tác vụ ứng dụng*: thực hiện một chức năng cụ thể của ứng dụng và được điều khiển tương thích.

(2) *Tác vụ tương thích*: thực hiện thuật toán điều khiển tương thích và tạo ra các báo hiệu điều khiển ứng dụng cụ thể để tương thích chất lượng.

(3) *Tác vụ quan sát*: thực hiện quan sát và đánh giá trạng thái của tác vụ ứng dụng và báo hiệu cho tác vụ tương thích. Trạng thái tác vụ của tác vụ ứng dụng được xác định bởi các tham số bên trong tương ứng.

Biến đổi động yêu cầu tài nguyên của tác vụ ứng dụng được mô tả bằng phương trình có dạng sau [4]:

$$\frac{dx(t)}{dt} \cong \dot{x}(t) = f[x(t), u(t), w(t), t] \quad (1)$$

$$z(t) = h[x(t), v(t), t]. \quad (2)$$

Với mô hình trên, các yêu cầu tài nguyên khi thực hiện một ứng dụng được coi là cân bằng nếu:

$$\dot{x}(t) = 0 = f[x(t), u(t), w(t), t] \quad (3)$$

ở đây x là vector trạng thái của tác vụ, u là vector tham số đầu vào có thể điều khiển được, z là vector tham số đầu ra quan sát được, w là các biến không điều khiển được và v là vector lỗi quan sát.

2. Mô hình rời rạc cho tác vụ ứng dụng

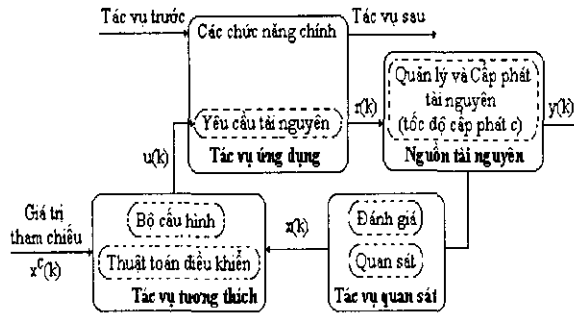
Mô hình tổng quát trên (có thể phi tuyến và biến đổi theo thời gian) là quá phức tạp để xem xét một thuật toán điều khiển phù hợp. Để đơn giản hàm biến đổi động về yêu cầu tài nguyên của tác vụ ứng dụng được đơn giản hóa bằng các hàm tuyến tính rời rạc và yêu cầu tài nguyên của ứng dụng được mô tả như sau:

$$x(k) = \Phi x(k-1) + \Gamma u(k-1) + w(k-1) \quad (4)$$

$$y(k) = Hx(k) \quad (5)$$

$$z(k) = y(k) + v(k) \quad (6)$$

ở đây $k = 1, \dots, k_{\max}$ là các khoảng thời gian và Φ, Γ, H là các ma trận truyền đạt không có lỗi của hệ thống.



Hình 3. Mô hình rời rạc cho tác vụ ứng dụng

Với mô hình yêu cầu tài nguyên của ứng dụng như trên, mô hình điều khiển tác vụ rời rạc được mô tả như hình 3. Trong mô hình rời rạc, chúng tôi sử dụng các kí hiệu sau:

- (1) t_c là khoảng thời gian lấy mẫu, chu kỳ lấy mẫu trong khoảng $[k, k+1]$, k là hằng số thời gian.
- (2) $r_i(k)$ và $u_i(k)$ là số tài nguyên gốc và cho phép trong khoảng $[k, k+1]$ bởi tác vụ tương thích của T_i .
- (3) c là tổng yêu cầu được cấp phép, $x(k)$ là tổng yêu cầu tài nguyên trong hàng đợi tại thời điểm k .
- (4) $M(k)$ là tổng số tác vụ ứng dụng tích cực cạnh tranh tài nguyên trong hệ thống.
- (5) $A(k)$ và $N(k)$ là tập các tác vụ ứng dụng phải và không phải tương thích ở thời điểm k .
- (6) $l(k)$ là số lượng tác vụ trong $A(k)$ (nghĩa là $l(k) = |A(k)|$), $M(k) - l(k)$ là tổng số các tác vụ trong $N(k)$ (nghĩa là $|N(k)| = M(k) - l(k)$). Giả thiết cả $M(k)$ và $l(k)$ là hằng số trong khoảng thời gian $[k, k+1]$.
- (7) ω_i là trọng số tĩnh chỉ ra tầm quan trọng của tác vụ T_i , $\gamma_i(k)$ là trọng số động tương ứng.

Giả thiết tại thời điểm k có $l(k)$ tác vụ được tương thích bởi các tác vụ tương thích ($u_i(k) < r_i(k)$) và $M(k) - l(k)$ tác vụ không phải tương thích ($u_i(k) = r_i(k)$), tổng số yêu cầu tài nguyên trong hàng đợi của hệ thống được mô tả như sau [4]:

$$x(k) = \Psi_{c_{\max}} \left\{ x(k-1) + \sum_{T_i \in A(k-1)} u_i(k-1) + \sum_{T_i \in N(k-1)} r_i(k-1) - c \right\} \quad (7)$$

$$= \Psi_{c_{\max}} \left\{ x(k-1) + l(k-1) \bar{u}(k-1) + R(k-1) - c \right\} \quad (8)$$

ở đây $R(k) = \sum_{T_i \in N(k)} r_i(k)$ là tổng yêu cầu tài nguyên của các tác vụ T_i thỏa mãn $u_i(k) > r_i(k)$; $\bar{u}(k)$

là yêu cầu tài nguyên trung bình của các tác vụ T_i thỏa mãn $u_i(k) < r_i(k)$ và $u_i(k) = \gamma_i(k) l(k) \bar{u}(k)$.

3. Thuật toán điều khiển tương thích

Mô hình rời rạc của tác vụ ứng dụng được mô tả trong phương trình (8). Tương tự lí thuyết điều khiển truyền thống, mục đích điều khiển tác vụ là duy trì tổng số yêu cầu tài nguyên trong hàng đợi $x(k)$ của ứng dụng nằm xung quanh giá trị tham chiếu $x^c(k)$. Để thực hiện, tác vụ tương

thích phải thực hiện một thuật toán điều khiển dựa trên yêu cầu tài nguyên trong hàng đợi $x(k)$ và giá trị tham chiếu $x^c(k)$ để đưa ra tín hiệu điều chỉnh yêu cầu tài nguyên $u_i(k)$. Theo tính toán trong [4] tín hiệu điều chỉnh yêu cầu tài nguyên $u_i(k)$ của tác vụ tương thích T_i thỏa mãn $x(k)$ nằm xung quanh giá trị tham chiếu $x^c(k)$ được mô tả bởi phương trình sau:

$$u_i(k) = \Psi_{r_i, \beta_i} \{u_i(k-1) + \alpha(x^c(k) - x(k)) + \beta[(x^c(k) - x(k)) - (x^c(k-1) - x(k-1))]\} \quad (9)$$

ở đây α và β là các hệ số cấu hình của tác vụ tương thích.

IV. ĐỘ ỔN ĐỊNH VÀ CÂN BẰNG CỦA THUẬT TOÁN ĐIỀU KHIỂN TƯƠNG THÍCH

Kết hợp phương trình (8) và (9) ta có đặc tính đầy đủ của hệ thống điều khiển tương thích. Trường hợp lí tưởng $x(k)$ sẽ hội tụ về giá trị tham chiếu $x^c(k)$, khi đó $u_i(k)$ hội tụ tới trạng thái cân bằng cho mỗi tác vụ cạnh tranh cùng một nguồn tài nguyên. Giả thiết trong khoảng $[k_1, k_2]$, các giá trị $x^c(k)$, $l(k)$, $M(k)$ và $R(k)$ cân bằng ổn định về các trạng thái tương ứng x_s^c , l_s , M_s và R_s . Khi đó hệ thống tương thích có đặc tính sau:

1. Độ cân bằng của hệ thống

Đặc tính 1: Nếu hệ thống tương thích được thiết lập bởi các phương trình (8) và (9) thì trong khoảng thời gian $[k_1, k_2]$ số lượng các yêu cầu tài nguyên trong hàng đợi $x(k)$ sẽ hội tụ tới một giá trị cân bằng tương ứng với giá trị tham chiếu x_s^c . Ngoài ra hệ thống cũng cân bằng yêu cầu tài nguyên giữa các tác vụ cạnh tranh theo trọng số tĩnh ω_i của chúng.

Đặc tính trên của hệ thống tương thích có thể được chứng minh như sau: Giả sử x_s và \bar{u}_s là các giá trị cân bằng tương ứng cho hệ thống được thiết lập bởi phương trình (8) và (9). Khi đó ta có:

$$x_s = \Psi_{C_{\max}} \{x_s + l_s \bar{u}_s + R_s - c\} \quad (10)$$

$$\gamma_i l_s \bar{u}_s = \Psi_{r_i} \{\gamma_i l_s \bar{u}_s + \alpha(x_s^c - x_s)\}. \quad (11)$$

Bỏ qua trường hợp ngưỡng, giải phương trình (10) và (11) ta có:

$$x_s = x_s^c \quad (12)$$

$$\bar{u}_s = \frac{c - R_s}{l_s}. \quad (13)$$

Giả sử tập các tác vụ được tương thích ổn định là A_s , phương trình (13) có thể được viết lại đối với tác vụ T_i ở điểm cân bằng như sau:

$$(u_i)_s = \gamma_i l_s \bar{u}_s = \frac{\omega_i l_s}{\sum_{T_j \in A_s} \omega_j} \frac{c - R_s}{l_s} = \frac{\omega_i l_s}{\sum_{T_j \in A_s} \omega_j} \left\{ \frac{c}{M_s} + \frac{(M_s - l_s) \frac{c}{M_s} - R_s}{l_s} \right\} \quad (14)$$

ở đây $(u_i)_s$ là điểm cân bằng ổn định của $u_i(k)$ và phụ thuộc vào trọng số tĩnh ω_i .

Các giá trị x_s^c và $(u_i)_s$ là các điểm cân bằng của hệ thống và của từng tác vụ T_i .

2. Độ ổn định của hệ thống

Đặc tính 2: Một hệ thống tương thích được thiết lập bởi phương trình (8) và (9) ổn định tiệm cận đối với tác vụ T_i xung quanh giá trị lân cận trong điều kiện $\alpha > 0$, $\beta > 0$ và $\alpha + 2\beta < 4\gamma_i$.

Để chứng minh điều này chúng ta định nghĩa

$$e(k) = x^c(k) - x(k) \quad (15)$$

$$\hat{u}_i(k) = \gamma_i l(k) \left[\bar{u}(k) - \frac{c - R(k)}{l(k)} \right]. \quad (16)$$

Thay phương trình (15) và (16) vào phương trình (8) và (9) để nhận được các phương trình sau:

$$\hat{u}_i(k) = \hat{u}_i(k-1) + \alpha e(k) + \beta[e(k) - e(k-1)] \quad (17)$$

$$x(k) = x(k-1) + \frac{1}{\gamma_i} \hat{u}_i(k-1) \quad (18)$$

Để nhận được các điều kiện ổn định, chúng ta thực hiện biến đổi z đối với phương trình (17) và (18) và thu được phương trình đặc tính sau:

$$z^2 + \left(\frac{\alpha}{\gamma_i} + \frac{\beta}{\gamma_i} - 2 \right) z - \left(\frac{\beta}{\gamma_i} - 1 \right) = 0. \quad (19)$$

Theo [4], để hệ thống ổn định, nghiệm của phương trình (19) phải nằm trong miền ổn định là vòng tròn đơn vị. Có thể chứng minh rằng các đặc tính này duy trì nếu điều kiện sau được đáp ứng:

$$\alpha > 0, \beta > 0 \text{ và } \alpha + 2\beta < 4\gamma_i. \quad (20)$$

Phương trình (20) chính là điều kiện ổn định của hệ thống điều khiển tương thích.

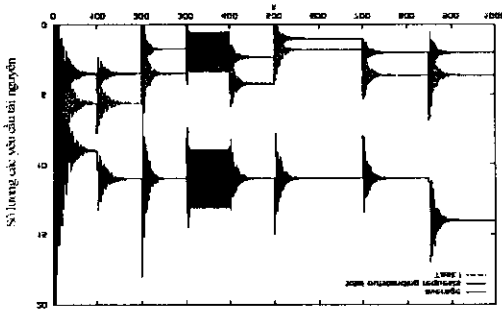
V. KẾT QUẢ THỰC NGHIỆM

1. Cấu hình hệ thống thử nghiệm

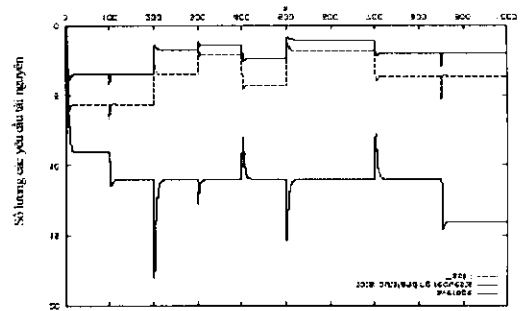
Để kiểm tra khả năng ứng dụng mô hình điều khiển tác vụ trong kiến trúc middleware nhận biết ứng dụng, mô phỏng thực nghiệm cho hệ thống tương thích được thực hiện. Trong thực nghiệm này, cả kiến trúc middleware tương thích nhận biết ứng dụng và mô hình điều khiển tương thích tác vụ được thực hiện trên nền Windows NT 4.0 sử dụng Visual C++ 6.0. Ngoài trừ việc truyền thông giữa các bộ đàm phán của kiến trúc middleware tương tác với nhau thông qua socket mạng chuẩn, tất cả các thành phần khác tương tác với nhau thông qua platform CORBA (như đã mô tả trong [1]).

2. Ảnh hưởng của các tham số cấu hình đến số lượng yêu cầu tài nguyên của hệ thống

Với mô hình thực nghiệm kiến trúc middleware có sử dụng bộ điều khiển tác vụ trên nền Window NT như trên, khi thay đổi các tham số cấu hình α và β của bộ tương thích mức độ ổn định của hệ thống cũng thay đổi theo.



(a) Hệ số cấu hình $\alpha = 0,7$ và $\beta = 0,25$



(b) Hệ số cấu hình $\alpha = 0,1$ và $\beta = 0,45$

Hình 4. Ảnh hưởng của tham số cấu hình

Kết quả thực nghiệm chỉ ra trong hình 4, trong đó mô tả ảnh hưởng của các hệ số cấu hình α và β đến độ ổn định của hệ thống middleware. Trong hình 4a, với các hệ số cấu hình α là cao, β là thấp hệ thống thể hiện các dao động tắt dần. Đặc biệt trong khoảng thời gian [300, 400] hệ thống mất ổn định và dao động. Hình 4b cho thấy hệ thống tương thích cân bằng ổn định không có dao động và đáp ứng tương thích trong khoảng thời gian ngắn. Như vậy để hệ thống cân bằng, ổn định phải lựa chọn hệ số cấu hình α và β phù hợp.

VI. KẾT LUẬN VÀ ĐỊNH HƯỚNG NGHIÊN CỨU

Bài viết này đã trình bày: (1) thuật toán điều khiển cho kiến trúc middleware tương thích QoS theo ứng dụng; (2) xác định điểm cân bằng và độ ổn định của thuật toán. Trên cơ sở mô hình toán học và thuật toán điều khiển này, bộ tương thích trong kiến trúc middleware điều khiển tương thích ứng dụng được thiết kế để có thể tương thích với hai kiểu tài nguyên mạng là băng thông mạng và tải CPU.

Kết quả thực nghiệm kiến trúc middleware điều khiển tương thích QoS và mô hình điều khiển tương thích đã chứng minh tính hợp lý của mô hình điều khiển này.

Trong bài viết chúng ta mới chỉ xem xét trường hợp các biến đổi động yêu cầu tài nguyên và thuật toán điều khiển là tuyến tính. Tuy nhiên trong thực tế biến đổi động yêu cầu tài nguyên có thể là phi tuyến. Vì vậy một hướng nghiên cứu tiếp theo của bài viết này là nghiên cứu trường hợp các biến đổi tài nguyên và thuật toán điều khiển trong các điều kiện phi tuyến.

TÀI LIỆU THAM KHẢO

1. V. H. Hieu and H. D. Hai - An Application-aware Adaptive Middleware Architecture for Distributed Multimedia Systems, The First International Conference on Communications and Electronics (ICCE2006), HaNoi - VietNam, Oct-2006, pp.141-146.
2. D.Waddington and D.Hutchison - A General Model for QoS Adaptation, 6th IWQoS Proceedings Napa Valley California, May 18-20 1998, pp.275-278.
3. D. Hull, A. Shankar, K. Nahstedt and J. Liu - An End-to-End QoS Model and Management Architecture, In: Proceedings of IEEE Workshop on Middleware for Distributed Real-Time Systems and Services, Dec. 1997, pp.82-89.
4. G. Franklin and J. Powell - Digital Control of Dynamic Systems, Addison-Wesley, 1981.

5. G. Cao, W. Feng and M. Singhal - Online VBR Video Traffic Smoothing, In: Proceedings of 8th IEEE International Conference on Computer Communications and Networks, 1999, pp.502-507.

SUMMARY

QOS ADAPTIVE CONTROL ALGORITHM FOR APPLICATION-AWARE ADAPTIVE MIDDLEWARE

In a distributed environment, many applications compete and share a limited amount of system resources. Due to the statistical share of resources to multiple applications and a dynamic number of application tasks, the resource availability is in fluctuation and unpredictability. In order to guarantee QoS, these applications have to suffer from variations in resource availability and adapt their behaviors. In [1], we proposed an application-aware adaptive middleware architecture which allows controlling adaptive behaviors of applications and achieving the desired QoS level. This paper proposes a novel QoS adaptive control algorithm for this architecture, examines the stability and equilibrium of this algorithm using the classical control theory principles. The simulation result shows the validation of our algorithm.

Địa chỉ:

Nhận bài ngày 10 tháng 6 năm 2006

Học viện Công nghệ Bưu chính Viễn thông.