Effective congestion control for QoS enhancement of self-similar multimedia traffic

H. Yin, C. Lin, G. Min and X. Chu

Abstract: Owing to the existence of noticeable concentrated periods of contention and idleness, self-similar traffic can greatly increase packet delay and loss probability and thus reduce system resource utilisation. The development of efficient congestion control mechanisms plays a central role in the improvement of network quality of service (QoS), in particular for real-time multimedia applications. By exploiting the property of scale-invariant burstiness and correlation inherent in self-similar traffic, the authors propose an effective congestion control scheme, named adaptive wavelet and probability-based scheme (AWP), which concurrently operates over multiple time scales. AWP adopts the extended multifractal wavelet model (EMWM) for analysing estimated traffic volume across multiple time scales. Furthermore, a new auto-correction algorithm based on Bayes' theory for confidence analysis is employed to examine the validity of the predicted information. The analysis results can be used to enhance the adaptability of the prediction algorithm. In particular, the AWP framework can be easily extended to more than two time scales by increasing the level of wavelet transforms, which brings AWP a natural advantage in implementation and scalability. A series of simulation experiments have demonstrated that the proposed AWP scheme is superior to TCP and TFRC as it can greatly improve the QoS of multimedia data transmission while avoiding congestion collapse on the network.

1 Introduction

Recently, the internet has been experiencing an explosive growth in multimedia applications, such as voice-over-IP and video conferencing, which require high bandwidth for maintaining desired quality of service (QoS) including an upper bound on end-to-end latency and jitter. However, it is impossible for the current internet to guarantee the QoS for multimedia applications as the influx of large amounts of multimedia traffic into the internet may cause serious network congestion. To resolve these problems, many congestion control schemes have been proposed and can be divided into two basic categories: windows-based congestion control [1] and rate-based congestion control [2, 3]. Current real-time streaming applications in the internet typically rely on rate-based congestion control including TCP-friendly rate control (TFRC) owing to its ability to generate smooth and TCP-friendly data flow [4]. Nevertheless, most of the research work on rate-based congestion control is performed to improve smoothness of output rate and TCP friendliness, but pays inadequate attention to the exploitation of traffic characteristics for improving the performance of congestion control mechanisms.

Many recent high-quality measurement studies [5–7] have demonstrated that internet traffic exhibits noticeable self-similar properties (i.e. scale-invariant burstiness and correlation). Due to the existence of noticeable concentrated periods of contention and idleness, self-similar traffic can greatly increase packet delay and loss rate and thus reduce system resource utilisation. The objective of this paper is to exploit the self-similar properties of network traffic for the design of a novel rate-based congestion control scheme in order to maximise the QoS of multimedia applications while preventing congestion collapse of the network.

To improve the performance of congestion control of self-similar traffic, Park *et al.* [8] proposed a two-level multiple time scale congestion control protocol (MTSC) which experiences congestion control across typically two time-scales. Although self-similar properties are applied for the development of such a congestion control scheme, it does not take the demand of multimedia data delivery into account and relies on empirical calculation when handling network traffic. Ribeiro *et al.* [9] developed a model-based technique called Delphi algorithm for inferring the instantaneous competition volume of traffic across an end-to-end network path. It uses the multifractal wavelet model (MWM), a parsimonious multifractal parametric model, to capture multi-scale statistical properties and queuing behaviour of network traffic. This algorithm provides accurate cross-traffic estimation when the bandwidth utilisation is high; however it overestimates the cross-traffic when the bandwidth utilisation becomes low. As a result, the MWM cannot respond properly to severe network traffic changes due to lack of adaptation to network variation.

Building upon the aforementioned research efforts, this study proposes a novel adaptive scheme for congestion control of multimedia traffic across multiple time scales, which is called the adaptive wavelet and probability-based

© The Institution of Engineering and Technology 2006

IEE Proceedings online no. 20050671

doi:10.1049/ip-com:20050671

Paper first received 22nd December 2005 and in revised form 19th March 2006

H. Yin and C. Lin are with the Department of Computer Science, Tsinghua University, 100084, Beijing, People's Republic of China

G. Min is with the Department of Computing, School of Informatics, University of Bradford, Bradford BD7 1DP, UK

X. Chu is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong

E-mail: h-yin@mail.tsinghua.edu.cn

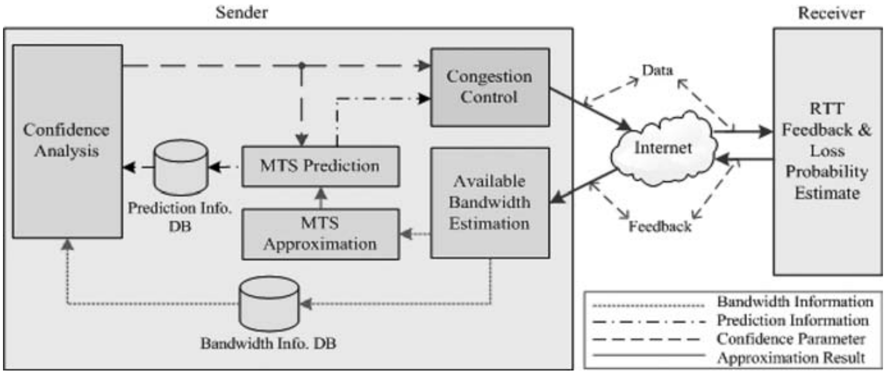

Fig. 1 *Brief view of AWP framework*

(AWP) scheme. AWP refers to the framework of MTSC and enhances its performance by adding the new adaptive mechanism, an improved traffic prediction model and a rate control algorithm. As a consequence, AWP not only provides better controllability and adaptability but also is more appropriate to multimedia data delivery. Furthermore, AWP extends the traditional MWM model using an adaptive algorithm proposed by this research. This extended multifractal wavelet model (EMWM) is able to predict available network bandwidth over multiple time-scales and can improve the flexibility of AWP under various network environments.

AWP is a two-level multiple time-scale congestion control scheme for self-similar multimedia traffic in modern high-speed networks. On the normal time-scale (20–200 ms), implicit prediction of available bandwidth obtained from feedback is used to support smooth and TCP-friendly rate-based feedback congestion control; on the large time-scale (1–5 s), explicit prediction for detecting bandwidth changes in overall network contention is applied to modulate the rate control exhibited on the normal time-scale. Unlike the aforementioned MTSC scheme, AWP adopts the wavelet transform-based EMWM model for analysing the estimated traffic volume across multiple time-scales. Furthermore, a new auto-correction algorithm based on Bayes' theory for confidence analysis is employed to examine the validity of the predicted information. The analysis results can be used to enhance the adaptability of the prediction algorithm. In particular, the AWP framework can be easily extended to more than two time-scales by increasing the wavelet transform level, which brings AWP a natural advantage in implementation and scalability.

A series of simulation experiments have been carried out in order to compare the relative performance merits of AWP, TCP and TFRC. The results have shown the unique advantages for AWP including: (1) the network bandwidth utility is greatly improved along with the low packet loss probability since the self-similar properties of network traffic are exploited in the design of the congestion control scheme; (2) AWP contains good adaptation to the variation of network environments; (3) AWP preserves smoothness and TCP-friendliness of output rate, which are very important for the QoS improvement of multimedia application and network stability.

2 Overview of AWP congestion control scheme

This section outlines the framework of the AWP scheme. The detailed algorithms to implement this scheme will be presented in the next section. The AWP consists of the sender side and the receiver side, as illustrated in Fig. 1, where MTS stands for multiple time-scale and DB for database.

2.1 Sender functionality

The overall congestion control starts with the sender side to collect the feedback information from the receiver side on the normal time-scale at millisecond level. The feedback is used to estimate the available end-to-end bandwidth. According to the TCP throughput formula, the available bandwidth A (in bytes/s) is a function of packet size s, round-trip time R, steady-state loss event probability p, and the TCP retransmission timeout value t_{RTO}. The available bandwidth can be expressed by [4]

$$A = \frac{s}{R\sqrt{2p/3} + t_{RTO}(3\sqrt{3p/8})\,p(1+32p^2)} \qquad (1)$$

The estimated instantaneous available bandwidth A is imported into the multiple time-scale approximation (MTSA) module where an extended multifractal wavelet model (EMWM) is used to inosculate current and past available bandwidth. The inosculation results are inputted to the multiple time-scale prediction (MTSP) module for predicting future network traffic on the large time-scale at second level. The instantaneous volumes of A form a series of available bandwidths $A(t)$ at different instants t:

$$A(t),\ t = \omega, \omega + \varepsilon, \ldots, \omega + T, \omega \geq 0, T \geq 0 \qquad (2)$$

where ω is the start time of the ith control interval, ε is the sample period of estimating the available end-to-end bandwidth, and T is the duration in which MTSA is performed. T also represents the granularity of the large time-scale of congestion control. The time schedule of control intervals at the sender is depicted in Fig. 2.

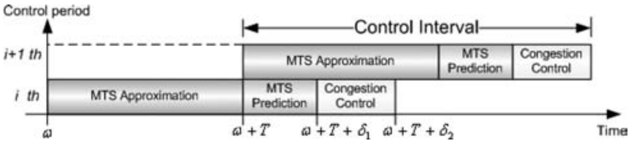

Fig. 2 *Time schedule of control intervals of AWP*

AWP runs control intervals periodically. Each control interval consists of the time T for running MTSA module, δ_1 for MTSP and δ_2 for the congestion control module, where $\delta_1 + \delta_2 < T$. At time $\omega + T$ of the ith control interval, the work of MTSA is completed. Based on the inosculation results produced by MTSA, the MTSP module starts to generate a new time series $B(t)$, which involves the values of predicted available bandwidth for the $(i+1)$th control interval:

$$B(t),\ t = \omega + T,\ \omega + T + \varepsilon,\ \ldots,\ \omega + 2T,\ \omega \geq 0,\ T \geq 0 \tag{3}$$

The congestion control module of the ith control interval starts to work at time $\omega + T + \delta_1$, whereas the MTS approximation of the next control interval begins at time $\omega + T$. $A(t)$ is stored in the bandwidth information database for further use for confidence analysis, and $B(t)$ is stored in the prediction information database for the purpose of congestion control and adaptation of the MTS approximation.

Adaptation is another important feature of AWP. It is achieved through confidence analysis based on Bayes' theorem [10, 11]. By analysing the series $A(t)$ and $B(t)$, the confidence analysis module generates the confidence level denoted by α, which is used to evaluate the confidence of the predicted available bandwidth from the MTSP module. α is then transferred into both the congestion control module and MTSP module, in order to guide the congestion control algorithm in the congestion control module and to auto-correct parameters used in the prediction algorithm in the MTSP module.

AWP performs congestion control at time $\omega + T + \delta_1$ (as shown in Fig. 2). On the large time-scale, the congestion control module makes the control decision according to confidence parameter α and the expectation of $B(t)$. The control strategy is:

$$\left.\begin{array}{l}\text{if } \alpha > \theta \text{ then } \Lambda = E(B(t)) \\ \\ \text{else wait for next prediction}\end{array}\right\} \tag{4}$$

where θ is a threshold to decide whether the prediction information provided by $B(t)$ is credible, Λ is the expected throughput on the large time-scale for the next control interval (the $(i+1)$th control interval in Fig. 2), and $E(B(t))$ is the expectation of $B(t)$. On the normal time-scale, the congestion control algorithm performs TCP-friendly and smooth rate control, and the controlled throughput λ is kept consistent with Λ. The choice of θ and the detailed congestion control algorithm will be discussed in Section 3.3.

2.2 Receiver functionality

The receiver side needs to estimate the probability p of packet loss. Assume that N packets are sent out within a round-trip time (RTT) and individual packets follow a Bernoulli loss model. In such a model, each link independently drops packets with a fixed probability P_{loss}. The probability of successful packet delivery is $1 - p = (1 - p_{loss})^N$. The probability of packet loss can be further written as [4]

$$p = 1 - (1 - p_{loss})^N \tag{5}$$

The receiver then sends the feedback information including the packet loss probability and the time stamp of the latest received packet to the sender. The packet loss probability p is essential for the control procedure. The sender will use the feedback information to calculate the RTT and estimate the available bandwidth at the sender side, as shown in (1).

3 Algorithms for implementing AWP

This section will present the key algorithms of AWP including: (1) multiple time scale prediction, (2) confidence analysis, and (3) congestion control algorithm.

3.1 Multiple time-scale prediction for self-similar traffic

The self-similar properties of network traffic are related to scale-invariant burstiness and correlation, long-range dependence (LRD) and heavy-tailed distribution of file sizes on the networks [12, 13], which can cause significant increase in queuing delay and packet loss. Therefore it is necessary to take self-similar properties into account when designing a network traffic model.

An important contribution of this study is to propose a new extended multifractal wavelet (EMWM) model for approximating the network traffic. Based on the Haar wavelet system [14], the EMWM can properly portray the self-similar properties of network traffic [9]. Figure 3 shows a part of the binary tree of scaling coefficients from normal time-scale to large time-scale of the EMWM. The root node (denoted by $U_{0,0}$ in Fig. 3) represents the scaling coefficient on the large time-scale, and the leaf nodes $U_{j+2,4k+i}$, $(i = 0, 1, \ldots, 2^{\lceil \log_2(L) \rceil})$ represent the scaling coefficients on the normal time-scale. The scaling coefficient is a measure of the local mean of available bandwidth over the corresponding time-scale.

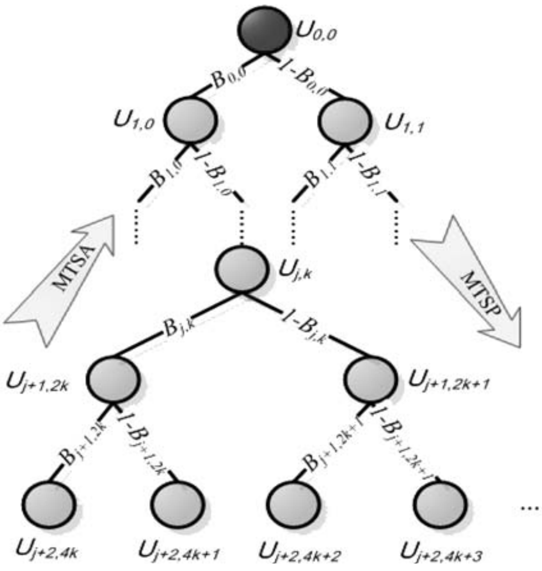

Fig. 3 *Binary tree of scaling coefficients from normal time-scale to large time-scale*

As defined in Section 2.1, $A(t)$ is the estimated available bandwidth series. If $A(t)$ contains L elements, the binary tree depth is $\log_2 2L + 1$ because the left half of the binary tree leaf nodes is about the data source of MTSA (corresponding to $A(t)$) and the right half is the prediction result of MTSP (corresponding to $B(t)$). Using the following equations we can apply wavelet transform [15] onto $A(t)$:

$$U_{j,k} = W(A(t)) = \int A(t) \cdot 2^{j/2} \phi_{j,k}(2^j t - k) dt \tag{6a}$$

$$A(t) = W^{-1}(U_{j,k}) = \frac{2^{-j/2}}{\phi_{j,k}(2^j t - k)} \cdot \frac{dU_{j,k}}{dt} \tag{6b}$$

where $U_{j,k}$ measures the local mean of available bandwidth around time $2^{-i}k$ and $\phi_{j,k}(t)$ is the bandpass wavelet function. Thus $U_{j+2,4k+i}, (i = 0, 1, \ldots, 2^{\lceil \log_2(L) \rceil})$ are the scaling coefficients of $A(t)$. Note that if $L \neq 2^m$ $(m = \lceil \log_2(L) \rceil)$, then zero elements are appended to $A(t)$ until L reaches 2^M. In the binary tree structure, the following equations hold [9]:

$$U_{j,k} = U_{j+1,2k} + U_{j+2,2k+1} \tag{7}$$

$$U_{j+1,2k} = B_{j,k} \times U_{j,k}, \quad U_{j+1,2k+1} = (1 - B_{j,k}) \times U_{j,k} \tag{8}$$

Equations (7) and (8) show that the coefficients on the large time-scale (denoted as parent node) can be calculated by the sum of those on the normal time-scale (children nodes), and the coefficients on the normal time-scale can be obtained by splitting the parent node with random multipliers $B_{j,k}$. For the sake of simplicity, symmetric beta distribution is chosen for $B_{j,k}$ [10, 14]. To ensure that the prediction result from the MTSP module is reasonable, the independent random variables $B_{j,k}$ should be limited to the area $[-1, 1]$.

Given that $A(t)$ contains L elements, from the ith $(1 \leq i \leq \log_2 L)$ scale of the binary tree it takes $L/2^i$ steps of calculation in the bottom-up accumulation procedure. The total number of computational steps required from the left-half leaves to the root node is therefore

$$\frac{L}{2} + \frac{L}{2^2} + \frac{L}{2^3} + \cdots + \frac{L}{2^{\log_2 L}} = \sum_{i=1}^{\infty} \frac{L}{2^i} - \sum_{i=\log_2(L)+1}^{\infty} \frac{L}{2^i}$$
$$= L\left(1 - \frac{1}{L}\right) = L - 1$$

The number of computational steps of the prediction procedure varies according to the prediction resolution. Supposing the prediction resolution is j $(1 \leq j \leq \log_2 L)$, e.g. the jth scale down from the root, then the number of computational steps is $2 + 2^2 + \cdots + 2^{j-1} = 2^j - 2$.

In the MTSA module, the coefficients of $A(t)$ on the normal time-scale are accumulated upward using (7) to obtain traffic load information on the large time-scale. Meanwhile, the scaling coefficients at different scales in the binary tree structure are used to obtain the parameters j and k of the random variable $B_{j,k}$. The inosculation results are then sent to the MTSP module.

In the MTSP module, the coefficient on the large time-scale is split downward by the obtained random multipliers $B_{j,k}$ to calculate those on the normal time-scale. These coefficients are applied by the inverse wavelet transform to generate the prediction series $B(t)$. In Fig. 3, the right half of the binary tree represents the predicted scaling coefficients of $B(t)$ after the wavelet transform is carried out. With the inverse transform using (6b), the elements of $B(t)$ are converted from the wavelet domain to the time domain, which is available bandwidth predicted on the normal time-scale for congestion control purposes.

However, the prediction results from the EMWM are not always valid as the aggregation level of dataflow in the network varies from time to time. It is necessary to make corrections to adjust the prediction results by virtue of the history information, so as to maintain the error of the predicted series at an acceptable level. The coefficients on the normal time-scale are adjusted by referring to the confidence parameter and the latest predicted information:

$$B(t+T) = \alpha \times W^{-1}(U_{j+2,4k+i}) + (1-\alpha) \times B(t),$$
$$t \in [\omega, \omega + T], i \in [2^{j-1}, 2^j - 1] \tag{9}$$

where $B(t+T)$ is the predicted available bandwidth series from time $t = \omega + T$ to $\omega + 2T$. $W^{-1}(U)$ is the inverse wavelet transform of U. Equation (9) shows that the new prediction series weights more in $B(t+T)$ as the confidence α approaches 1, whilst the history prediction series weights more as α approaches 0; we can find that $B(t+T) = W^{-1}(U_{j+2,4k+i})$ if $\alpha = 1$, and thus $B(t+T) = B(t)$ if $\alpha = 0$.

The overhead of corrective actions is mainly determined by the inverse wavelet transform. The cost of this transform is dependent on the exponential computation evolved in (6b), which can be easily implemented by the shift of binary operation. As a result, the overall overhead of the corrective operation is acceptable for online estimation.

3.2 Confidence analysis

The confidence analysis of bandwidth prediction in AWP is conducted using Bayes' theorem [10] which has the following form:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} = P(M)\frac{P(D|M)}{P(D)} \tag{10}$$

AWP is designed for the network environment where the traffic exhibits self-similar properties and the EMWM is effective under this traffic condition. Through experimental analysis, we found that the EMWM holds validity for more than 80% of the overall running time of the experiments, so the prior probability $P(M)$, which is determined by the theory model EMWM, is set to a constant value (i.e. 0.8), for a default high validity of the model.

$P(D)$ is the probability that $A(t)$ exhibits second-order self-similarity. For the discrete time series $A(t)$ with length L, the R/S plot method [12] is employed to determine the Hurst parameter H. Define the ratio R/S as

$$\frac{R}{S} = \frac{\max\limits_{1 \leq j \leq L}\left[\sum\limits_{t=1}^{j}(A(t) - M(L))\right] - \min\limits_{1 \leq j \leq L}\left[\sum\limits_{t=1}^{j}(A(t) - M(L))\right]}{\sqrt{(1/L)\sum\limits_{t=1}^{L}(A(t) - M(L))^2}} \tag{11}$$

with $M(L) = (1/L)\sum_{t=1}^{L} A(t)$ being the average value of time series $A(t)$. For self-similar processes, $R/S \sim (L/2)^H$ when L is large. H is the Hurst parameter of self-similar traffic. Assuming that AWP has run for time S, and a new Hurst parameter is estimated for every interval T, then the total number of H values becomes $n = \lfloor S/T \rfloor$, and $P(D)$ can be calculated using (12) as follows. It gives the average count of series $A(t)$ that exhibits self-similar behaviours per control period.

$$P(D) = \frac{1}{n}\sum_{i=1}^{n}\mu(H_i), \ \mu(H_i) = \begin{cases} 1, & \text{if } 0.5 < H_i < 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$P(D|M)$ is the probability that $B(t)$ inosculates the estimated available bandwidth. Define the error rate, Err_i, as

$$Err_i = \frac{|B(i) - A(i)|}{A(i)} \tag{13}$$

where $B(i)$ is the ith value of the predicted available bandwidth and $A(i)$ is the corresponding estimation value. Similar to (12), we have

$$P(D|M) = \frac{1}{n}\sum_{i=1}^{n}\tau(Err_i),$$
$$\tau(Err_i) = \begin{cases} 1, & \text{if } Err_i < 0.09 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Since the average error rate was found to be 0.0868 through experimental observation (see Section 4.2.3), the error rate threshold θ is set to 0.09.Thus the confidence parameter α, which is the posterior probability $P(M|D)$, can be obtained from (10). It denotes the degree to which the EMWM matches the real network environment based on observed data. The parameter α is used both in the congestion control module and the MTSP module to enhance the performance of AWP.

3.3 Congestion control scheme

3.3.1 Threshold for large time-scale rate control scheme: In the current design, the threshold θ in (4) is set to 0.5. If α falls in the closed interval [0.5, 1], it means that $B(t)$ has the accuracy of over 50% and AWP accepts the prediction information $B(t)$. Otherwise it implies that the prediction is not creditable. In order to efficiently avoid making fault control decisions under this situation, AWP makes no change of rate control on the large time-scale and waits for the next valid prediction.

3.3.2 Normal time-scale rate control scheme: To prevent network congestion it is necessary to reduce the sender transmission rate once the queuing delay or RTT increases. To do this the sender maintains an estimated long-term RTT and modifies its transmission rate according to how the most recent sample of the RTT differs from such an RTT value. The long-term RTT sample, $\overline{R}$, is set as follows [4]:

$$\overline{R} = \sqrt{RTT_0} \tag{15}$$

$$\overline{R} = q \cdot \overline{R} + (1-q)\sqrt{RTT_i}, \quad i = 1, 2, \ldots \tag{16}$$

Equation (15) takes effect when no feedback is received and (16) works as soon as the first feedback is received where RTT_i corresponds to the RTT value obtained from the ith feedback. Thus $\overline{R}$ gives the exponentially weighted moving average of the square root of the RTT samples. AWP is not sensitive to the value of constant q, and the recommended default value is 0.9 [16]. When $\sqrt{RTT_i}$ is greater than $\overline{R}$, it is deemed that the queue is increasing and the transmission rate should be reduced in order to maintain network stability.

On the normal time-scale, AWP employs one scheme out of the class of non-linear TCP-compatible congestion control called as binomial congestion control schemes, which are well suited for real-time multimedia applications [17, 18]. This scheme can modify the throughput smoothly in comparison to the sharp changes of the actual available bandwidth:

$$\lambda' = \begin{cases} \lambda + \delta & \text{if } E(\lambda) < \Lambda \\ \lambda - 0.6 \times \sqrt{\lambda / RTT_i} & \text{if } \sqrt{RTT_i} > \overline{R} \end{cases} \tag{17}$$

where λ' is the new throughput on the normal time scale, λ is the current throughput, δ is an additive linear increment, RTT is the round-trip time, and Λ is the expectation of the predicted series $B(t)$. With the rate control scheme revealed by (17), AWP increases its throughput on the normal time-scale when current throughput is less than the expected value on the large time-scale, and decreases its throughput on the normal time-scale when current throughput has caused increasing queuing length in the network environment. In such a way, the AWP throughput can follow the burstiness and idleness behaviour of self-similar traffic while keeping a relatively smooth output transmission rate, which is important for multimedia streams.

4 Performance evaluation of AWP scheme

To evaluate the performance of the AWP congestion control scheme, we have run extensive experiments built upon the network simulation (NS) platform [19] with multimedia applications and self-similar traffic conditions. We have compared the performance of the proposed AWP scheme to that of both the TCP and the well-known TCP-friendly rate control (TFRC) scheme. The results reveal that AWP is able to support higher available bandwidth utility and lower packet loss probability; meanwhile, it maintains smoothness and TCP-friendliness at a good level.

4.1 Experiment scenarios

We simulated a network environment where four concurrent connections are multiplexed over a shared bottleneck network link. Figure 4 shows the network topology. It comprises four client nodes, four corresponding sinks and a bottleneck link connecting gateways Router1 and Router2. The queue management scheme at Router1 of the bottleneck link is random early detection (RED), which alleviates the unbalanced problem between TCP-friendliness and compatibility by explicitly equalising packet loss probability across traffic flows [20]. The bottleneck link bandwidth is set to 10 Mbps with latency set to 40 ms. The rest link bandwidths are 100 Mbps with latency 10 ms.

In our experiments, the performance of TFRC and AWP was evaluated separately. Node 0 (in conjunction with sink Node 7) ran TFRC protocol and Node 1 (with sink Node 6) ran AWP protocol. Each simulation experiment was run for 200 s; it was observed that all simulations reached their stable state. At Node 2 (with sink Node 5), we set up 16 UDP agents that transfer files with Pareto distributed sizes [7] and act as infinite sources. Another 16 UDP agents with the same traffic were deployed to act as finite sources running from the 50th second to the 150th second, to generate traffic with higher burstiness. The Pareto distribution parameter of file sizes was set to 1.05, 1.25, 1.65 and 1.85 to generate different distribution shapes. There is an FTP source transferring packets with the size of 1400 bytes at Node 3 (with sink Node 4) using the TCP mechanism, with maximum window size of the bottleneck link

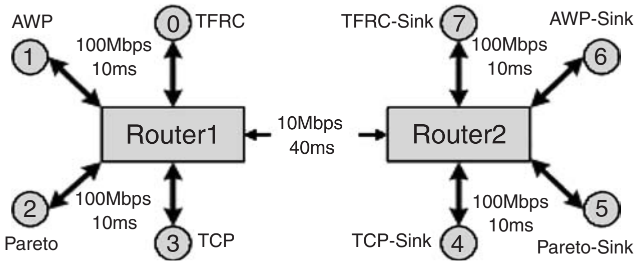

Fig. 4 *Network topology for NS simulations with bottleneck link*

bandwidth. For each simulation, the traces were collected at 0.3 s granularity. This potentially yields 667 data points for a single run of the simulation which can help us to obtain reliable performance results even under high variability associated with Pareto file distribution.

4.2 Experiment results

4.2.1 Average bandwidth utilities at various time-scales:

The first performance measure we present here is the average bandwidth utility over different time-scales. The range of time-scales considered here is from 0.5 s to 5 s, with a step interval of 0.5 s. Figure 5 shows that AWP has the highest bandwidth utility of 87.47% on average, whereas TFRC makes use of 81.93% of available bandwidth on average, and TCP reaches 76.69% only.

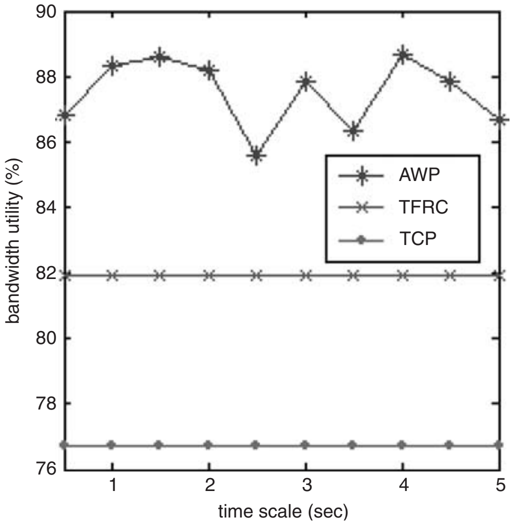

Fig. 5 *Average bandwidth utility at various scales of AWP, TFRC and TCP*

4.2.2 Dynamic available bandwidth utility:

We further examine the dynamic available bandwidth utility of AWP on a chosen time-scale of 3 s, in contrast to those of TFRC. The choice of 3 s is because this value can maintain the balance of a high available bandwidth utility and also a low packet loss probability.

In Fig. 6, ABW, AWP, and TFRC stand for the actual available bandwidth, and available bandwidth utility achieved by AWP and TFRC respectively. Figure 6 shows that the dynamic available bandwidth utility of AWP follows the change of the actual available bandwidth more rapidly and closely than TFRC, especially when the network condition changes sharply. This property ensures a swift and adaptive response of AWP to various network conditions.

4.2.3 Accuracy of prediction:

One of the most important features of AWP is its predictability. The congestion control is made according to prediction information. Figure 7 illustrates the prediction trace against the real traffic. In order to demonstrate the prediction accuracy numerically, the error rate is defined using (13). The calculation reveals that the mean error rate is 0.0868 and the standard variance is 0.0744. These results reveal that the prediction process can predict the forthcoming self-similar network traffic with high accuracy.

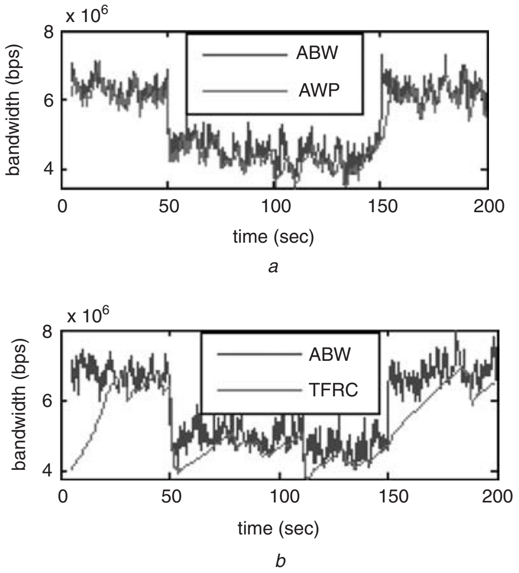

Fig. 6 *Dynamic available bandwidth utility of AWP and TFRC*

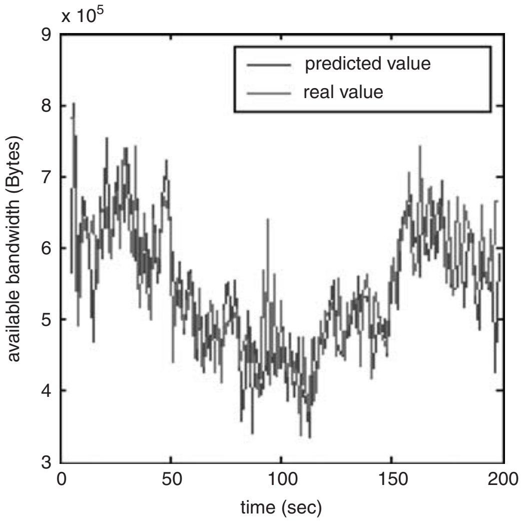

Fig. 7 *Comparison of predicted traffic and real traffic*

4.2.4 Packet loss probability:

The packet loss probability is calculated using (5). Figure 8 illustrates the packet loss probability of AWP with comparison to TFRC. On average, AWP has a slightly higher packet loss probability than TFRC, but the difference is very trivial. More specifically, AWP causes only 2 more packet loss events out of 10,000 than TFRC on average. It is worth noting that the impacts caused by a few more drops can be eliminated by an error control mechanism for multimedia transmission. Therefore, it can be concluded that AWP has similar packet loss performance to TFRC.

4.2.5 Standard variance of RTT:

Figure 9 depicts the standard variance of the round-trip time (RTT) of AWP, TFRC and TCP, respectively. A low level of variance accounts for relatively steady transmission RTT. We can find that the standard variance of the RTT achieved by AWP is almost half as much as that of TCP across all the time scales. Moreover, the variance value has a descent trend as the large time-scale increases. TFRC has the lowest variance of RTT, which is consistent with its smoother transfer property.

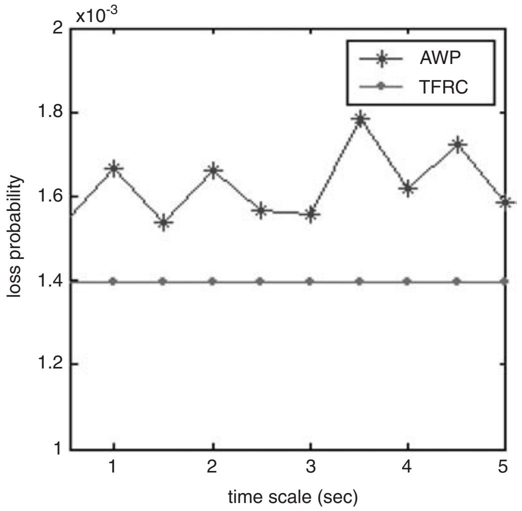

Fig. 8 *Packet loss probability of AWP and TFRC*

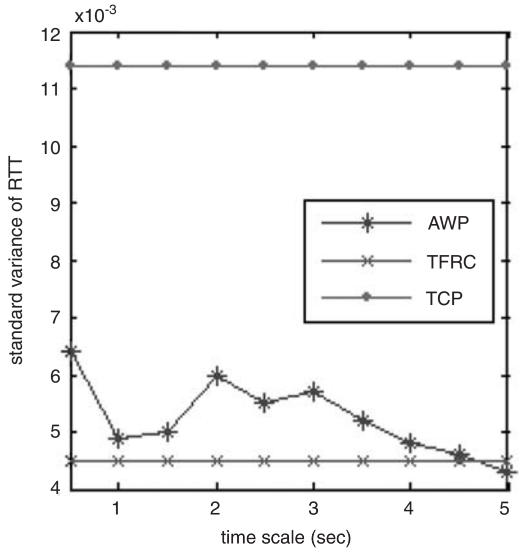

Fig. 9 *Standard variance of round-trip time (RTT)*

4.2.6 Transfer rate smoothness:

Let R^j_{tcp}, R^j_{awp}, R^j_{tfrc} denote the *j*th sampled transfer rate for TCP, AWP and TFRC, respectively, and let R_{awp}, S_{tfrc}, S_{tcp} represent the total number of samples. Define ΔAWP, $\Delta TFRC$, ΔTCP as follows:

$$\Delta AWP = \sum_{j=1}^{S_{awp}} |R^j_{awp} - R^{j-1}_{awp}| \quad \Delta TFRC = \sum_{j=1}^{S_{tfrc}} |R^j_{tfrc} - R^{j-1}_{tfrc}|$$

$$\Delta TCP = \sum_{j=1}^{S_{tcp}} |R^j_{tcp} - R^{j-1}_{tcp}| \tag{18}$$

to express the aggregated absolute value of rate difference [21].

Assuming that the smoothness degree of TCP is 1, we define relative smoothness degree of AWP and TFRC against TCP as

$$S_{awp} = \frac{\Delta AWP_i}{\Delta TCP_i} \quad S_{tfrc} = \frac{\Delta TFRC_i}{\Delta TCP_i} \tag{19}$$

Using (18) and (19), the degree of transfer rate smoothness can be calculated and shown in Fig. 10.

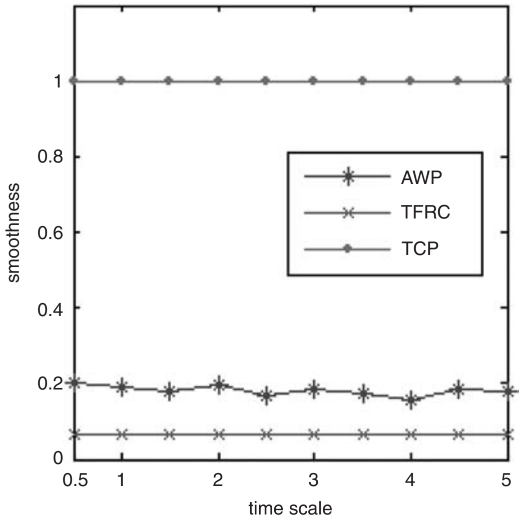

Fig. 10 *Comparison of the smoothness of AWP, TCP and TFRC*

Note that the lower the result of (19) is, the smoother the corresponding mechanism is. In Fig. 10, AWP and TFRC are both smoother than TCP. Note that AWP has shown to be less smooth than TFRC because AWP makes higher utility of dynamic available bandwidth and thus results in more changes of throughput. However, AWP is still much smoother than TCP and is close to the performance of TFRC.

4.2.7 TCP-friendliness of AWP:

Finally we investigate the TCP-friendliness of AWP. Let T_{awp}, T_{tfrc} and T_{tcp} denote the average throughput of AWP, TFRC and TCP, respectively. The friendliness is then defined as [21]:

$$F_{awp} = \frac{T_{awp}}{T_{tcp}} \quad F_{tfrc} = \frac{T_{tfrc}}{T_{tcp}} \tag{20}$$

With this definition, the friendliness of TCP is set to 1. The closer to 1, the more TCP-friendly the mechanism is. Figure 11 reveals that AWP is more friendly than TFRC on average, which means AWP has preserved TCP-friendliness during transmission. This advantage is very important when different network transmission protocols (e.g. TCP or

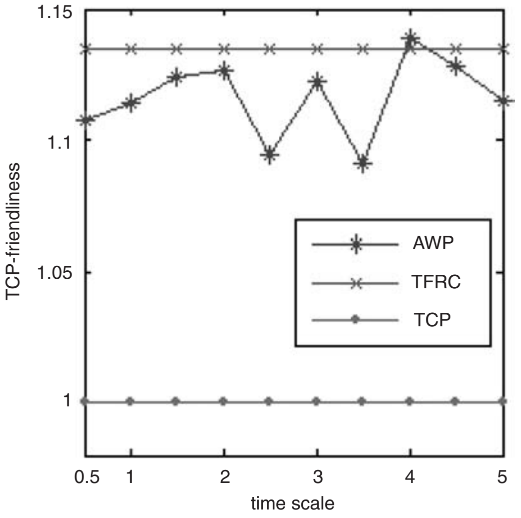

Fig. 11 *Comparison of the friendliness of AWP, TCP and TFRC*

TCP-compatible protocols) are running concurrently in one network environment.

5 Conclusions

Traffic self-similarity (i.e. scale-invariant burstiness and correlations) demonstrated by many high-quality measurement studies on the modern internet has a detrimental impact on network performance, including a great increase in both packet delay and loss rate. This paper has proposed a novel adaptive scheme for congestion control of multimedia self-similar traffic over multiple time-scales, which we refer to as adaptive wavelet and probability-based (AWP) scheme. On the normal time-scale, implicit prediction of available bandwidth obtained from feedback is used to support smooth and TCP-friendly rate-based feedback congestion control; on the large time-scale, explicit prediction to detect persistent changes in overall network contention is applied to modulate the rate control exhibited on the normal time-scale.

Unlike the existing congestion control schemes for self-similar traffic reported in the current literature, this proposed AWP scheme is more suitable for multimedia transmission as it uses a TCP-friendly and smooth rate control scheme. Moreover, it directly exploits traffic self-similar properties in congestion control in conjunction with the extended multifractal wavelet model (EMWM). In particular, AWP is adaptive to network environment changes by virtue of confidence analysis and auto-correction algorithm during the process of analysing network available bandwidth.

A series of simulation experiments have been carried out in order to compare the relative performance merits of AWP, TCP and TFRC. The results have shown the unique advantages for AWP including: (1) the network bandwidth utility is greatly improved along with the low packet loss probability since the self-similar properties of network traffic are exploited in the design of the congestion control scheme; (2) AWP contains good adaptation to the variation of network environment; (3) AWP preserves smoothness and TCP-friendliness of output rate, which are very important for the QoS improvement of multimedia application and network stability.

6 Acknowledgments

This work is supported by the National Science Foundation of China (No. 60372019, 90104002, 60173012 and 60473086), NSFC and RGC (No. 60218003), the Projects of Development Plan of the State High Technology Research (No. 2001AA112080).

7 References

1 Sastry, N.R., and Lam, S.S.: 'CYRF: a theory of window-based unicast congestion control', *IEEE/ACM Trans. Netw.*, 2005, **13**, (2), pp. 330–342

2 Cho, K.H., Park, S.J., Jung, E.H., Shin, S.W., and Lee, H.H.: 'End-to-end rate-based congestion control using EWMA for multicast services in IP networks', *IEE Proc., Commun.*, 2005, **152**, (5), pp. 668–672

3 Widmer, J., Denda, R., and Mauve, M.: 'A survey on TCP-friendly congestion control', *IEEE Netw. Mag.*, 2001, **15**, (3), pp. 28–37

4 Floyd, S., Handley, M., Padhye, J., and Widmer, J.: 'Equation-based congestion control for unicast applications'. Proc. ACM SIGCOMM, 2000, pp. 43–56

5 Crovella, M.E., and Bestavros, A.: 'Self-similarity in world wide web traffic: evidence and possible causes', *IEEE/ACM Trans. Netw.*, 1997, **5**, (6), pp. 835–846

6 Leland, W.E., Taqqu, M.S., Willinger, W., and Wilson, D.V.: 'On the self-similar nature of Ethernet traffic (extended version)', *IEEE/ACM Trans. Netw.*, 1994, **2**, (1), pp. 1–15

7 Paxson, V., and Floyd, S.: 'Wide area traffic: the failure of Poisson modeling', *IEEE/ACM Trans. Netw.*, 1995, **3**, (3), pp. 226–244

8 Tuan, T., and Park, K.: 'Multiple time scale congestion control for self-similar network traffic', *Perform. Eval.*, 1999, **36–37**, (1–4), pp. 359–386

9 Ribeiro, V., Coates, M., Riedi, R., and Sarvotham, S. *et al.*: 'Multifractal cross-traffic estimation'. Proc. ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, Monterey, CA, September 2000, pp. 15(1–10)

10 Weisstein, E.: 'Bayes' theorem, beta distribution'. From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/BayesTheorem.html, 2004

11 Baldi, P., and Brunak, S.: 'Bioinformatics – the machine learning approach' (MIT Press, 2001)

12 Stallings, W.: 'High-speed networks and internets: performance and quality of service', (Prentice Hall, 2002, 2nd edn.), pp. 181–207

13 Sahinoglu, Z., and Tekinay, S.: 'On multimedia networks: self-similar traffic and network performance', *IEEE Commun. Mag.*, 1999, **37**, (1), pp. 48–52

14 Riedi, R., Crouse, M., Ribeiro, V., and Baraniuk, R.: 'A multifractal wavelet model with application to network traffic', *IEEE Trans. Inf. Theory*, 1999, **45**, (3), pp. 992–1018

15 Abry, P., Flandrin, P., Taqqu, M.S., and Veitch, D.: 'Wavelets for the analysis, estimation, and synthesis of scaling data', in Park, K. and Willinger, W. (Eds.): 'Self similar network traffic analysis and performance evaluation' (Wiley, 2000), pp. 39–84

16 Handley, M., Floyd, S., Padhye, J., Widmer, J.: RFC 3448, Jan. 2003

17 Bansal, D., and Balakrishnan, H.: 'Binomial congestion control algorithms'. Proc. IEEE INFOCOM, April 2001, pp. 631–640

18 Chandrayana, K., Sikdar, B., and Kalyanaraman, S.: 'Comparative study of TCP compatible binomial congestion control schemes'. Proc. IEEE HPSR, Kobe, Japan, May 2002, pp. 319–323

19 McCanne, S., and Floyd, S.: 'The LBNL network simulator, ns-2', http://www.isi.edu/nsnam/ns/, 2004

20 Bansal, D., and Balakrishnan, H.: 'TCP-friendly congestion control for real-time streaming applications'. MIT Technical Report MIT-LCS-TR-806, 2000

21 Zhang, Q., Zhu, W.W., and Zhang, Y.Q.: 'Network-adaptive rate control and unequal loss protection with TCP-friendly protocol for scalable video over internet', *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, 2001, **5**, pp. 109–112

Copyright of IEE Proceedings -- Communications is the property of Institution of Engineering & Technology and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.