Packet-Level Traffic Allocation for Real-Time Streaming over Multipath Networks

Yanfeng Zhang, Cuirong Wang, and Yuan Gao

Northeastern University, Shenyang LN 110004, P.R. China
threewells14@gmail.com

Abstract. We address packet-level traffic allocation problem for real-time media streaming under multipath network environment. Based on an in-depth analysis of multipath real-time streaming model, also considering fluctuation of multipath network status as well as burst of media sending rate, we suggest that traffic load should be allocated to paths in proportion to the paths' available bandwidths, which minimizes the overall bandwidth overload probability. Moreover, due to the smallest transmission unit is packet, in order to execute the traffic allocation policy exactly, weighted size-aware packet distribution algorithm is proposed to avoid the actual traffic deviation due to variance of packet sizes. Simulation results show that the proposed algorithm outperforms other traditional algorithms, especially for reducing packet late arrivals, which has negative impaction in real-time transmission.

Keywords: traffic allocation, multipath, real-time streaming, available bandwidth, path redundance.

1 Introduction

In despite of the development of novel network infrastructures and constantly increasing bandwidth, Internet media streaming applications still suffer from limited and fluctuated bandwidth. Multipath streaming transmission has recently been proposed as a solution to overcome packet networks limitations [1], [2], [3]. It allows to increase the streaming bandwidth by balancing the load over multiple disjoint network paths between media sender and receiver. It also improves the error resilience of the media streaming system by means of redundant paths. Essential to such a multipath streaming system, at sender, is the packet distributor that dispatches media packets to the paths. It is necessary for the sender to distribute workload in a reasonable manner so that the multipath system can achieve its full potential.

How to distribute packets to achieve maximum benefit? Numerous studies [3], [4], [5], have made contributions on this research field. The fundamental concept is to allocate traffic in terms of available bandwidth. While all these works do not consider the fluctuation of network status enough. Unlike these approaches, which rely on UDP for streaming, some researchers focus on exploiting TCP for multipath real-time streaming, imposing TCP's state-awareness ability [6],

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 149–162, 2009.
© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2009

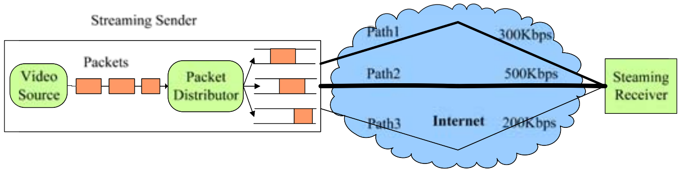

Fig. 1. Multipath streaming framework

[7]. For real-time specific, based on UDP, we try to implement a dynamic traffic allocation mechanism to "sense" the transmission characteristics of each path, and distribute packets fairly over the paths to achieve the designed goal.

In the framework of multipath network as shown in Fig. 1, our work addressed to the problem of streaming packet distribution, which takes into account real-time streaming characteristics. We are aiming at distributing packets fairly in order to achieve efficient utilization of bandwidth resources. Two key challenges are what is the distribution policy and how to execute this policy exactly. By means of analyzing media specific scenario, this paper gives corresponding solutions of these challenges.

In this paper, we make the following three contributions. (i) We analyze end-to-end multipath real-time streaming system in depth, and provide a model of bandwidth overload probability. (ii) Based on the model, we prove that allocating traffic in proportion to paths' available bandwidths respectively helps to reduce the overall overload probability. (iii) Following the traffic allocation policy, a weighted size-aware packet distribution algorithm for multipath real-time streaming is proposed, which is fine grained for its perceiving the smallest data unit (i.e. packet) over packet switching networks.

The rest of the paper is organized as follows. The multipath real-time streaming model is analyzed in Section II. Section III provides our optimal media-driven traffic allocation scheme and proves it. In section IV, we propose weighted size-aware packet distribution algorithm imposing upon traffic allocation policy. Simulation results are presented in Section V. Section VI concludes the paper.

2 Multipath Real-Time Streaming Analysis

2.1 Multipath Real-Time Streaming

We consider an end-to-end transmission framework where the media streaming application uses $M(M \geq 2)$ disjoint paths. Paths are considered to be disjoint if they do not share performance bottlenecks. The set of available loop-free paths between a media sender and a receiver is defined as $P = \{P_1, P_2, \ldots, P_M\}$.

For end-to-end perspective, we do look into the network status from an end-to-end point of view, rather than focus the hop-by-hop process during transmission. The network available bandwidth $b_i(t)$ (i.e., spare bandwidth), that is the bandwidth left unused by idle and non-greedy connections, is hence given by the following expression:

$$b_i(t) = C_i - \sum_{k \in K} \eta_i^k, \forall P_i \in P \tag{1}$$

where the first summation represents the total bandwidth of path P_i, while the latter summation of η_i^k represents the bandwidth allocated to other applications K, known as background traffic. Background traffic is always unsteady, and this instability lead to $b_i(t)$'s up and down.

Real-time video streaming is usually captured frame by frame by a video capture device every other fixed time, and the raw video frames are instantly encoded into compressed frames using some video encoder (e.g. H.264/AVC or MPEG-4). These compressed frames are commonly of different sizes in terms of video sequence characteristics and video encoder's configuration. Every encoded frame is then fragmented into network packets under the general rule stating that 1) each network packet contains data relative to at most one video frame, 2) several packets may contains data belong to the same frame. Let $\Pi = \{p_1, p_2, \ldots, p_N\}$ be the chronologically ordered sequence of N network packets, after fragmentation of the encoded frames. Any network packet p_n is characterized by its size s_n in bytes, frame number f_n, and its timestamps t_n. Timestamp is important for video player to play video packet at the right time. For the packets derived from the same frame, their frame numbers f_n, and timestamps t_n are uniform, which can be written as

$$f_n = f_{n+1} = \ldots = f_{n+k} \Longleftrightarrow t_n = t_{n+1} = \ldots = t_{n+k} \tag{2}$$

A packet distributor is set to permit data packets to be dispersed on multiple outgoing paths under a distribution scheme. Steaming application sends data at instantaneous rate of $R(t)$, which is split into many "fractional" rate $r_i(t)$, i.e. $R(t) = \sum_{i=1}^{M} r_i(t)$. $r_i(t)$ is the sending rate allocated to P_i at time instant t.

We denote by $\Phi = (\phi_1, \phi_2, \ldots, \phi_N)$ the distribution policy adopted by the streaming sender, and the ϕ_n represents the path chosen for packet p_n. In the multipath network scenario presented above, the sender can decide to send packet p_n through any path. Therefore, if p_n is distributed to path P_m, the packet p_n's imposed action $\phi_n = m$.

2.2 Packet Loss and Packet Late Arrival

In our streaming model, in order to decrease the video quality distortion, the streaming strategy aims at avoiding allocated bandwidth overload that results in packet losses and late arrivals. Firstly, we consider that the transmission links are lossless, and that packet loss only happens when sending a packet with the sending rate higher than the available bandwidth. Assuming a packet p_n allocated on P_i, i.e. $\phi_n = i$, we have

$$p_n \text{ is lost, if } r_i(t_n^s) > b_i(t_n^s) \tag{3}$$

where t_n^s is packet p_n's send time.

At the same time, even packet p_n is not lost, i.e. $r_i(t_n^s) < b_i(t_n^s)$, it still suffers from the danger of late arrival, which will be dropped too. Note that, time related metric such as packet late arrival and transmission delay is highly important for real-time real-time streaming, which distinguishes real-time streaming traffic from other traffic such as large file transmission.

Based on the previous work [8], we model the bottleneck link of each path as a work conserving queuing system with a service rate b_i, $i = 1, 2, \ldots$. We assume that the source flow is regulated by a σ, ρ leaky bucket (or a token bucket, which is implemented in most commercial routers). Let the real-time traffic's sending rate at t be $r(t)$, which is regulated by a σ, ρ leaky bucket, i.e., $r(t)$ conforms to a deterministic envelope process [9]. Due to this traffic shaping function, the source instantaneous rate on every path is shaped as:

$$r(t) = \rho + \sigma(t) \tag{4}$$

where ρ is the long-term average rate of the process (the rate factor), and $\sigma(t)$ is the burst during a small period of time, which is related to video sequence's characteristics.

Consider a work conserving queue with capacity $b(t)$, i.e. the available bandwidth. If the queue is stable, the queuing delay is upper bounded by the maximum busy period of the system [10]

$$d = \frac{\sigma}{\int_0^t b(u)du - \rho}$$

by means of (4), the the instantaneous fractional delay at t can be computed

$$d(t) = \frac{\sigma(t)}{b(t) - r(t) - \sigma(t)}$$

Given a decoding deadline's upper bound, and a packet p_n allocated on P_i, i.e. $\phi_n = i$, for $\sigma(t)$ is fixed in terms of video sequence, we have

$$p_n \text{ is late, if } b_i(t_n^s) - r_i(t_n^s) < \varepsilon \tag{5}$$

where ε is a positive bound to indicate late packets and t_n^s is p_n's send time.

2.3 Bandwidth Overload Probability

Packet loss or late arrival (i.e. unsuccessfully decoded packet) happens in terms of (3) and (5), which is due to traffic allocated overload. It is clear that lost packets is a subset of late packets, that is $b(t) - r(t) < \varepsilon$ limitation tighter than $r(t) > b(t)$ limitation. So we define the overload situation if $b(t) - r(t) < \varepsilon$ occurs.

Assuming at time η, the network available bandwidth is measured as $b(\eta)$, possibly with feedback of the receiver or other bandwidth detection approaches [11], [12]. However, network available bandwidth usually experiences change abruptly, given instantaneous detected bandwidth $b(\eta)$, during the period between two consecutive bandwidth detections, the actual available bandwidth is

$$b(t) = b(\eta) - X, t \in [\eta, \eta + \tau),$$

where X is the available bandwidth variance (i.e., traffic load variance) from $b(\eta)$, also known as background traffic burst, and τ is the bandwidth detection interval. Therefore, the probability of overload can be written as

$$Pr\Big\{[b(\eta) - X] - r(t) < \varepsilon\Big\} = Pr\Big\{X > b(\eta) - r(t) - \varepsilon\Big\}.$$

The burst length X (negative when light load) is commonly considered according to Pareto distribution [13]. Hence, according to Pareto property, we can carry on this consequence

$$Pr\Big\{X > b(\eta) - r(t) - \varepsilon\Big\} = \left[\frac{b(\eta) - r(t) - \varepsilon}{X_m}\right]^{(-\alpha)},$$

where the burst X converges to X_m in the limit of a large value of the exponent α, and α is a positive parameter (note that, the smaller α is, the greater probability overload occurs). In other words, X_m is the expected value of $b(\eta) - r(t) - \varepsilon$, i.e. $E\big[b(\eta) - r(t) - \varepsilon\big]$. For $b(\eta)$, its expected value keeps the same until the next available bandwidth detection, and ε is determined by the streaming application, while for $r(t)$, its expected rate can be computed as the mean rate during time scale $t \in [\eta, \eta + \tau)$, which is

$$E\big[r(t)\big] = \frac{\int_{\eta}^{\eta+\tau} r(t)dt}{\tau}$$

To sum up, given the instantaneous detected available bandwidth $b(\eta)$ at time η and packet late bound ε, during the period of $t \in [\eta, \eta + \tau)$, the overload probability is

$$Pr\Big\{b(t) - r(t) > \varepsilon\Big\} = \left\{\frac{b(\eta) - r(t) - \varepsilon}{b(\eta) - E\big[r(t)\big] - \varepsilon}\right\}^{(-\alpha)} \tag{6}$$

The analysis of multipath streaming as well as bandwidth overload probability, provides an in-depth study of multipath network behavior's character, and help us propose the optimal traffic allocation in the next section.

3 Traffic Allocation: Path Weight Determination

We generalize the previous observations, and derive theorems that guide the design of an optimal traffic allocation strategy. Since sending rate of every path decides the traffic load on that path, traffic allocation problem can be transformed to the problem of allocating rate among multiple paths. This section shows that, in the optimal traffic allocation, sending rate of every path is assigned in proportion to the path's available bandwidth, which minimize the overall bandwidth overload probability. We start from a multipath streaming scenario assuming available bandwidth of paths can be precisely detected periodically.

Theorem 1 (Rate allocation). *Given media application's instantaneous sending rate $R(t) = \sum_{i=1}^{M} r_i(t)$, and the detected available bandwidth $b_i(\eta)$ over P_i at time η, the optimal rate allocation $\boldsymbol{R(t)^*} = [r_1(t), \ldots, r_M(t)]^*$ during time interval $t \in [\eta, \eta + \tau)$, that minimizes the overall bandwidth overload probability based on (6):*

$$\boldsymbol{R(t)^*} = \Big[r_1(t), \ldots, r_M(t)\Big]^* = \arg\min_{\boldsymbol{R(t)}} \sum_{i=1}^{M} Pr\Big\{b_i(t) - r_i(t) > \varepsilon\Big\} \tag{7}$$

is set in proportion to paths' available bandwidths

$$\boldsymbol{R(t)^*} = \left[R(t) \cdot \frac{b_1(t)}{\sum_{i=1}^{M} b_i(t)}, \ldots, R(t) \cdot \frac{b_M(t)}{\sum_{i=1}^{M} b_i(t)}\right] \tag{8}$$

Proof. Deriving the minimum function given in (7),

$$\sum_{i=1}^{M} \left\{ \frac{b_i(\eta) - r_i(t) - \varepsilon}{b_i(\eta) - E\big[r_i(t)\big] - \varepsilon} \right\}^{(-\alpha)}$$

its minimum value is obtained when all the items are equal

$$\frac{b_1(\eta) - r_1(t) - \varepsilon}{b_1(\eta) - E\big[r_1(t)\big] - \varepsilon} = \ldots = \frac{b_M(\eta) - r_M(t) - \varepsilon}{b_M(\eta) - E\big[r_M(t)\big] - \varepsilon}$$

Only focusing on the first two paths, we have the cumulative equation during time period of $(\eta,\eta + \tau]$,

$$\int_{\eta}^{\eta+\tau} \Big\{b_1(\eta) \cdot E\big[r_2(t)\big] - b_2(\eta) \cdot r_1(t) + r_1(t) \cdot E\big[r_2(t)\big]\Big\}dt$$
$$= \int_{\eta}^{\eta+\tau} \Big\{b_2(\eta) \cdot E\big[r_1(t)\big] - b_1(\eta) \cdot r_2(t) + r_2(t) \cdot E\big[r_1(t)\big]\Big\}dt$$

Since

$$\int_{\eta}^{\eta+\tau} r_i(t)dt = \int_{\eta}^{\eta+\tau} E\big[r_i(t)\big]dt,$$

we finally obtain

$$\frac{\int_{\eta}^{\eta+\tau} r_1(t)dt}{\int_{\eta}^{\eta+\tau} r_2(t)dt} = \frac{b_1(\eta)}{b_2(\eta)}.$$

Considering instantaneous rate allocation, we let

$$\frac{r_1(t)}{b_1(\eta)} = \frac{r_2(t)}{b_2(\eta)}$$

to get the minimum value. In a similar way, we have

$$\frac{r_1(t)}{b_1(\eta)} = \frac{r_2(t)}{b_2(\eta)} = \ldots = \frac{r_M(t)}{b_M(\eta)},$$

where the path's rate is set in proportion to the available bandwidth.

Considering the constraint $R(t) = \sum_{i=1}^{M} r_i(t)$, to get the optimal rate allocation $\boldsymbol{R(t)^*} = [r_1(t), \ldots, r_M(t)]^*$, we should set path P_j's rate according to P_j's fraction of total available bandwidth, which minimizes the overall overload probability.

$$r_j(t)^* = R(t) \cdot \frac{b_j(t)}{\sum_{i=1}^{M} b_i(t)} \tag{9}$$

The traffic allocation method provides a reasonable way of distributing packets in order to reduce packet loss and late arrival probability. In our packet distribution scheme, path weight vector $(\omega_1, \omega_2, \ldots, \omega_M)$ is introduced, which indicates respective distribution capabilities of paths. By means of all paths' instant available bandwidth acquired by periodic detection, path P_m's weight can be determined

$$\omega_m = \frac{b_m(t)}{\sum_{i=1}^{M} b_i(t)}, \text{ and } \sum_{i=1}^{M} \omega_i = 1, \tag{10}$$

by which we execute packet distribution. A path with larger weight, is more likely to attract media traffic. Actually, $b_m(t) = 0$ is possible, which means no available resource can we consume on path P_m, then the path's weight $\omega_m = 0$ allows us to transmit no packet through path P_m, i.e. path P_m is abandoned. Extremely when only one path have available bandwidth, multipath transmission transforms to unipath transmission, which is reasonable in practical environment [17]. In the next section, we describe our complete packet distribution algorithm applying the path weight in detail.

4 Weighted Size-Aware Packet Distribution Algorithm

Suppose real-time streaming application generates a sequence of frames every other capture time interval, and they are encoded by some encoder (e.g. MPEG-4 or H.264/AVC). In practical, if an encoded frame's size is larger than network MTU, it is fragmented into several smaller network packets, each with size s_n in bytes. Then, in multipath streaming, these network packets $\Pi = p_1, p_2, \ldots, p_N$ are distributed to a set of M paths $P = P_1, P_2, \ldots, P_M$. Except this packet distributing thread, another work for available bandwidth detection thread is running. This detection and path weight computation are carried out every other interval τ. The path weight vector $(\omega_1, \omega_2, \ldots, \omega_M)$ is acquired in terms of (10). Actually, path weight indicates that path's expected traffic load proportion and it is updated periodically.

Focus back to the main sending thread, given periodic renewed path weight vector, packets should be exactly distributed according to path weight (i.e., the expected traffic load). Despite the rate allocation approach is an idealized scheme, but the smallest possible data unit in streaming is a packet, differentiated by size. Thus, a more explicit packet distribution scheme aware of packet size is proposed, whose philosophy is to minimize the deviation of actual traffic distribution from the given path weight vector, i.e., from the expected distribution.

Let $T_m(n)$ and $T_m^{'}(n)$, respectively, be the expected traffic load in bytes (determined by ω_m), and the actual traffic load in bytes to be sent on path P_m, just after the packet p_n's distributing decision has been made. For an idealized packet distributor, we have

$$T_m(n) = \omega_m \cdot \sum_{j=1}^{n} s_j$$

where s_j is the size of packet p_j and $j = 1, 2, \ldots, N$.

The main idea of packet distribution is to simulate optimal rate allocation as closely as possible. However, the assignment of a complete packet to a path may cause a transient load imbalance with respect to the targeted traffic allocation, that is some paths may be fed more traffic than expected temporarily while other paths may have less, after the distribution for a certain packet. Those paths fed with more traffic than expected have the tendency of not having the next packet assigned to them. Therefore, the current level of load imbalance as well as the size of the next successive packet is required for the traffic distributor to make the next distribution decision.

To quantify the above selection criterion, a metric is introduced to measure the traffic underload on a path. The residual traffic load of every path, just before distributing the packet p_n, $R_m(n)$, is defined as the amount of traffic load in bytes that should be fed on path P_m in order to achieve the expected traffic load. In other words,

$$R_m(n) = T_m(n) - T_m^{'}(n-1), \ \sum_{i=1}^{M} R_i(n) = s_n.$$

We use $R_m(n)$ to measure the streaming traffic underload on P_m, just before distributing p_n. If $R_m(n) > 0$, path P_m has been injected with less traffic than expected and, hence, p_n can be sent on this path. On the other hand, if $R_m(n) < 0$, there is too much streaming traffic being assigned on it and, hence, packet p_n should not be transmitted on this path. Briefly, $R_m(n)$ provides an indicator to the packet distributor for deciding which path p_n should be transmitted on.

Algorithm 1 presents the sketch of the main distributting process, where, for clarity, we bring up again $\phi_n = m$, if packet p_n is sent on path P_m. After running this algorithm, we can determine the optimal distribution policy Φ^*.

Concerning the distributing packet procedure's time and space complexities, it takes $O(N)$ time for processing each packet as it searches for a path P_m such that $R_m(n)$ is maximized. Also, it needs $O(N)$ counters to store its working variables. As the number of paths is generally small and fixed, we consider that the computational and storage costs are minimal. For the path weight computation, due to its simplicity and executed not very soon, its complexities are neglectable. At the same time, we argue that the packet distribution is fair and explicit. For any sequences of packets to be dispersed, the variance between the actual traffic load and the expected traffic load allocated to each path is always bounded by a finite constant.

Algorithm 1. Weighted Size-Aware Packet distribution

Require: $p_n, s_n, M, P_m, 1 \leq m \leq M$
Ensure: Optimal packet distribution $\Phi^* = [\phi_1, \phi_2, \ldots, \phi_n]^*$

Initialize the variables
while frame capture time comes **do**
  capture frame
  while bandwidth detection time comes **do**
    invoke Update_Pathweight()
  end while
  encode frame
  split frame into a packet sequence A_p with $n^{'}$ packets
  for all packets p_n in A_p **do**
    invoke Distribute_Packet(p_n)
  end for
end while

procedure DISTRIBUTE_PACKET(p_n)
  $S \leftarrow s_n$
  for all each $m, m \in 1, 2, \ldots, M$ **do**
    $R_m(n) \leftarrow R_m(n-1) + \omega_m \cdot S$
  end for
  choose a path $P_{m^{'}}$ such that $R_{m^{'}}(n)$ is maximized
  $\phi_n \leftarrow m^{'}$
  $R_{m^{'}}(n) \leftarrow R_{m^{'}}(n-1) - S$
end procedure

procedure UPDATE_PATHWEIGHT
  for all each $m, m \in 1, 2, \ldots, M$ **do**
    detect P_m's available bandwidth b_m
    update $\omega_m = b_m(t) / \sum_{i=1}^{M} b_i(t)$
  end for
end procedure

In summary, our packet distribution algorithm guarantees the variance between the actual traffic and the expected traffic under a limit bound. It is deployed at the media sender side, usually working for just one media flow, thus its complexity is acceptable for practical streaming applications.

5 Simulation Results

5.1 Simulation Setup and Relate Algorithms

We use ns-2 [16] to simulate multipath network scenarios. Two disjoint paths are selected between video sender (source) and video receiver (sink), with bandwidths of 1Mbps and 500Kbps respectively, and with the same end-to-end transmission delay of 100ms. A background traffic flow is generated according to the

On/Off Pareto distribution on the first path (namely path1) and on the second path (namely path2). The available bandwidth for our streaming application is considered to be the background traffic's rate subtracts from the total link bandwidth, which is detected every other 1 second.

Four packet distribution algorithms are studied, namely, weighted size-aware (WSA), weighted round robin (WRR), additive increase and multiplicative decrease (AIMD), and greedy (Greedy) [4], while the WSA approach is described in Section IV. WRR distributes packets to each path in a weighted cyclical fashion, where the weight is determined in terms of the total bandwidth of each path. AIMD focuses on a particular path, and utilizes this path in a probe manner. A initial threshold working as traffic load indicator is set at first, and media applications allocate traffic load lighter than the threshold. When the allocated traffic load does not exceed the available bandwidth, this threshold increases additively, otherwise, it decreased multiplicatively. The following packets after threshold hitting are distributed to the next path, where another instance of AIMD is running. Greedy method is based on [4], it will not chose another path for transmission unless all other available paths with higher available bandwidth have been chosen. Moreover, the chosen paths should be used at their maximum available bandwidth. Certainly, this available bandwidth is detected periodically by video streaming applications. Excluding WRR, all the other three algorithms are working by means of detected available bandwidth. Except the difference between traffic allocation schemes, an extraordinary of WSA from other schemes is its fine-grained property resulted from packet size awareness.

5.2 Comparison of Performance

We evaluate these algorithms introducing standard CIF sequences *foremancif* under different background traffic load levels, which are set as presented in Table 1. Fig. 2a compares the number of lost packets achieved by the four packet distribution schemes. Greedy as well as WSA performs better even under high background traffic load level. On the other hand, in order to test the late arrivals under different background traffic load levels, packet's maximum endurable transmission delay is set to 500ms, all the packets arrive later than this deadline are late arrivals. Fig. 2b gives the comparison of late arrivals over four algorithms. As expected,

Table 1. Background traffic load setup

Path	**Param**	**L1**	**L2**	**L3**	**L4**	**L5**
Path1	burst time (ms)	200	200	250	250	250
	idle time (ms)	50	50	30	30	30
	mean rate (Kbps)	750	800	850	900	950
Path2	burst time (ms)	100	100	200	200	200
	idle time (ms)	50	50	30	30	30
	mean rate (Kbps)	200	250	350	400	420

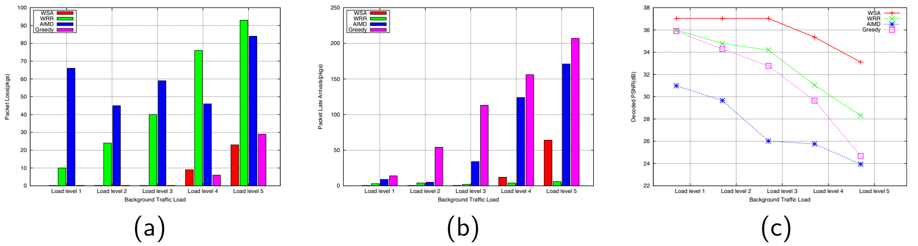

Fig. 2. Comparison of performance from four packet distribution algorithms under different load levels. (a) Lost packets. (b) Late packets. (c) PSNR.

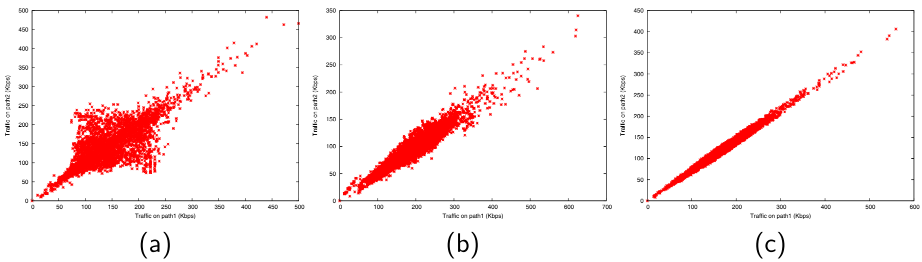

Fig. 3. Sample streaming rate on two paths with weigh ratio 3 to 2. (a) Round robin distributing packets. (b) Weighted round robin distributing packets. (c) Weighted size-aware (WSA) distributing packets.

Greedy generates a much larger number of late arrivals than other schemes, and AIMD also produces amount of late arrivals. Interestingly, WRR seems to have late arrivals avoidance, but we take notice that it has lost numerous packets, that have already deteriorated video quality.

As an approach of comprehensively considering packet loss and late arrival, we evaluate received video's quality measured by PSNR metric, as depicted in Fig. 2c. It demonstrates that, WSA always has the highest PSNR in all background traffic load levels. Another observation is that, Greedy's performance degrades faster than other schemes with the increasing of background traffic load.

5.3 Packet Size Aware

We now elucidate that WSA distributes streaming traffic load fairly by means of packet size awareness, which is subsequent upon path weight determination (i.e., rate allocation). Since encoded packets are of different sizes, even though every path's rate has been determined, distributing packets without considering packet size may lead to actual traffic load deviation from expected.

Fig. 3 plots a set of sample streaming rate vectors to demonstrate this by contrasting simple round robin and weighted round robin distribution with our weighted size aware distribution (i.e., WSA). *StarWarsIV* is used to generate

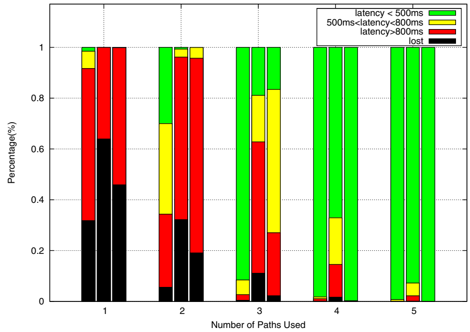

Fig. 4. Effect of path number

streaming traffic in this test. As shown is Fig. 3, each sample streaming rate vector consists of two sample rates, each corresponding to a path, and measured every other second. We observe that the distribution of sample rate vectors for streaming traffic with expected ratio of 3:2 between path1 and path2. It is clear that, when WSA is used, the sample streaming rate vectors are concentrated on a region of a shin diagonal stripe, where the slope of that stripe is equal to the expected ratio between path1 and path2. There is a much thicker stripe for using weighted round robin packet distribution, and even a worse sample rate vectors when simple round robin is employed. Anyway, WSA distributes packets to paths in a fine-grained manner according to expected traffic allocation.

5.4 Effect of Path Number

In the above simulation, we have focused on two paths transmission scenario to test weighted size-aware packet distribution algorithm. The effect of path number used for real-time streaming based on our WSA scheme is also evaluated. To survey this effect accurately, three long-term video trace files (i.e., *StarWarsIV*, *SouthPark*, and *OfficeCam*) is used as video source. Each path's bandwidth is set to 1Mbps, and background traffic of mean bitrate 800Kbps with On/Off exponential distribution is running on every path. All the three sequences are streamed using 1 to 5 paths.

The results are presented in Fig. 4, three columns of each path number represent the situations when introducing different source files. With the increase of path number, multipath's benefit is gained significantly. Interestingly, by increasing only one path improves the performance more than double times, and the effect of multipath streaming is quite tremendous, which implies our excellent packet distribution scheme. Additionally, it shows that, transmission latency is minimized greatly, which contributes to late arrivals avoidance. The simulation results prove that, by using weighted size-aware packet distribution for multipath real-time streaming, we make efficient utilization of network resources taking into account real-time streaming characteristics.

In summary, WSA packet distribution scheme performs better for multipath real-time streaming, it distributes packets through multiple paths to avoid bandwidth overload of a single path. The similar method aiming at balancing traffic load between different paths is WRR, which generates no late arrived packet as well as WSA. On the other hand, path with higher available bandwidth is preferred to other paths with lower available bandwidth in Greedy, and this strategy bears less packet losses than other strategies. However, packets distributor with Greedy algorithm brings a great number of late packets, which will be dropped by real-time streaming applications.

6 Conclusions

In this paper, we provide an in-depth analysis of multipath real-time streaming system considering media characteristics. These analyses point that by splitting traffic in proportion to the path's available bandwidth, streaming applications experience minimal bandwidth overload probability, which results in packet losses and packet late arrivals. And based on the distribution policy, a novel weighted size-aware packet distribution algorithm (i.e., WSA) for multipath real-time streaming is described, which ensures actual load distribution with a small deviation from expected. Our simulation results demonstrate the effectiveness of WSA in reducing overall packet loss rate and packet late arrivals as well as in improving video quality. Due to its satisfied effect and low complexity, the weighted size-aware packet distribution algorithm provides a very practical solution to efficient real-time streaming over multipath networks.

References

1. Golubchik, L., et al.: Multi-path continuous media streaming: What are the benefits? ACM J. Perform. Eval. 49, 429–449 (2002)
2. Apostolopoulos, J.G., Trott, M.D.: Path diversity for enhanced media streaming. IEEE Communication Magazine 42, 80–87 (2004); Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Frossard, P., de Martin, J.C., Civanlar, M.R.: Media Streaming With Network Diversity. Proc. IEEE. 96, 39–53 (2008)
4. Jurca, D., Frossard, P.: Media Flow Rate Allocation in Multipath Networks. IEEE Trans. on Multimedia 9, 1227–1240 (2007)
5. Bui, V., Zhu, W.: Improving multipath live streaming performance with Markov Decision Processes. In: Proc. ISCIT 2007, pp. 580–585. IEEE Press, New York (2007)
6. Wang, B., Wei, W., Guo, Z., Towsley, D.: Multipath Live Streaming via TCP: Scheme, Performance and Benefits. In: Proc. ACM SIGCOMM CoNEXT 2007. ACM, New York (2007)
7. Fang, C., Fu, X.: Probe-Aided MulTCP: an Aggregate congestion control mechanism. ACM SIGCOMM Computer Communication Review 38, 17–28 (2008)
8. Shiwen, M., Panwar, S.S., Hou, Y.T.: On minimizing end-to-end delay with optimal traffic partitioning. IEEE trans. on Vehicular Technology 55, 681–690 (2006)

9. Cruz, R.L.: A calculus for network delay. I. Network elements in isolation. IEEE Trans. on Information Theory 37, 114–131 (1991)
10. Chang, C.S.: Stability, queue length, and delay of deterministic and stochastic queueing networks. IEEE Trans. on Automatic Control 39, 913–931 (1994)
11. Qi, F., Wei, J., Wu, J.: Available bandwidth detection with improved transport control algorithm for heterogeneous networks. In: Proc. IEEE Workshops on Distributed Computing Systems, pp. 656–659. IEEE Press, New York (2005)
12. Li, Y., Munro, A., Kaleshi, D.: Multi-rate congestion control using packet-pair bandwidth detection with session and layer changing manager. In: Proc. IEEE IPCCC, pp. 485–490. IEEE Press, New York (2005)
13. Zukerman, M., Neame, T.D., Addie, R.G.: Internet traffic modeling and future technology implications. INFOCOM 2003 Review 1, 587–596 (2003)
14. Bauke, H.: Parameter estimation for power-law distributions by maximum likelihood methods. The European Physical Journal B 58, 167–173 (2007)
15. MPEG-4 Video Traces for Network Performance Evaluation, `http://www.tkn.tu-berlin.de/research/trace/trace.html`
16. Network Simulator, `http://www-mash.cs.berkeley.edu/ns/`
17. Andersen, D., Snoeren, A., Balakrishnan, H.: Best-path vs.multi-path overlay routing. In: Proc. ACM SIGCOMM Internet Measurement Conference, pp. 91–100. ACM Press, New York (2003)