This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

# Weighted Size-Aware Packet Distribution for Multipath Live Streaming

Yanfeng Zhang, Cuirong Wang, Yuan Gao
College of Information Science and Engineering
Northeastern University
Shenyang, P.R.China 110004
Email: threewells14@gmail.com and {wangcr, gao}@mail.neuq.edu.cn

***Abstract*—We focus on the problem of packet distribution for live media streaming over multipath networks. After proposing multipath live streaming model and an in-depth analysis of it, we suggest that traffic load should be allocated to paths in proportion to the paths' available bandwidths, considering complicated network status changing as well as burst of media sending rate, that can minimize the bandwidth overload probability. Moreover, weighted size-aware packet distribution algorithm is proposed to avoid the actual traffic deviation from expected due to variance of packet sizes. Simulation results show that the proposed algorithm outperforms other traditional algorithms, especially on reducing late packets, which has negative impact on real-time requirement.**

***Index Terms*—Packet distribution, multipath networks, live streaming, packet size aware.**

## I. Introduction

Multipath streaming has recently been proposed as a solution to overcome media streaming application's annoying problems (e.g., limited and fluctuated bandwidth) [1]. It allows to increase the streaming bandwidth by balancing the load over multiple disjoint network paths between live streaming server (i.e., sender) and client (i.e., receiver). It also improves the error resilience of the media streaming system by means of redundant paths [2], [3]. 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.

Numerous studies [1], [4] have made contributions on this research field. The fundamental concept is to allocate traffic in terms of available bandwidth. While most of 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 live streaming, imposing TCP's state-awareness ability [5], [6].

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 networks as shown in Fig. 1, our work addressed to the problem of streaming packet distribution, which takes into account live streaming characteristics. We are aiming at distributing packets fairly in order to achieve efficient utilization of bandwidth resources. Two key challenges are

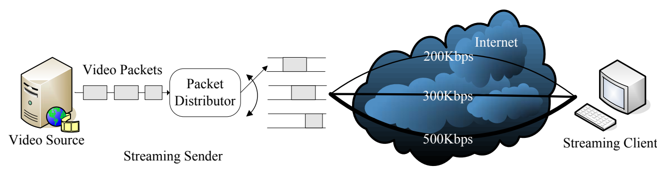



Fig. 1. Multipath streaming framework

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.

The rest of the paper is organized as follows. The multipath live streaming problems 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.

## II. Problems analysis

### A. Multipath Networks Model

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. Let $l_i$ and $d_i$ denote respectively the packet loss probability and the end-to-end transmission delay of path $P_i$. Due to continuous transformation of network status, we add time varying property to these notations, i.e., $l_i(t)$, $d_i(t)$ denoting the average corresponding states during the interval $(t - \Delta t, t]$. Certainly, $l_i(t)$ and $d_i(t)$ are relevant to the network available bandwidth $b_i(t)$ (i.e., spare bandwidth), that is the bandwidth left unused by idle and non-greedy connections. $b_i(t)$ is hence given by the following expression:

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

where the first summation represents the total bandwidth of path $P_i$, while the latter summation of $\eta_i^k$ 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.

### B. Multipath Live Streaming

Live 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 contain 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.

A packet distributor is set to permit data packets to be dispersed on multiple outgoing paths under a distribution scheme. Streaming 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 path $P_i$ at time instant $t$.

We denote by $\Phi = (\phi_1, \phi_2, \ldots, \phi_N)$ the distribution policy adopted by the streaming server, and the $\phi_n$ represents the path chosen for packet $p_n$. In the multipath network scenario presented above, the server 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 $\phi_n = m$.

### C. Lost and Late Packets

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

$$p_n \text{ is lost, if } r_i(t_n^s) > b_i(t_n^s) \tag{2}$$

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 live streaming, which distinguishes live streaming traffic from other traffic such as large file transmission.

Based on the previous work [7], 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 $\sigma, \rho$ 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 $\sigma, \rho$ leaky bucket, i.e., $r(t)$ conforms to a deterministic envelope process [8]. Due to this traffic shaping function, the source instantaneous rate on every path is shaped as:

$$r(t) = \rho + \sigma(t) \tag{3}$$

, where $\rho$ 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 [9]

$$d = \frac{\sigma}{\int_0^t b(u)du - \rho} \tag{4}$$

, by means of (3) the instantaneous fractional delay at $t$ is

$$d(t) = \frac{\sigma(t)}{b(t) - r(t) - \sigma(t)} \tag{5}$$

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

$$p_n \text{ is late, if } b_i(t_n^s) - r_i(t_n^s) < \varepsilon \tag{6}$$

, where $\varepsilon$ is a positive bound to indicate late packets and $t_n^s$ is packet $p_n$'s send time.

### D. Bandwidth Overload Probability

Packet loss or late arrival (i.e. unsuccessfully decoded packet) happens in terms of (2) and (6), which is due to traffic allocated overload. It is clear that lost packet ratio 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 $\eta$, the network available bandwidth is measured as $b(\eta)$, possibly with feedback of the receiver or other bandwidth detection approaches [10], [11]. 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

$$b(t) = b(\eta) - X, t \in [\eta, \eta + \tau) \tag{7}$$

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

$$Pr\Big\{[b(\eta) - X] - r(t) < \varepsilon\Big\} = Pr\Big\{X > b(\eta) - r(t) - \varepsilon\Big\} \tag{8}$$

. The burst length $X$ (negative when light load) is commonly considered according to Pareto distribution [12]. Hence, according to Pareto property, we can carry on this consequence

$$Pr\Big\{X > b(\eta) - r(t) - \varepsilon\Big\} = \left[\frac{b(\eta) - r(t) - \varepsilon}{X_m}\right]^{(-\alpha)} \tag{9}$$

, where the burst $X$ converges to $X_m$ in the limit of a large value of the exponent $\alpha$, and $\alpha$ is a positive parameter (note that, the smaller $\alpha$ 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 $\varepsilon$ 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

$$E\big[r(t)\big] = \frac{\int_\eta^{\eta+\tau} r(t)dt}{\tau} \tag{10}$$

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

$$Pr\Big\{b(t) - r(t) > \varepsilon\Big\} = \left\{\frac{b(\eta) - r(t) - \varepsilon}{b(\eta) - E\big[r(t)\big] - \varepsilon}\right\}^{(-\alpha)} \tag{11}$$

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.

## III. TRAFFIC ALLOCATION:PATH WEIGHT DETERMINATION

We now 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 assume the available bandwidth of paths can be precisely detected periodically.

*Theorem 1 (Traffic 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 $\eta$, the optimal rate allocation $\vec{R(t)}^* = [r_1(t), \ldots, r_M(t)]^*$ during time interval $t \in [\eta, \eta + \tau)$, that minimizes the overall bandwidth overload probability based on (11):

$$\begin{aligned}\vec{R(t)}^* &= \Big[r_1(t), \ldots, r_M(t)\Big]^* \\ &= \arg\min_{\vec{R(t)}} \sum_{i=1}^M Pr\Big\{b_i(t) - r_i(t) > \varepsilon\Big\}\end{aligned} \tag{12}$$

is set in proportion to paths' available bandwidths

$$\vec{R(t)}^* = \left[R(t) \cdot \frac{b_1(t)}{\sum_{i=1}^M b_i(t)}, \ldots, R(t) \cdot \frac{b_M(t)}{\sum_{i=1}^M b_i(t)}\right] \tag{13}$$

*Proof:* Deriving the minimum function given in (12),

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

its minimum value is obtained when all the items are equal

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

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

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

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 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 $\vec{R(t)}^* = [r_1(t), \ldots, r_M(t)]^*$, we should set path $P_j$'s rate according to $P_j$'s fraction of total available bandwidth

$$r_j(t)^* = R(t) \cdot \frac{b_j(t)}{\sum_{i=1}^M b_i(t)} \tag{14}$$

,which minimizes the overall overload probability. ■

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

$$\omega_m = \frac{b_m(t)}{\sum_{i=1}^M b_i(t)}, \text{ and } \sum_{i=1}^M \omega_i = 1 \tag{15}$$

, 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 $P_m$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

is abandoned. Extremely when only one path have available bandwidth, multipath transmission transforms to unipath transmission, which is reasonable in practical environment [13]. In the next section, we describe our complete packet distribution algorithm applying the path weight in detail.

## IV. Weighted Size-Aware Packet Distribution Algorithm

Suppose live 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 multipath streaming, these encoded 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. The path weight vector $(\omega_1, \omega_2, \ldots, \omega_M)$ is acquired in terms of (15). Actually, path weight indicates that path's expected traffic load proportion and it is updated periodically.

Focus back the packet distribution 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 is an idealized scheme, 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 $\omega_m$), and the actual traffic load in bytes to be sent on path $P_m$, just after $p_n$'s distributing decision. For an idealized packet distributor, we have

$$T_m(n) = \omega_m \cdot \sum_{j=1}^{n} s_j \tag{16}$$

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

The main idea 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 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,

$$R_m(n) = T_m(n) - T^{'}_m(n-1), \ \sum_{i=1}^{M} R_i(n) = s_n. \tag{17}$$

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 packet $p_n$ should be transmitted on.

Algorithm 1 presents the packet distribution process. We bring up again $\phi_n = m$, if packet $p_n$ is sent on path $P_m$. After running this algorithm for each incoming packet $p_n$, we can obtain the optimal distribution policy $\Phi^* = (\phi_1, \phi_2, \ldots, \phi_N)^*$.

**Algorithm 1** Weighted Size-Aware Packet distribution

**Require:** $p_n$
**Ensure:** Optimal packet distribution $\Phi^* = [\phi_1, \phi_2, \ldots, \phi_n]^*$

```
S ← s_n
for all each m, m ∈ 1, 2, ..., M do
    R_m(n) ← R_m(n − 1) + ω_m · S
end for
choose a path P_m' such that R_m'(n) is maximized
φ_n ← m'
R_m'(n) ← R_m'(n − 1) − S
```

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 neglectable for practical streaming applications.

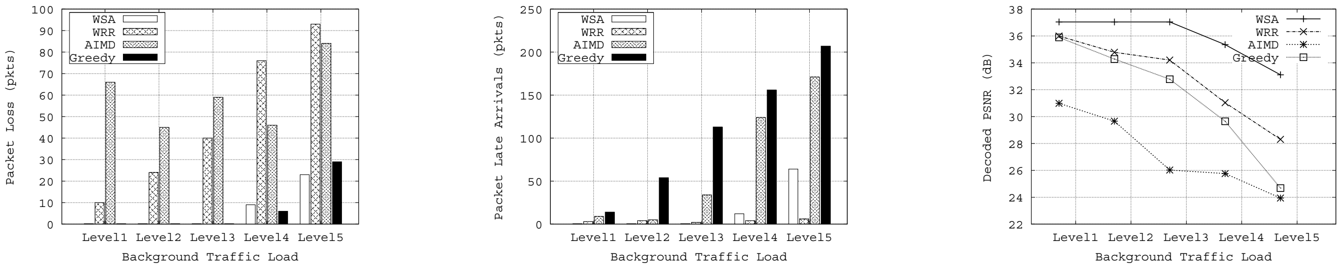
## V. Simulation Results

We use ns-2 to simulate multipath networks scenario. 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 second.

Four packet distribution algorithms are studied, our proposed weighted size-aware (WSA), weighted round robin (WRR), additive increase and multiplicative decrease (AIMD), and greedy (Greedy) [4]. 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. When the allocated traffic load on that path does not exceed the available bandwidth, this threshold increases additively, otherwise, it decreased multiplicatively. 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.

We evaluate these algorithms introducing standard CIF sequences *foremancif* under different background traffic load levels, which are set as described in Table I.

Fig. 2a compares the number of lost packets achieved by the four packet distribution schemes. Greedy as well as WSA




(a) (b) (c)

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

TABLE I
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 |

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, 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.

In summary, WSA packet distribution scheme performs better for multipath live 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 live streaming applications.

## VI. CONCLUSIONS

In this paper, we provide an in-depth analysis of multipath live 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 live 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.

## REFERENCES


[1] P. Frossard, J. C. de Martin, and M. R. Civanlar, "Media streaming with network diversity," in *Proc. IEEE*, no. 1, Jan. 2008, pp. 39–53.

[2] M. Zhang *et al.*, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," in *Proc. USENIX Annual Technical Conference*, Apr. 2004, pp. 99–112.

[3] J. Han, D. Watson, and F. Jahanian, "An experimental study of internet path diversity," *IEEE Trans. on Dependable and Secure Computing*, vol. 3, no. 4, pp. 273–288, Oct./Dec. 2006.

[4] D. Jurca and P. Frossard, "Media flow rate allocation in multipath networks," *IEEE Trans. on Multimedia*, vol. 9, no. 6, pp. 1227–1240, Oct. 2007.

[5] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via tcp: Scheme, performance and benefits," in *Proc. ACM SIGCOMM CoNEXT07*, Columbia University, New York, Dec. 2007.

[6] C. Fang and X. Fu, "Probe-aided multcp: an aggregate congestion control mechanism," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 1, pp. 17–28, 2008.

[7] S. Mao, S. Panwar, and Y. Hou, "On minimizing end-to-end delay with optimal traffic partitioning," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 2, pp. 681–690, March 2006.

[8] R. Cruz, "A calculus for network delay. i. network elements in isolation," *Information Theory, IEEE Transactions on*, vol. 37, no. 1, pp. 114–131, Jan 1991.

[9] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *Automatic Control, IEEE Transactions on*, vol. 39, no. 5, pp. 913–931, May 1994.

[10] F. Qi, J. Wei, and J. Wu, "Available bandwidth detection with improved transport control algorithm for heterogeneous networks," in *Proc. IEEE Workshops on Distributed Computing Systems*, Jun. 2005.

[11] Y. Li, A. Munro, and D. Kaleshi, "Multi-rate congestion control using packet-pair bandwidth detection with session and layer changing manager," in *Proc. IEEE IPCCC*, Apr. 2005.

[12] M. Zukerman, T. Neame, and R. Addie, "Internet traffic modeling and future technology implications," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 1, pp. 587–596 vol.1, March-3 April 2003.

[13] D. Andersen, A. Snoeren, and H. Balakrishnan, "Best-path vs.multi-path overlay routing," in *Proc. ACM SIGCOMM Internet Measurement Conference*, Miami Beach, FL, 2003.