# A Simulation Study of the Stochastic Compensation Effect for Packet Reordering in Multipath Data Streaming

Dmitry Korzun
*Department of Computer Science*
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
Email: dkorzun@cs.karelia.ru

Dmitriy Kuptsov, Andrei Gurtov
*Helsinki Institute for Information Technology*
Aalto University
Helsinki, Finland
dmitriy.kuptsov@gmail.com, gurtov@hiit.fi

***Abstract* — Multipath routing gains clear network performance advantages for data streaming. The level of packet reordering, however, becomes higher: distant packets are reordered, the application performance is reduced due to head-of-line blocking at the destination, and a large resequencing buffer is needed for sorting incoming packets. In this paper, we study the stochastic compensation effect to reduce packet reordering. If a source randomizes packet scheduling into multiple paths of random transmission delays, then these two sides of randomness "quench" each other. We perform simulation experiments to test this hypothesis in various multipath configurations and compare deterministic vs. randomized scheduling. The experiments show the existence of the stochastic compensation effect and its considerable influence on the application performance.**

***Keywords – data streaming; multipath routing; packet reordering; performance simulation; traffic modelling***

## I. Introduction

In multipath data streaming [1]–[4], a source schedules packets to split the stream among the different paths, proportionally to their rates. The rates are estimated based on such observable path characteristics as delay or bottleneck bandwidth. Rate-proportional scheduling is deterministic (e.g., round-robin) or randomized (e.g., Bernoulli scheme). Since network path characteristics are random there may be a large number of out-of-order packets at the destination. Packet reordering courses degradation of the application performance (user-perceived performance), especially when distant packets are reordered. The destination has to keep a large resequencing buffer for sorting incoming packets and then releasing them in the proper order to the application.

In stochastic systems, a randomized strategy instead of a deterministic one may lead to improvements. Variability of system state parameters are partially compensated by randomizing input parameters, which we call *the stochastic compensation effect*. For instance, it is exploited for TCP traffic by randomizing the packet sending times at TCP sources [5] and for distributed system maintenance by random neighbor replacement [6]. Nevertheless, the role of this effect in networking applications engineering is still opened for investigation. We expect that solutions with stochastic compensation become of higher utilization due to the growing parallelization in recent communication systems. In particular, the Internet of Things introduces extremely many parallel participants into communication, including many long-lived data flows [7].

In this paper, we study the problem of packet reordering in multipath data streaming. Our focus is on stochastic compensation. The performance degradation at the destination can be reduced by randomizing the packet scheduling at the source. Using simulation experiments, we measure packet reordering for various multipath configurations: the number of paths is varied, they have different probability distributions of transmission delay. We compare round-robin and Bernoulli scheduling and conclude that a randomized strategy can outperform a deterministic one.

Our observation indicates that the stochastic compensation effect is important for the data stream performance. Consequently, packet scheduling algorithms can benefit by incorporating some degree of randomization. Our simulation model assumes that all paths are independent, order-preserving, and their transmission delays are the key characteristic. Even in these simplified settings our simulation study indicates existence of the stochastic compensation effect. We expect that in more advanced settings (e.g., differentiating path bandwidth, introducing feedback from the destination) the effect can also take a place and be used in for improving the data stream performance.

The rest of the paper is organized as follows. Section II reviews related work. Section III states the packet reordering problem and our hypothesis on the stochastic compensation. Section IV describes our simulation model of packet reordering. Section V discusses the experiment results. Section VI summarizes the paper.

## II. Related Work

Packet reordering is an extensively exploited research problem in the Internet [1]–[4]. In particular, many packets within a TCP session can arrive out of the order at the destination [8]. Similar observation is for multipath traffic in wireless networks [9]. In load balancing of distributed systems [10], one of the well known techniques is splitting

978-1-5090-0206-1/15 $31.00 © 2015 IEEE
DOI 10.1109/EMS.2015.65

a single flow across multiple network paths selecting the lowest utilized path and implicitly mitigating the risk of packet reordering. For instance, [11] suggests using the earliest delivery path first routing policy given that the path characteristics, such as bandwidth and delay, are known precisely. Instead of operating on per-packet granularity, forwarding decisions can be made for large enough batches of subsequent packets scheduled to the same path [12]. The packet scheduling can be considered as an optimization problem [13] where the optimal forwarding policy takes into account the importance of each packet, in addition to the bandwidth and delay.

Optimal path rate computation was considered in [14]. In [9], [15] source and destination side buffers are applied to equalize the delays on all the paths and to minimize the overall packet reordering. Deterministic proportional scheduling is assumed in many proposals: the best path is selected according to the current network state and some optimization criterion [4], [11], [13]. Nevertheless, [16] pointed that sending packets through one of the two best paths according to a certain probability is better strategy than sending through the best available path.

None of the above works has addressed comparison of deterministic and randomized scheduling in multipath data streaming. There is still little evidence how effective the deterministic scheduling is in networks with high jitter.

## III. Packet Reordering Problem

Let us consider source $A$ operating in discrete uniform time $n = 1, 2, \ldots, N$ and serving an $N$-packet stream to destination $B$. Time instance $n$ is used as the unique packet sequence number; the assumption is common for streaming [1]–[3]. Intuitively, $A$ always has data to send and the forwarding cost is negligible for all paths.

There are $M > 1$ paths for forwarding packets (Fig. 1). Each packet $n$ is instantaneously dispatched to a path $i$ using a scheduler at $A$. Let $S_n^{(i)} > 0$ be the end-to-end delay of packet $n$ in path $i$. The destination $B$ reassembles the sequence of packets received via multiple network paths. The $n$th position in the output sequence is occupied by the packet of input sequence number $x_n$. Assuming no network loss, the output is a permutation of the input sequence.

We assume no bandwidth bottlenecks: an arbitrary number of packets can be sent sequentially to path $i$ without affecting the delay. The network has no loss: $S_n^{(i)} < \infty$. The sequence $\{S_n^{(i)}\}$ is i.i.d with generic element $S^{(i)}$ and the mean delay $\tau_i = \mathsf{E}[S^{(i)}]$, $0 < \tau_i < \infty$. Denote $\mu_i = 1/\tau_i$ the transmission rate. Then path $i$ has normalized rate

$$p_i = \frac{\mu_i}{\mu_i + \cdots + \mu_M}, \quad i = 1, \ldots, M. \tag{1}$$

As in many multipath models (see, e.g., [3], [9], [14], [17], [18]), $A$ does not know the full path states (e.g., the number of on-the-fly packets). Nevertheless, $A$ can estimate $\mu_i$. This case corresponds to parallel non-observable queues in queuing systems [19].

We assume that all $M$ paths are order-preserving. A path preserves the order if for any two packets $x_n$ and $x_m$ having followed the same path the inequality $x_n < x_m$ leads to $n < m$. It corresponds to the first-come first-serve (FCFS) service discipline and a single infinite-room queue. Other studies [1], [8], [20] confirm that reordering happens primarily due to the high path diversity.

Packet scheduler splits the input stream between $M$ paths proportionally to $p_i$. We distinguish two variants of proportional schedulers: deterministic and randomized. As particular representatives we consider weighted round-robin (WRR) scheduler and Bernoulli (BN) scheduler.

WRR scheduler operates in loop $i = 1, 2, \ldots, M$. Each iteration assigns a batch of subsequent packets to path $i$. The number of packets in a batch for path $i$ is fixed to $Cp_i$, where $C$ is a constant common for all paths. Lengthy packet batches, however, are constructed for fast paths when there are slow paths. It can occasionally amplify the packet reordering due to random path delays. For other deterministic schedulers let us mention the billiard scheme used in [21] to decrease the average waiting time.

BN scheduler forwards any packet $n$ to path $i$ with probability $p_i$. In contrast to WRR, forwarding decisions are independent. Packet batches of variable length appear.

Let $1 \leq n < m \leq N$ be packet positions in the output sequence. A reorder instance is a pair $(n, m)$ such that $x_n > x_m$. Set $r_{nm} = m - n$ if $(n, m)$ is a reorder instance and $r_{nm} = 0$ otherwise. If $r_{nm} > 0$ then packet $m$ has arrived late (equivalently, $n$ has arrived early). The value of $r_{nm}$ characterizes the reorder distance. Now we can define the reorder distance probability

$$d_k = \mathsf{P}[r_{n,n+k} > 0]. \tag{2}$$

In other words, an arbitrary output packet $n$ is reordered on distance $k$ with probability $d_k$. Note that $\sum_{k=1}^{N-1} d_k \neq 1$.

Reorder density is a probabilistic characterization of reordering [1], [22]. Let $r_n = n - x_n$ be displacement of packet $n$ from its original position. An early packet corresponds to a negative displacement and a late packet has a positive displacement. The displacement probability distribution is

$$f_k = \mathsf{P}[r_n = k], \quad -N < k < N, \tag{3}$$

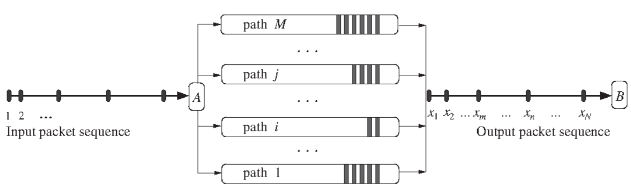


Figure 1. Positions $1 \leq n < m \leq N$ of the output sequence are occupied by packets $x_n$ and $x_m$.

i.e., $f_k$ shows the frequency of packets of displacement $k$.

For each given $n$, consider the last reordered packet $m$. Then the maximum reorder distance is on average

$$\rho_{\max}(N) = \frac{1}{N}\sum_{n=1}^{N} \max_{n<m} r_{nm}.$$

It estimates the resequencing buffer size needed at the destination. We expect that $\rho_{\max}$ and $d_k$ provide simpler model of the buffer demands than buffer-occupancy density [1], [22]. In particular, they can be used to characterize the user-perceived performance, since the larger buffer the application needs the higher the total delay.

Reorder entropy [23] expresses the total disorder of a packet sequence, including the fraction of displaced packets and the degree to which the packets are displaced,

$$\rho_{\mathrm{ent}}(N) = -\sum_{f_k>0} f_k \ln f_k.$$

It defines the randomness of the displacement and can be used as a summary metric of a displacement distribution tendency to be concentrated (low entropy) or dispersed (high entropy). When there is no reordering ($f_0 = 1$), then the minimum $\rho_{\mathrm{ent}} = 0$ is achieved. The maximum $\rho_{\mathrm{ent}} = \ln(2N+1)$ is when packets are displaced uniformly.

## IV. Simulation Model

We consider the following two path delay distributions. The first distribution is exponential, which is typical for modeling service time in queuing systems. The second distribution is power-law, which has the heavy-tail property—it appears in Internet traffic modeling.

*Exponential:* The probability density function (PDF) for $S^{(i)}$ is $f_i(x) = \lambda_i e^{-\lambda_i x}$ with the mean $\tau_i = 1/\lambda_i$ and the variance $\sigma_i^2 = 1/\lambda_i^2$.

*Power-law:* The PDF $f_i(x) = \frac{\alpha_i-1}{s_i^{\min}}\left(\frac{x}{s_i^{\min}}\right)^{-\alpha_i}$ has the minimal value $s_i^{\min}$. If $\alpha_i \le 2$ the mean is infinite, otherwise $\tau_i = \frac{\alpha_i-1}{\alpha_i-2}s_i^{\min}$. If $\alpha_i \le 3$ the variance is infinite, otherwise $\sigma_i^2 = \frac{\alpha_i-1}{\alpha_i-3}(s_i^{\min})^2$. The distribution is equivalent to Pareto distribution (by substitution $\alpha_{\mathrm{prt}} = \alpha - 1$) and is the continuous counterpart of the Zipf-like distribution. Further we shall use the Pareto distribution notation.

The ranges and variations of delay distribution parameters depend on concrete experiments (see Tables I and II). The mean-value ratio constraint $\tau_i/\tau_j < 10^2$ is preserved for any two paths $1 \le i, j \le M$, i.e., we do not consider degeneration with extreme path discrepancy.

Our experiments consist of two groups. In Group I, the number of paths is varied as $M = 2, 3, \ldots, 15$. The case corresponds to a network node with several appropriate neighbors for streaming. In Group II, the number of paths is fixed to $M = 2$. It is common for a multi-homed host.

TABLE I
DELAY PARAMETERS IN EXPERIMENTS WITH $M$ PATHS

| Pattern | Exponential | | Power-law | | | |
|---|---|---|---|---|---|---|
| | $\lambda \sim U[a,b]$ | $\tau_{\mathrm{avg}} = \sigma_{\mathrm{avg}}$ | $\alpha_{\mathrm{prt}} \sim U[a,b]$ | $s_{\min}$ | $\tau_{\mathrm{avg}}$ | $\sigma_{\mathrm{avg}}$ |
| Identical | $0.3$ | 3.3 | 2.3 | 10 | 17.7 | 21.3 |
| Similar | $0.3, 0.4$ | 2.9 | $2, 3$ | $10, 20$ | 25 | 22.4 |
| Slow | $0.1, 0.12$ | 9 | $2, 2.3$ | $10, 30$ | 37.4 | 65.8 |
| Fast | $0.3, 0.4$ | 2.9 | $3.3, 3.4$ | $1, 5$ | 4.3 | 2 |

TABLE II
DELAY PARAMETERS IN EXPERIMENTS WITH FIXED $M = 2$ PATHS

| | Exponential | | Power-law, $s^{\min} \sim U[2,3]$ | | |
|---|---|---|---|---|---|
| | $\lambda \sim U[a,b]$ | $\tau_{\mathrm{avg}} = \sigma_{\mathrm{avg}}$ | $\alpha_{\mathrm{prt}} \sim U[a,b]$ | $\tau_{\mathrm{avg}}$ | $\sigma_{\mathrm{avg}}$ |
| Fixed path | $1.00, 2.00$ | $0.67$ | $2.5, 2.7$ | $4.06$ | $3.25$ |
| Varied path | $0.01, 0.02$ | $66.67$ | $1.5, 1.8$ | $6.35$ | $\infty$ |
| | $0.03, 0.05$ | $25.00$ | $2.0, 2.3$ | $4.67$ | $8.23$ |
| | $0.10, 0.20$ | $6.67$ | $2.5, 2.8$ | $4.02$ | $3.06$ |
| | $0.30, 0.50$ | $2.50$ | $3.0, 3.3$ | $3.66$ | $1.92$ |
| | $1.00, 2.00$ | $0.67$ | $3.5, 3.8$ | $3.44$ | $1.41$ |
| | $2.50, 3.00$ | $0.36$ | $4.0, 4.3$ | $3.29$ | $1.10$ |
| | $3.00, 3.50$ | $0.31$ | $4.5, 4.8$ | $3.18$ | $0.91$ |
| | $3.50, 4.00$ | $0.27$ | $5.0, 5.3$ | $3.10$ | $0.77$ |

### *Group I: varied $M = 2, 3, \ldots, 15$*

The experiments are based on the three patterns: identical, similar, and slow & fast. Each pattern defines assignment of delay distribution parameters to $M$ paths. Table I shows the parameters, including mean $\tau_{\mathrm{avg}}$ and variance $\sigma_{\mathrm{avg}}$. When a pattern is selected, delay distributions are taken from the same class (exponential or power-low) for any path.

*Identical:* Every path has the same delay probability distribution parameters: $\forall i\ S^{(i)} = S$.

*Similar:* The delay probability distribution parameters are varied such that $\forall i\ S^{(i)} \approx S$. The small difference is due to selecting a distribution parameter ($\lambda$ or $\alpha$) randomly from a short interval $[a,b]$. Let us write $\lambda \sim U[a,b]$ or $\alpha \sim U[a,b]$.

*Slow & fast:* Given $M$ paths are classified into two types: slow and fast packet delivery. That is, if $i$ is slow and $j$ is fast then $\tau_i \gg \tau_j$. Simulation parameter $0 < q_{\mathrm{slow}} < 1$ defines the share of slow paths by assigning any path $i$ to be slow with the probability $q_{\mathrm{slow}}$.

### *Group II: fixed $M = 2$*

The experiments assign one path to have exponential delay and the other path to have power-laws delay, see the parameters in Table II. For one path its delay distribution parameters are fixed and for the other path its parameters are varied reducing the mean delay. Consequently, there are two types of the experiments in this group, depending for which delay distribution the variation is performed.

Our simulator is a custom application written in C++ (more than 1000 LOC). We favored the custom implementation over *ns*2 or *omnetpp* due to simplicity of the simulation

**Algorithm 1** Simulation scheme

Initialize $t_0 = 0$;
**for** $n = 1, 2, \ldots, N$ **do**
  Scheduler decides path $1 \le i \le M$ for packet $n$;
  Generate random delay $S_n^{(i)}$;
  Set $t_n = t_{n^*} + S_n^{(i)}$, where $n^*$ is immediately precedes $n$ in $i$;
**end for**
Construct $(x_n)_{n=1}^N$ by sorting $(t_n)_{n=1}^N$;

task. Each path is represented with a FCFS queue. Whenever the scheduler forwards every current packet $n = 1, 2, \ldots, N$ into a selected path $i$, the packet is assigned with a random delay drawn from the delay probability distribution.

Each sample $x$ of output packet sequence is produced using Algorithm 1. The auxiliary parameter $t_n$ is the arrival time of packet $n$ (the time elapsed from the start of data streaming). Note that if packet $n$ is sent down link $i$ then $t_n$ does not depend on how packets $n^* + 1, \ldots, n - 1$ are scheduled to other links. In particular, the simulation assumes that the cost of scheduling a packet is negligible compared with the end-to-end path delay.

For any experiment, 100 samples are constructed to estimate the probabilities $d_k$ and $f_k$ as well as the average and mean-square deviation for metrics $\rho_{\text{ent}}$ and $\rho_{\max}$.

## V. Analysis

In this section, we show that the deterministic behavior of a multipath packet scheduler can lead to higher reordering compared with randomized scheduling. The latter achieves the stochastic compensation effect when the randomization smoothens the path delay variability. To present our experiment results the average values are plotted with their mean-square deviation bars in Figs. 2–6 further.

### *A. Identical paths*

Since $p_i = p$ for any path $i$, WRR scheduler forwards exactly one packet to path $i$, the next packet is forwarded to path $i + 1$, and so on. There is no batch of subsequent packets on the same path. In contrast, BN scheduler occasionally assigns a batch from a packet $n$, in accordance with the geometric distribution. That is, $l$ successive packets

$$n, n + 1, \ldots, n + l - 1 \tag{4}$$

go to path $i$ with the probability $(1 - p_i)p_i^l$. If path $i$ is selected for forwarding (i.e., $l > 0$) then the conditional probability of a batch is $\mathsf{P}(l > 1) = p_i$.

Intuitively, packet batches stimulate reordering. If batch (4) goes to one path then packet $n + l$ goes to another path and likely arrives before the preceding packet $n+l-1$. The experiment result in Fig. 2 confirms the above reasoning on the WRR outperformance. This result agrees with the theoretical optimality of Round-Robin scheduling for identical parallel FCFS queues with infinite buffers [24].

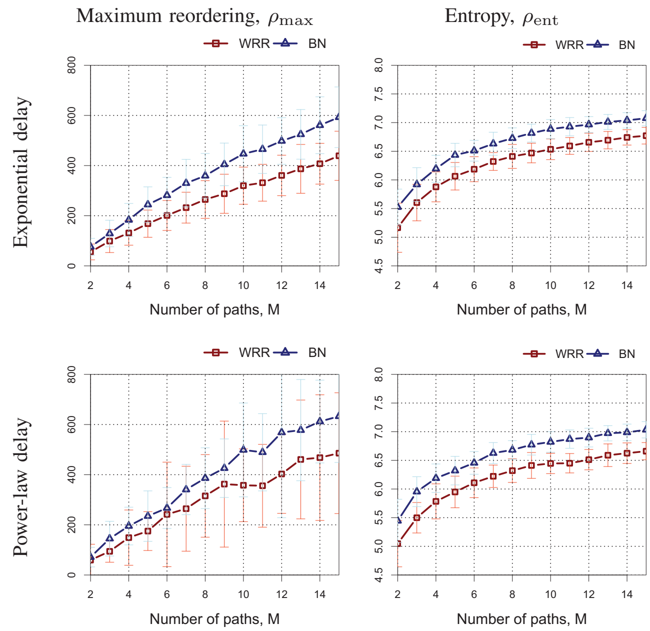


Figure 2. Identical paths in streaming $N = 10^4$ packets.

The required buffer size (estimated with $\rho_{\max}$) depends approximately linear on $M$ (Fig. 2, left column), supporting the intuition that reordering is proportional to the number of paths. Comparing the exponential and power-law distributions, the heavy-tail property of the latter leads to more variable maximum reordering.

The entropy (Fig. 2, right column) shows that the displacement randomness slowly grows with $M$ (the entropy maximum is about 9.9 for $N = 10^4$). The difference between WRR and BN is preserved approximately constant independently on $M$.

### *B. Similar paths*

Since $p_i \neq p_j$ for paths $i \neq j$, WRR scheduler assigns packet batches of fixed length for each path. Our experiments show that the BN randomization of batch length provides stochastic compensation (Fig. 3). It confirms that sometimes preferable to assign shorter or longer packet batches.

For similar paths BN scheduling leads to reordering almost the same as for identical paths. In contrast, the result of WRR scheduler is drastically changed. BN scheduling is advantageous, especially for heavy-tailed delays. The WRR performance variability is high even for exponential delays. The difference in entropy is decreasing for larger $M$.

### *C. Slow & fast paths*

We set $q_{\text{slow}} = 75\%$ and $q_{\text{slow}} = 25\%$. If $i$ is a slow path and $j$ is a fast path then $p_i \ll p_j$, see the delay parameters in Table I. In each experiment, both slow and fast paths have either exponential delay or power-law delay.

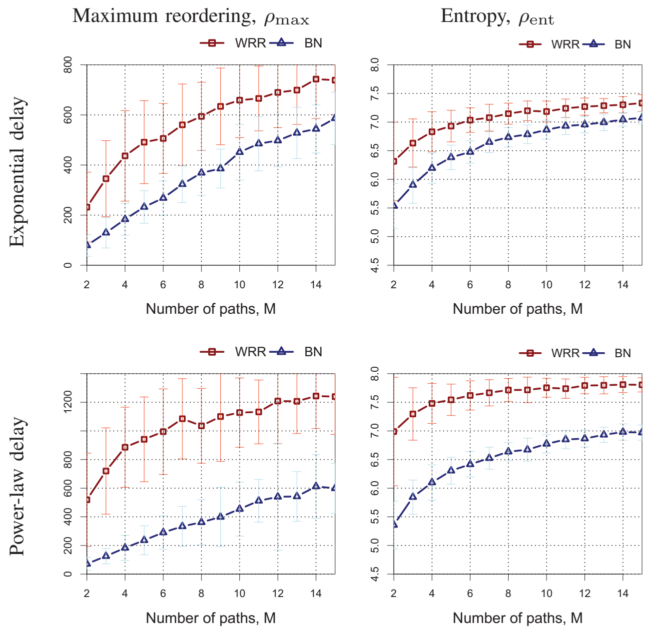


Figure 3. Similar paths for $N = 10^4$.

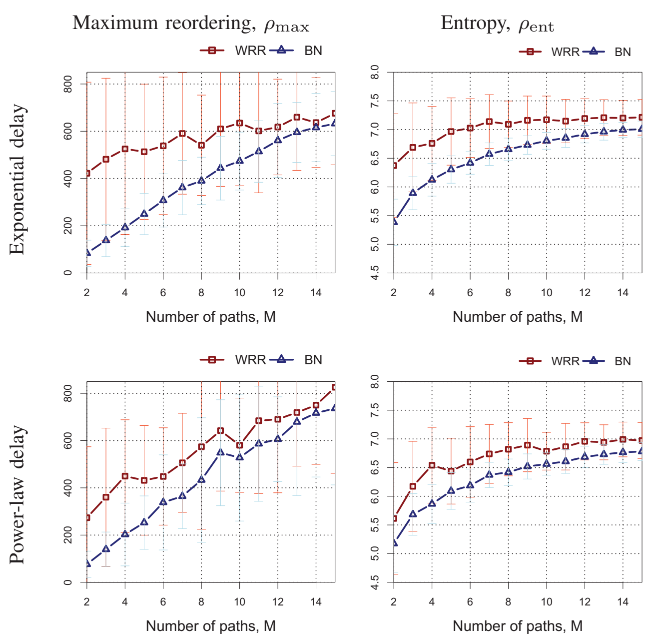


Figure 4. Slow & fast paths for $N = 10^4$ and $q_{\text{slow}} = 75\%$.

The result is shown in Fig. 4 for $q_{\text{slow}} = 75\%$ and in Fig. 5 for $q_{\text{slow}} = 25\%$. For all $M$, BN outperforms WRR and has lower variability. The BN outperformance degrades with increasing $M$. When $M$ is large the buffer sizes, estimated with $\rho_{\max}$, become close for WRR and BN.

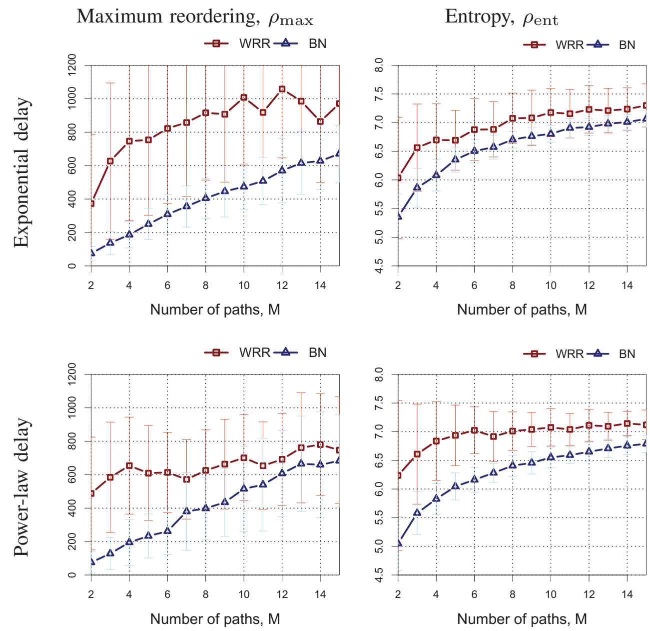


Figure 5. Slow & fast paths for $N = 10^4$ and $q_{\text{slow}} = 25\%$.

## D. Fixed two paths

Two paths are qualitatively different: one has exponential delay and the other is power-law. Experiment results in Fig. 6 show that BN outperforms WRR scheduler in this configuration when the delay of one path is varied.

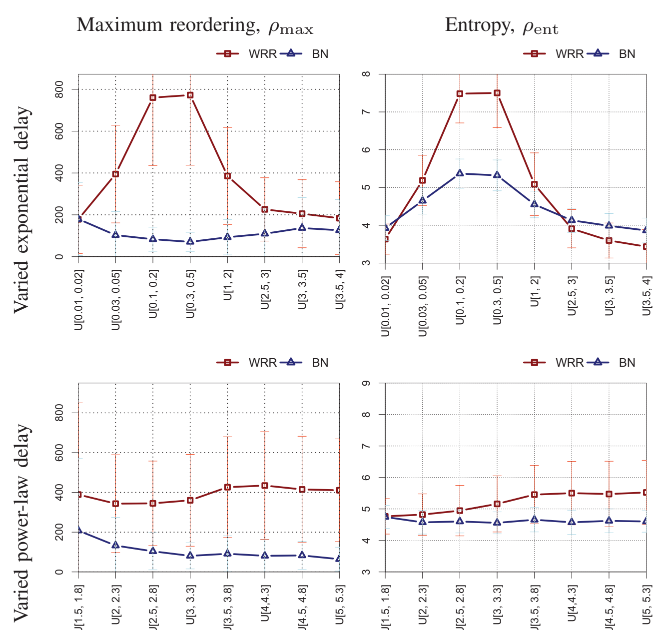


Figure 6. One path is fixed and the other is varied.

When the exponential delay is varied the bending point (for WRR) occurs as the ratio of the mean delays for both paths approaches 1, i.e., WRR scheduler starts to resemble pure round-robin. When the power-law delay is varied then BN scheduler shows clear advantages.

In this variation, we also observe similarity with the slow & fast experiments. A fast path often is being assigned with larger packet batches than a slow path.

## VI. Conclusion

In this paper we studied the packet reordering problem in multipath data streaming. Two strategies a source can apply to send data packets in proportion to the path transmission delay: deterministic and randomized. Our hypothesis is that randomness of transmission delay can be compensated with randomizing the packet scheduling on the source. We compared weighted round-robin scheduler (deterministic) and Bernoulli scheduler (randomized) using simulation experiments. Our results show that the randomization reduces packet reordering, and the stochastic compensation effect has a place for certain multipath configurations. We expect that utilization of this effect in networking applications can essentially improve the application performance when the path diversity of underlying networks is high.

## Acknowledgments

The reported study has been partially funded by RFBR according to the research project # 14-07-00252. The work of D. Korzun is financially supported by the Ministry of Education and Science of Russia within project # 1481 of the basic part of state research assignment for 2014–2016.

## References

[1] N. M. Piratla, A. P. Jayasumana, A. A. Bare, and T. Banka, "Reorder buffer-occupancy density and its application for measurement and evaluation of packet reordering," *Computer Communications*, vol. 30, pp. 1980–1993, June 2007.

[2] Y. Nebat and M. Sidi, "Parallel downloads for streaming applications: a resequencing analysis," *Perform. Eval.*, vol. 63, no. 1, pp. 15–35, Jan. 2006.

[3] J. Wang, J. Liao, and T. Li, "OSIA: Out-of-order scheduling for in-order arriving in concurrent multi-path transfer," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 633–643, Mar. 2012.

[4] J. Wu, B. Cheng, Y. Shang, J. Huang, and J. Chen, "A novel scheduling approach to concurrent multipath transmission of high definition video in overlay networks," *J. Netw. Comput. Appl.*, vol. 44, pp. 17–29, Sep. 2014.

[5] K. Chandrayana, S. Ramakrishnan, B. Sikdar, and S. Kalyanaraman, "On randomizing the sending times in TCP and other window based algorithms," *Computer Networks*, vol. 50, no. 3, pp. 422–447, Feb. 2006.

[6] P. B. Godfrey, S. Shenker, and I. Stoica, "Minimizing churn in distributed systems," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 147–158, 2006.

[7] D. Korzun, S. Balandin, and A. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the design challenges," in *Proc. 13th Int'l Conf. Next Generation Wired/Wireless Networking and 6th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2013)*, ser. LNCS 8121. Springer-Verlag, Aug. 2013, pp. 48–59.

[8] K.-C. Leung, V. O. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (TCP): Problems, solutions, and challenges," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 522–535, 2007.

[9] D. Kaspar, K. R. Evensen, A. F. Hansen, P. E. Engelstad, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and IP packet reordering over multiple wireless networks," in *Proc. IEEE Symp. on Computers and Communications (ISCC 2009)*. IEEE, 2009, pp. 637–642.

[10] E. Altman, U. Ayesta, and B. Prabhu, "Load balancing in processor sharing systems," *Telecommunication Systems*, vol. 47, no. 1-2, May 2011.

[11] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 388–403, Apr. 2006.

[12] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 51–62, March 2007.

[13] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Transactions on Multimedia*, vol. 9, pp. 629–641, Apr. 2007.

[14] M.-F. Tsai, N. Chilamkurti, J. Park, and C.-K. Shieh, "Multi-path transmission control scheme combining bandwidth aggregation and packet scheduling for real-time streaming in multi-path environment," *IET Communications*, vol. 4, no. 9, pp. 937–945, 2010.

[15] K. Evensen, D. Kaspar, A. F. Hansen, C. Griwodz, and P. Halvorsen, "Using multiple links to increase the performance of bandwidth-intensive UDP-based applications," in *Proc. 2011 IEEE Symp. on Computers and Communications (ISCC '11)*. IEEE Computer Society, 2011, pp. 1117–1122.

[16] M. A. Igartua, L. J. de la Cruz Llopis, V. C. Frías, and E. S. Gargallo, "A game-theoretic multipath routing for video-streaming services over mobile ad hoc networks," *Computer Networks*, vol. 55, no. 13, pp. 2985–3000, Sep. 2011.

[17] K.-C. Leung and V. O. K. Li, "A resequencing model for high-speed packet-switching networks," *Computer Communications*, vol. 33, no. 4, pp. 443–453, Mar. 2010.

[18] T. Zinner, P. Tran-Gia, K. Tutschku, and A. Nakao, "Performance evaluation of packet reordering on concurrent multipath transmissions for transport virtualisation," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 6, no. 3, pp. 322–340, Apr. 2011.

[19] J. Anselmi and B. Gaujal, "The price of forgetting in parallel and non-observable queues," *Perform. Eval.*, vol. 68, no. 12, pp. 1291–1311, Dec. 2011.

[20] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering, high speed networks and transport protocol performance," in *Proc. 13th Int'l Conf. on Computer Communications and Networks (ICCCN 2004)*. IEEE, 2004, pp. 73–78.

[21] A. Hordijk and D. van der Laan, "Periodic routing to parallel queues and billiard sequences," *Mathematical Methods of Operations Research*, vol. 59, pp. 173–192, 2004.

[22] N. M. Piratla and A. P. Jayasumana, "Metrics for packet reordering—a comparative analysis," *Int. J. Commun. Syst.*, vol. 21, no. 1, pp. 99–113, Jan. 2008.

[23] B. Ye, A. P. Jayasumana, and N. M. Piratla, "On monitoring of end-to-end packet reordering over the Internet," in *Proc. Int'l Conf. Networking and Services (ICNS 2006)*. IEEE Computer Society, 2006.

[24] K. P. Tsoukatos and A. M. Makowski, "Asymptotic optimality of the Round—Robin policy in multipath routing with resequencing," *Queueing Syst. Theory Appl.*, vol. 52, no. 3, pp. 199–214, Mar. 2006.