University of Würzburg
Institute of Computer Science
Research Report Series

Influence of Packet Re-ordering on Concurrent Multipath Transmissions for Transport Virtualization

Thomas Zinner[1], Kurt Tutschku[2], Akihiro Nakao[3],
Phuoc Tran-Gia[1]

Report No. 452 January 2009

[1] University of Würzburg
Institute of Computer Science
Department of Distributed Systems
Am Hubland, D-97074 Würzburg, Germany
{zinner,trangia}@informatik.uni-wuerzburg.de

[2] University of Wien
Institute of Computer Science
Department of Future Communication
Universitätsstrasse 10, 1090 Wien, Austria
kurt.tutschku@univie.ac.at

[3] University of Tokyo,
Graduate School of Interdisciplinary Information Studies
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
also with the NICT
1-33-16, Hakusan, Bunkyo-ku, Tokyo, 113-0001, Japan
nakao@iii.u-tokyo.ac.jp

Influence of Packet Re-ordering on Concurrent Multipath Transmissions for Transport Virtualization

Thomas Zinner, Phuoc Tran-Gia
University of Würzburg
Institute of Computer Science
Department of Distributed Systems
Am Hubland, D-97074 Würzburg, Germany
{zinner,trangia}@informatik.uni-wuerzburg.de

Kurt Tutschku
University of Wien
Institute of Computer Science
Department of Future Communication
Universitätsstrasse 10, 1090 Wien, Austria
kurt.tutschku@univie.ac.at

Akihiro Nakao
University of Tokyo, Graduate School of Interdisciplinary Information Studies
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
also with the NICT
1-33-16, Hakusan, Bunkyo-ku, Tokyo, 113-0001, Japan
nakao@iii.u-tokyo.ac.jp

Abstract

From the viewpoint of communication networks *Network Virtualization (NV)* extends beyond pure operational issues and addresses many of the impasses of the current Internet. The idea of *Transport Virtualization (TV)* progresses the capabilities of NV and enables the independence from a specific network transport resource. The independence is achieved by pooling multiple transport resources and selecting the best resources for exclusive or concurrent use. However, the application and selection of concurrent paths is rather complex and introduces inevitable packet re-ordering due to different stochastic delay characteristics on the used paths. Packets arriving at the destination out-of-order have to be stored in a resequencing buffer before reassembled packets are forwarded to the application. Therefore we perform an analysis of the fundamental behaviors and factors for packet re-ordering in concurrent multipath transmissions. This analysis applies analytical and simulative techniques which are used to compute the buffer occupancy probability distribution.

1 Introduction

The concept of *Network Virtualization (NV)* has recently emerged from operating system research [1]. NV enables the easy consolidation of multiple networks or overlays into a single physical system. Hence, an immediate benefit of NV is the reduction of the required amount of hardware (capital expenditures, CAPEX) and of the operational expenditures (OPEX) of network structures, since less physical systems have to be configured. As a result, NV is currently considered mainly as an operational technique.

From the viewpoint of architecting communication networks, the concept of NV has the potential to extend beyond the above outlined financial and operational issues. NV may address many of the impasses of today's Internet [2] and is therefore investigated as a key component of the Future Internet, e.g. [3, 4, 5].

The recently introduced idea of *Transport Virtualization (TV)* [6] progresses the capabilities of NV further. Like virtualization techniques in computer systems [7], which in general aim at the independence from a specific physical resource or its location, TV intends for the *independence from a specific network transport resource.* The independence is achieved by pooling multiple transport resources, e.g. transmission links or paths, and selecting the best resources for exclusive or concurrent use. The virtualization is achieved by simple and small network software developed for overlay network operation. Such a software selects one or more most appropriate transmission paths and coordinates the forwarding of data on them.

In this study we investigate the case when multiple paths are pooled and concurrently used. The application and selection of concurrent paths, however, is rather complex. First, concurrent multi-path transmission will inevitable introduce packet re-ordering due to different stochastic packet delay characteristics on the different paths. Second, the different stochastic delay processes can amend each other in their negative effects on the packet re-ordering. Third, the strength and occurrences of these combination effects are highly non-intuitive.

Thus, an analysis of the fundamental behaviors and factors for packet re-ordering in concurrent multi-path (CMP) transmission has to be carried out and will be presented in this paper. The analysis will apply analytical techniques as well as event-based simulations due to the complexity of the CMP mechanisms. Unfortunately, common simulation techniques might not be applicable for this investigation. Therefore, we will also discuss how to improve event-based simulation in order to achieve correct insights.

The paper is structured as follows. First, we will briefly repeat the idea of Transport Virtualization. Then, we detail how to implement TV using a *concurrent multipath (CMP)* transmission mechanism and discuss briefly on path selection in CMP transmission. After that, we introduce the CMP mechanism and its packet handling. Furthermore, we outline the analytical and simulative performance models used in this study. In particular, we discuss how to model and simulate packet delay realistically, i.e. packet re-ordering does not happen on a path. Finally, we provide a case study on the *re-sequencing buffer occupancy probability distribution* for different path delays. The investigation of this probability distribution gives insights in how to select the set of paths used in a TV mechanism using CMP transfer. The paper is concluded by a brief summary.

2 Transport Virtualization

One of the main benefit of virtualization techniques is the *abstraction of computer resources.* An operating system, for example, may provide virtual memory to an application program. This memory gives the program the feeling that it can use a contiguous working memory, while in fact it may be physically fragmented and may even overflow on to disk storage. The actual location of the data in the physically memory doesn't matter and is hidden. Thus, this virtualization technique makes the resource "memory" independent from its physical "location".

The idea of *Transport Virtualization (TV)* extends the concept of NV by transferring the feature of location independence to transport resources. TV is motivated partly by the abstraction introduced in P2P content distribution networks (CDNs). Advanced P2P CDNs, such as eDonkey or BitTorrent, apply the concept of *multi-source download (MSD)* where different peers are *pooled.* Upon a download, a peer receives multiple parts of a file in parallel from different peers. As a result, the downloading peer doesn't rely any more on a single peer which provides the data, and the reliability is increased. The providing peers are typically selected such that the

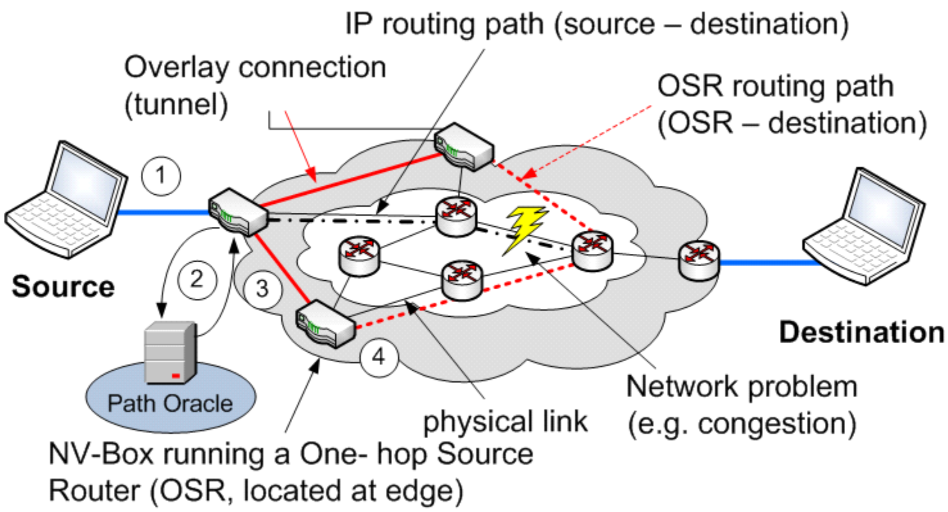

Figure 1: Simplified SORA Architecture

throughput is optimized. An appealing feature of the MSD concept is that the actual physical location of the file doesn't matter. Thus, these P2P CDNs can be viewed as an abstract and almost infinite storage for the data files. Thus, an abstraction of a storage resource, similar to the example of virtual memory, is achieved.

The above outlined concept of a virtual storage resource is now transferred to the area of data transport. *Transport Virtualization* can be viewed as an abstraction concept for data transport resources. Hereby, the physical location of the transport resource doesn't matter as long as this resource is accessible. In TV an abstract data transport resource can be combined from one or more physical or overlay data transport resources. Such a resource can be, e.g., a leased line, a wave length path, an overlay link, or an IP forwarding capability to a certain destination. These resources can be used preclusive or concurrently and can be located in even different physical networks or administrative domains. Thus, an abstract transport resource exhibits again the feature of location independence.

3 Implementing TV using Concurrent Multipath Transfer in Advanced Routing Overlays

A scalable approach for routing overlays is the concept of *one-hop source routing* [8] which is implemented by the *SORA architecture* [9]. Hereby, the user data is forwarded to a specific intermediate node, denoted as SORA router, which then relays the traffic to its destination using ordinary IP routing. The details of this architecture can be found in [9, 10] and [11].

The TV in the considered SORA architecture is achieved by a *Concurrent Multipath (CMP)* transfer mechanism. The mechanism combines multiple overlay paths (even from different overlays) into a single virtual high-capacity pipe. The combined paths are used in parallel by sending data packets concurrently on different overlay paths. This principle is also know as *stripping*.

The SORA architecture itself is depicted in Figure 1. The paths which form the virtual pipe are chosen by the ingress router out of large number of potential paths, cf. [10] and [11]. The *path oracle* discovers the available paths, i.e. the components of the abstract data transport resource. Current discussions suggest that a path oracle can be provided by the network operator or by other institutions [12]. Altogether, the CMP mechanism combined with the path oracle facilitates an abstraction of a data transport resource. Instead using a single fixed data transport resource, the system relies now on location independent, multiple and varying resources.

An important question for TV is the selection of pooled resources, i.e. the selection of potential paths. Typically, a *good* path has a short transmission delay, e.g. [10], and as a result, the mean path delay is an initial candidate as selection criteria. However, the selection of concurrent paths is rather complex. CMP transmission will inevitable introduce packet reordering due to different stochastic packet delay characteristics on the different paths, stochastic delay processes can amend each other in their negative effects (see Section 5.), and moreover, the strength and occurrences of these combination effects are highly non-intuitive. Therefore, a simple stochastic characterization of a path, such as the mean delay, is expected to be not sufficient. A path can exhibit a highly varying delay. Thus delay is composed of the propagation delay as well as queuing and processing delays at each router along the path. The path delay is therefore a complex stochastic process which can be measured and, upon the measurements, a delay distribution can be computed. We will not discuss how to obtain and how to measure such a delay distribution in this paper. Instead, we want to describe the influence of different delay distributions on TV when multiple paths are used.

4 Models

Next, we will outline the performance models for the suggested stripping mechanism for CMP transmission in TV. First, we will start with the stripping mechanism itself and then detail the analytical and simulative performance models. In particular, we discuss how to model and simulate packet delay realistically, i.e. packet re-ordering on a path does not happen.

4.1 Stripping Mechanism

Fig. 2 shows a detailed model of the stripping mechanism suggested for use in CMP transmission for TV. The data stream is divided into segments which are split into k smaller parts. The k parts are transmitted by the set of paths, i.e. in parallel on k different overlay links. The receiving router reassembles these parts. Unfortunately, the parts can arrive at the receiving router after different time intervals since they experience stochastically varying delay. Therefore, it is possible that parts arrive "out of order". It should be noted that part re-ordering can only happen between different paths. The order of parts on a path is maintained since packets typically can not overtake each other.

Part or packet re-ordering due to multi-path transmissions, may have a severe impact on the application performance. In order to level this behavior, the receiving router maintains a finite re-sequencing buffer. However, when the re-sequencing buffer is filled and the receiving router is still waiting for parts, part loss can still occur. This loss of parts is harmful for the application and should be minimized. Therefore, an important objective in operation of the system is to minimize the re-sequencing buffer occupancy. This can be achieved by a selection of paths with appropriate delay characteristics.

4.2 Analytical Model

In literature already exist initial analytical and approximative methods to estimate the re-sequencing buffer occupancy in case of multipath downloads or transmissions. A first analytical model can be found in [13]. The suggested performance model is already very powerful, however, we decided to conduct mainly simulation studies in order to address the full complexity of the mechanism. For example, it is easier to investigate different scheduling mechanisms within a simulative approach. Also, we don't have to care about memory boundaries during computation which are easily exceeded with an analytical approach based on combinatorial analysis.

4.3 Simulation Model

The behavior of the re-sequencing buffer occupancy is investigated by time discrete, event based simulation. The simulation model assumes a continuous data stream. The stream is divided into parts which are sent in parallel on either two or three paths. The delay on the paths is modeled by discrete delay distributions with a resolution of one time unit. A packet is transmitted every time unit on a path.

In order to achieve a realistic behavior of different path delays on the resequencing buffer, we have to ensure that no packet re-ordering on a single path can occur. That means that whenever a random delay for a path is generated, the previous delay has to be considered within the generation of the following packet. Hence, a current path delay has always to be bigger or equal than a previous path delay minus the interim time between the previous and the current packet. Furthermore, the relative frequency of all delays on a path has to converge against the given delay distribution for that path. We will discuss in the next subsection how we can generate packet delay distributions which fulfill these requirements.

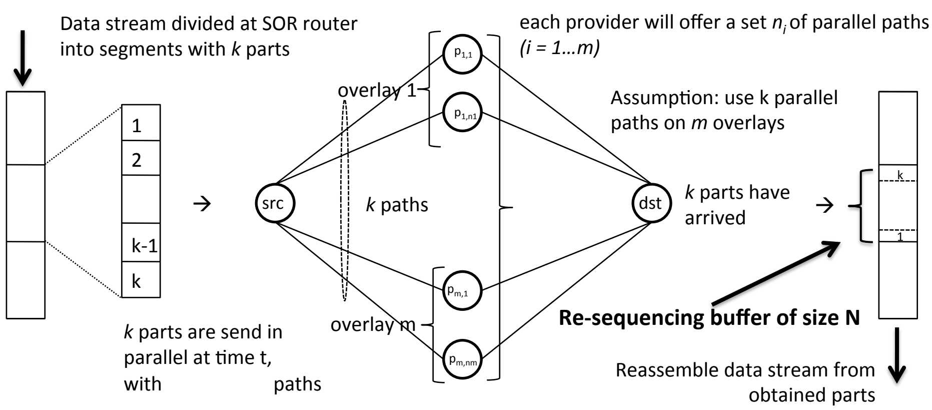

Figure 2: Transmission Mechanism

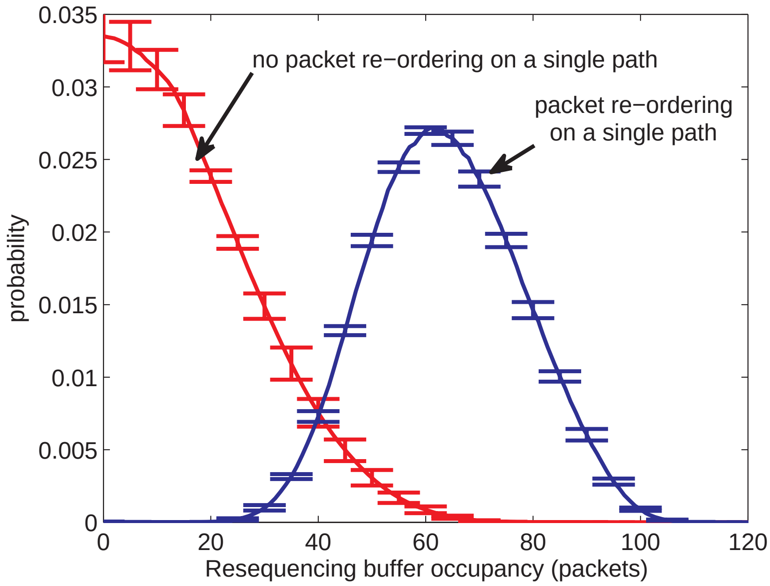

Figure 3: Buffer Occupancy with and without Packet Re-ordering within a Path

4.4 Modeling Packet Delay

Before we describe the suggested packet delay model, we outline the impact of an inappropriate delay model on the buffer occupancy.

In order to show this effect, we investigate the influence of a concurrent transmission over two paths with equal bandwidth. We assume that both paths to have the same delay distribution, which is set to a truncated gaussian-like distributions, each with mean value $\mu = 50$ time units and a standard deviation $\sigma = 20$ time units. The duration of the simulation run is 1 million time units, and we conducted 5 runs with different seeds. We investigate two scenarios: a) the delay distribution allows ”no packet re-ordering on a single path” and b) the delay distribution permits ”packet re-ordering on a single path”.

Fig. 3 shows the re-sequencing buffer occupancy distribution with $99,9\%-$confidence level for the two scenarios. One can see clearly a significant difference between the buffer occupancy distributions in case of dependent (”no packet re-ordering on a single path”) and independent packet delays (”packet re-ordering on a single path”). For an independent random delay (”packet re-ordering on a single path”), the re-sequencing buffer occupancy for the given input distributions follows a gaussian distribution. On the contrary, in case of a dependent random number generation (”no packet re-ordering on a single path”), it can be seen that the re-sequencing buffer occupancy is much smaller. We can conclude that with independant packet delays, the buffer occupancy will be significantly higher than in case of dependant packet delays. Hence, when aiming at accurate results, a simulative investigation has to use the dependent packet delay distribution, which considers no packet re-ordering on a path.

Now let us explain how we generate a dependent packet delay distribution. The applied path delay model is based on the delay model introduced in [13] and extended for the use in simulations. First, we describe the basic necessary conditions upon the delay distribution to ensure packets arrive at their destination in the same order they were transmitted. The complete derivation can be found in [13]. To facilitate the explanation, the following notations are used:

- d_i: the delay experienced by packet i
- Δ_i^t: the inter-departure time for packets $i-1$ and i

Consequently, there are only two restrictions for the experienced delay of packet i:

1. $d_i \geq 0$, this means that delay is always a positive value, and
2. $d_i \geq d_{i-1} - \Delta_i^t$, denoting the fact, that packet i can not overtake packet $i-1$. Nevertheless they may arrive at the destination simultaneously.

Therefore, d_i can be any probabilistic function $f(d_{i-1}, \Delta_i^t)$ that satisfies

$$f(d_{i-1}, \Delta_i^t) \geq \begin{cases} d_{i-1} - \Delta_i^t, & d_{i-1} - \Delta_i^t \geq 0 \\ 0 & d_{i-1} - \Delta_i^t < 0 \end{cases} \tag{1}$$

To ensure, the delay does not diverge, we need a stability constraint on $f(d_{i-1}, \Delta_i^t)$.

For the special case where packets are transmitted at a constant rate, i.e., every k time units we have a constant inter-departure time $\Delta_i^t = k \; \forall i$, and (1) becomes

$$f(d_{i-1}, \Delta_i^t) = f(d_{i-1}) \geq \begin{cases} d_{i-1} - k, & d_{i-1} \geq k \\ 0 & d_{i-1} < k \end{cases} .$$

The introduction of k is an extension to the model presented in [13] and enables us to adjust the path delay in a higher resolution.

For simplicity, let us assume that the delay is an integer value expressed as multiples of a time unit. Consequently, $f(d_{i-1})$ can be any integer that satisfies

$$d_i = f(d_{i-1}) \geq \begin{cases} d_{i-1} - k, & d_{i-1} > k \\ 0 & d_{i-1} \leq k \end{cases} \tag{2}$$

Any probabilistic function that corresponds to f in (2) can be represented by a Markov-chain, similar to Fig. 4, which is an example for $k = 1$, i.e. packets are sent every time unit. Here, state i corresponds to delay of i time units and the arrows correspond to the transitions among states with the respective probabilities. This transition probabilities can be written as a transition probability matrix P consisting of the elements $p_{i,j}$. For a finite maximum delay d_n, the transition probability matrix can be written as:

$$P = \begin{pmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,n-1} & p_{0,n} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,n-1} & p_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k,0} & p_{k,1} & p_{k,2} & \cdots & p_{k,n-1} & p_{k,n} \\ 0 & p_{k+1,1} & p_{k+1,2} & \cdots & p_{k+1,n-1} & p_{k+1,n} \\ 0 & 0 & p_{k+2,2} & \cdots & p_{k+2,n-1} & p_{k+2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & p_{n,n-1} & p_{n,n} \end{pmatrix} .$$

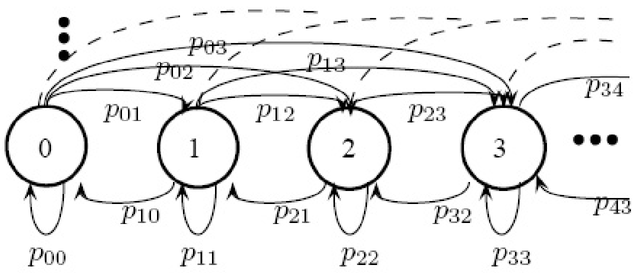

Figure 4: Markov-chain representing the delay, $k = 1$

The matrix P consists of four parts. On the main diagonal are the probabilities to stay in current delay state. On the right side of the main diagonal are the transition probabilities which increase the current delay state. The left side of the main diagonal illustrates a decrease in the delay state. The maximum decrease in the delay state depends on the inter departure time. Thus, not all states smaller than the current state can be reached. This is expressed by $p_{i,j} = 0\ \forall i < j - k - 1$.

For a given delay distribution d we have to solve the fix-point equations

$$d = d \cdot P\,, \tag{3}$$

so that the resulting transition probability matrix, and thus the represented Markov-chain is irreducible and aperiodic. That way we ensure that the delay process is recurrent. The transition probability matrix P can be used in our simulation model to assure that 1) the packet delay follows the given delay distribution d and that 2) no packet re-ordering occurs by a transmission on a single path. But before we can use P we still have to solve Equation 3, which will be described in the next subsection.

4.5 Computing the Transition Matrix by Using Linear Programming

In this subsection we describe our approach to determine a transition probability matrix P which fulfills the fix-point Equation 3. This equation is under-determined, i.e. we have to determine $\frac{1}{2} \cdot (n^2 + 2nk - k^2 + n - k)$ parameters with $n + 1$ equations, whereas $n + 1$ is the size of the delay state space d. For that, we model the problem as a Linear Program (LP) and solve this program with ILOG CPLEX [14]. In addition to the previously introduced variables, we define the following variables:

- c_1: vector with lower bounds for values on the main diagonal of matrix P
- c_2: vector with upper bounds for values on the main diagonal of matrix P

The LP depicted by Algorithm 1 aims to compute the transition probability matrix. This is expressed by Equation 4. The constraints on the variables are described in the following:

Algorithm 1 Determine the transition matrix

Maximize

$$f(P) = \sum_{i=0}^{n} \sum_{j=0}^{n} p_{i,j} \tag{4}$$

Subject to

$$\begin{aligned}
\sum_{i=0}^{n} p_{i,j} \cdot x_i &= x_j \;\; \forall j; && (5)\\
\sum_{j=0}^{n} p_{i,j} &= 1 \;\; \forall i && (6)\\
p_{i,j} &= 0, \;\; i < j-k-1; && (7)\\
0 < p_{i,j} &< 1, \;\; j-k < i < j && (8)\\
c_1 < p_{i,j} &< c_2, \;\; i = j && (9)\\
0 < p_{i,j} &< 1, \;\; i > j && (10)
\end{aligned}$$

- Equation 5 describes the $n+1$ fix point Equations derived from equation 3.
- The sum of each line in P has to satisfy the normalizing condition, i.e the sum over all probabilities is equal to 1, which is expressed by Equations 6.
- Equations 7 illustrate the transition probabilities in the lower left part of P. These probabilities have to be zero, in order to ensure that the next delay value is not smaller than the current delay minus the inter-departure time at the source.
- The probabilities between the main diagonal and the zero values are depicted by Inequalities 8. These probabilities denote a slow-going delay decrease without causing packet re-ordering.
- The main diagonal of the matrix, depicted by the Inequalities 9, denote that the delay remains constant. In order to avoid the trivial solution of the problem, the identity matrix I, these probabilities have to be smaller than 1, which is expressed by c_2.
- An increase of the delay between the departure of two packets is illustrated by the Inequalities 10.

5 Resequencing buffer occupancy behavior for different path delays - A case study

5.1 Outline of the Case Study

In this case study we want to investigate the resequencing buffer occupancy in case of different path delay distributions. The results presented next are of theoretical nature since they are obtained by the consideration of abstract models. However, they are intended to give a deeper insight intoich the practical question of how to select appropriate paths.

At first, we will examine the system behavior in case of similar path delay distributions with equal mean values and equal standard deviations. After that we study diverse path delay distributions, i.e. delay distributions with equal mean delay values but different standard deviations. In the last section we examine the impact of different path delay parameters for one distribution type and show results for transmissions via two and three concurrent paths.

5.2 Robustness of the Buffer Occupancy in Case of Similar Path Delay Distributions

As similar distribution types we choose a truncated gaussian (labeled *gaus*) and a truncated gamma (*gamma*) distribution. The probability mass functions are depicted in Fig. 5.

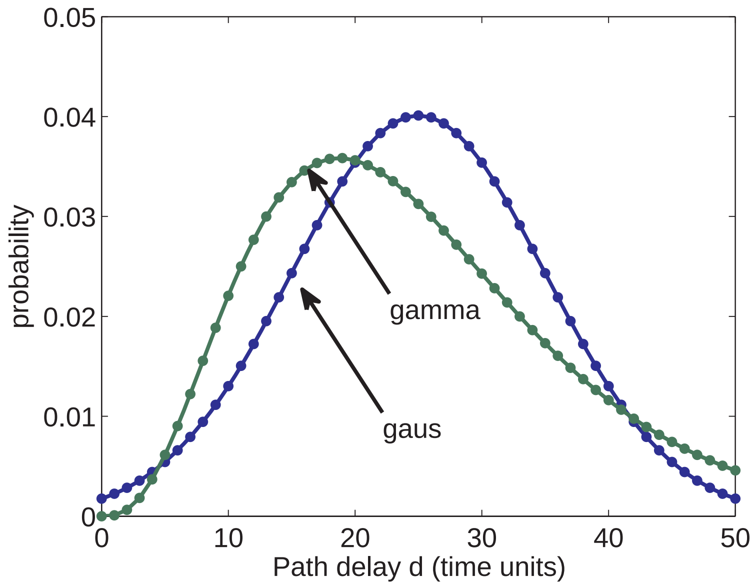

Figure 5: Gaussian and gamma distributions

The mean delay value and the standard deviation value for each of the depicted distributions are respectively $E[d] = \mu = 50$ and $\sigma = 20$. For these distributions we also investigate value pairs of $\mu = 50, \sigma = 10$ and $\mu = 25, \sigma = 10$. The delay values range between $d_{min} = 0$ and $d_{max} = 100$. We decided to investigate these distributions in order to evaluate the system behavior under quite similar conditions. We start with the investigation of a transmission over two equal paths and compare two similar path delay distribution types.

The results of these experiments are illustrated on a 95% confidence level as Probability Mass Functions (PMF)s in Fig 6. As parameters for the delay distributions we used $\mu = 50, \sigma = 20$ and $\mu = 25, \sigma = 10$. For both parameter sets, the difference between two equal gamma distributed path delays and two gaussian distributed path delays are negligible. The difference can be explained by the influence of upper statistical moments than the variance (skewness,

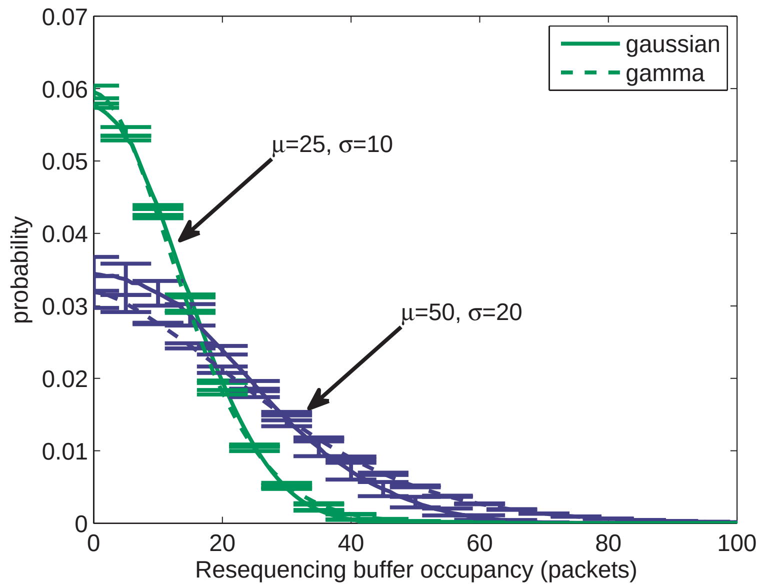

Figure 6: Resequencing buffer occupancy for similar path delay distributions

kurtosis,...). We can conclude, that these similar distribution types have, for equal average and variance, a similar influence on the buffer occupancy. Therefore, the buffer occupancy shows a good robustness, i.e. invariance, within this family of packet delay distributions. Hence, the distribution type with least computation requirements can be used.

5.3 Buffer Occupancy in Case of Diverse Path Delay distributions

Now we want to examine the behavior of the resequencing buffer in case of different types of delay distributions.

Therefore, three different path delay distributions are considered for the paths: a truncated gaussian (label *gaus*), a uniform (*uni*) and a bimodal distribution (*bi*). The PMFs of the distributions are depicted in Fig. 7. The mean delay value for each distribution is $\mu = 25$, whereas the delay ranges from $d_{min} = 0$ and $d_{max} = 50$. The coefficient of variation c_v varies between $c_v = 0.4$ for the gaussian distribution to $c_v = 0.8$ for the bimodal distribution. We decided to investigate these distributions in order to evaluate the system behavior under highly different condition, e.g. gaussian vs. bimodal delay.

We conduct an investigation of two concurrent paths. The buffer occupancies for different combinations of delay distributions are depicted in Figure 3. The y-axis denotes the probability of the packets stored in the re-sequencing buffer, assigned on the x-axis. For the sake of clarity we plotted only the bi,bi buffer occupancy distribution with confidence intervals for a confidence level of 99%.

For the case of two gaussian delay distributions, the buffer occupancy is left leaning and higher buffer occupancies are not very likely. However, for two bimodal delay distributions a

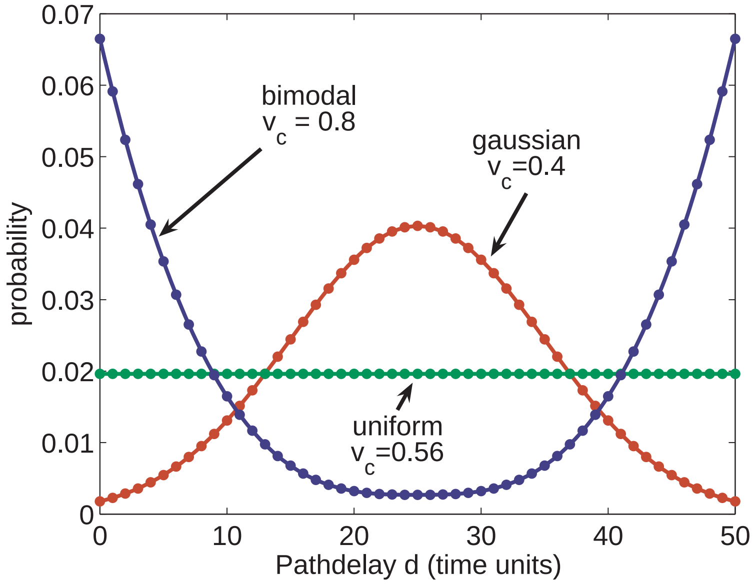

Figure 7: Nonsimilar distributrions

large fraction of the probability mass covers a buffer occupancy bigger than 30 packets. It should be noted that the maximum buffer occupancy in the investigated scenario $o_{max} = 50$.

5.4 Impact of Different Path Delay Parameters on the Buffer Occupancy

In this study we investigate the criteria for path selection. Intuitively a path selection algorithm should select those paths which provide the shortest delay. Typically this is of interest when the traffic comes from an interactive application with realtime constraints like video streaming or VOIP. For the investigation we consider truncated gaussian-like delay distributions with different mean packet delays $\mu = 50$ and $\mu = 25$ and different standard deviations $\sigma = 20$, $\sigma = 10$ and $\sigma = 5$. We start with a concurrent transmission over two paths.

The influence on the resequencing buffer is depicted on a $95\%-$confidence level in Fig. 9 as PMF. It can be seen that in case of a $\sigma = 10$ the buffer occupancy is almost independent from the mean value. For $\sigma = 20$ the distribution of the buffer occupancy gets lower and expands comparing to $\sigma = 10$. We can conclude that in case of a transmission over two paths with equal distributions the buffer occupancy depends mainly on the standard deviation.

In a next step we investigate the system behavior in case of a transmission over three paths. The results for five different scenarios are depicted as PMF on a $95\%-$confidence level in Fig. 10. The scenarios are a) all paths with same parameters $\mu = 50$, $\sigma = 10$, b) all paths with same parameters $\mu = 25$, $\sigma = 10$, c) two paths with parameters $\mu = 50$, $\sigma = 10$, one path with $\mu = 25$, $\sigma = 10$, d) the reverse of c), and e), similar to d), but two paths with parameters $\mu = 25$, $\sigma = 5$.

A close look at Fig. 10 reveals, that a) and b) have similar buffer distributions. This indicates that the pure delay has no impact on the buffer occupancy, which already has been shown for the

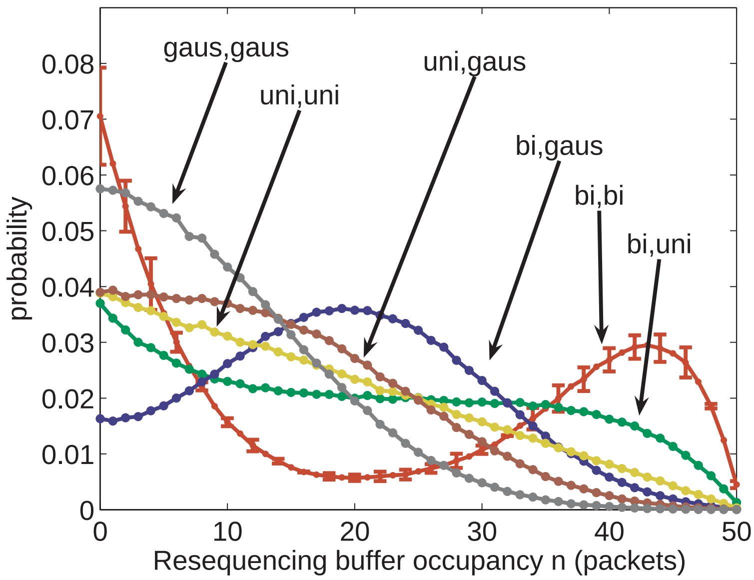

Figure 8: Re-sequencing buffer occupancy in case of nonsimilar delay distributrions

corresponding two path buffer occupancy. For the cases c) and d) it is remarkable that case d) with two high path delays performs in terms of buffer occupancy than case c). Intuitively more paths with lower delay should result in a better performance, but this is obviously not true as shown in case d).

Let us consider this behavior in greater detail. We assume therefore a single packet from the high delay path which is effectively much overdue. Until the arrival of this overdue packet, the low delay paths can easily increase the occupancy of the re-sequencing buffer. Thus, the buffer can be filled quickly by the low delay paths. This example shows that the high delay path becomes more dominant over low delays paths in terms of buffer occupancy. The selection of the paths should level the variation of the range of mean delays. Recent findings in Voice-over-IP systems, cf. [15] show that a constant delay has no significant impact on the Quality of Experience of the application, as long as the delay is below a certain threshold. In such a case the user does not notice whether the transmission is conducted via a path with a high or a low delay. Thus it might be better in TV to choose a path with a higher mean delay in order to relieve the resequencing buffer and avoid packet loss.

Finally, for the cases d) and e) we can see that in case e) the system performs worse in terms of buffer occupancy than in case d). Here the standard deviation on the smaller paths is lower than comparing to case d) and thus high and low delays are unlikely.

6 Related Work

While stripping mechanisms for IP networks are already suggested, c.f. [16, 17], limited research has been carried out so far in the context of path selection for CMP transmission mechanisms. The only work to our knowledge so far, which deals with this topic is the DaVinci

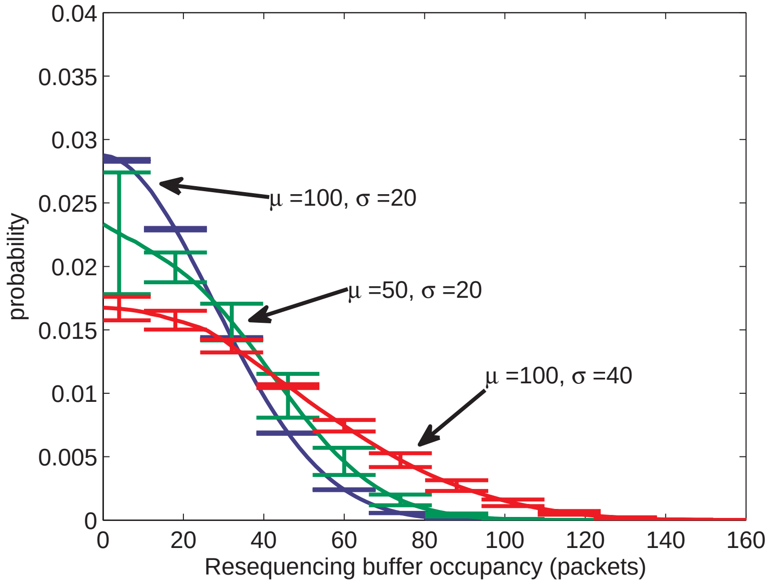

Figure 9: Different gaussian distributions

architecture [4]. In the DaVinci proposal, the paths are selected such that the creation of bottlenecks is avoided. The investigation in this paper goes beyond the results presented in the DaVinci architecture. Our paper discusses the fundamental performance issues in selecting the paths according to their statistical characteristics.

7 Conclusion

In this paper we have explained the idea of *transport virtualization (TV)*, which provides a location independent abstraction for data transport resources and outlined the TV concept by the example of *concurrent multipath (CMP)* transmission in one-hop source routing overlays. Intuitively, packet reordering will inevitable occur in case of CMP transmissions, due to the different stochastic delay properties on the involved paths. We discussed an important performance issue of CMP transmission, the *re-sequencing buffer occupancy probability distribution* under the influence of the delay distribution on the used paths in the CMP mechanism. Due to the complexity of this mechanism, we combined analytical and simulative techniques in order to investigate the re-sequencing buffer occupancy.

It turned out that different stochastic delay processes can amend each other in their negative effects on the packet reordering, leading to a higher re-sequencing buffer occupancy. Also, the strength and occurrences of these combination effects are highly non-intuitive.

With the presented approach continuing studies can be performed with the objective to find the best pool out of different provided paths in case of costs, re-sequencing buffer occupancy and QoS/QoE.

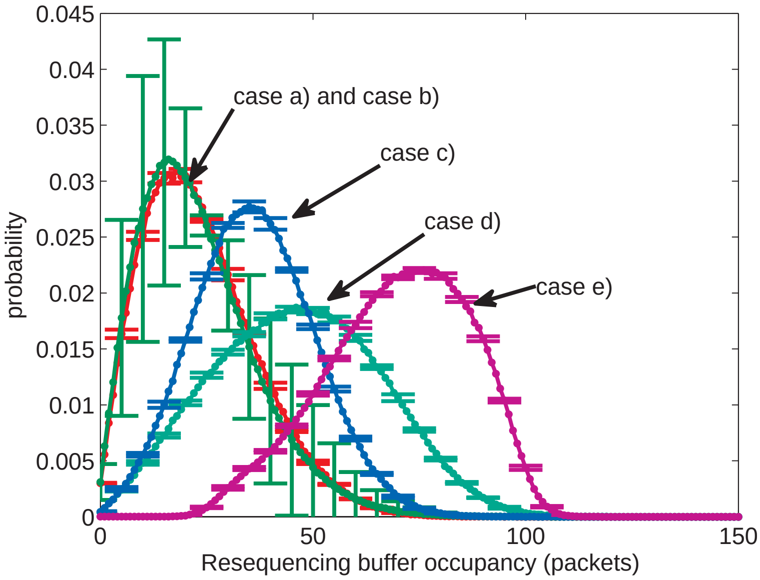

Figure 10: Buffer occupancy in case of 3 concurrent paths

Acknowledgments

The authors would like to thank Dirk Staehle and Tobias Hoßfeld for the fruitful discussions and their support during the course of this work.

References

[1] S. Rixner, "Network virtualization: Breaking the performance barrier," *ACM Queue*, Jan./Feb. 2008.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *IEEE Computer*, Apr. 2005.

[3] GENI Planning Group, "Geni design principles," *IEEE Computer*, Sep. 2006.

[4] Y. L. J. He, R. Zhang-Shen, C.-Y. Lee, J. Rexford, and M. Chiang, "DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet," in *Proc. 4th ACM International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT 2008)*, (Madrid, Spain), Dec. 2008.

[5] Y. Zhu, R. Zhang-Shen, S. Rangarajan, and J. Rexford, "Cabernet: Connectivity Architecture for Better NetworkServices," in *Proc. of the ReArch'08 Workshop at the 4th ACM International Conference on emerging Networking EXperiments and Technologies (ACM CoNEXT 2008)*, (Madrid, Spain), Dec. 2008.

[6] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Re-sequencing Buffer Occupancy of a Concurrent Multipath Transmission Mechanism for Transport System Virtualization," in *Proc. of the 16. ITG/GI - Fachtagung Kommunikation in Verteilten Systemen 2009 - KiVS 2009*, (Kassel, Germany), Mar. 2009.

[7] R. Daley and J. Dennis, "Virtual Memory, Processes, and Sharing in MULTICS," *Communications of the ACM*, vol. 11, May 1968.

[8] K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. Wetherall, "Improving the reliability of internet paths with one-hop source routing," in *Proc. OSDI'04*, (San Francisco, Ca., USA), Dec. 2004.

[9] J. Lane and A. Nakao, "Sora: A shared overlay routing architecture," in *Proceedings of the 2nd International Workshop on Real Overlays And Distributed Systems (ROADS)*, (Warsaw, Poland.), Jul. 2007.

[10] J. Lane and A. Nakao, "Best-Effort Network Layer Packet Reordering in Support of Multipath Overlay Packet Dispersion," in *Proc. of Globecom08*, (New Orleans, LA.), Nov./Dec. 2008.

[11] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network Virtualization: Implementation Steps Towards the Future Internet," in *Proc. of the Workshop on Overlay and Network Virtualization at KiVS 2009*, (Kassel, Germany), Mar. 2009.

[12] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can isps and p2p systems co-operate for improved performance?.," *ACM SIGCOMM CCR*, vol. 37, Jul. 2007.

[13] Y. Nebat and M. Sidi, "Resequencing considerations in parallel downloads," in *Proc. 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2002.

[14] "Ilog cplex." http://www.ilog.com/products/cplex/.

[15] T. Hoßfeld, D. Hock, P. Tran-Gia, K. Tutschku, and M. Fiedler, "Testing the IQX hypothesis for exponential interdependency between qos and qoe of voice codecs iLBC and g.711," in *18th ITC Specialist Seminar on Quality of Experience*, (Karlskrona, Sweden), may 2008.

[16] L. Ong and J. Yoakum, "RFC 3286: An Introduction to the Stream Control Transmission Protocol (SCTP)." Information available at http://www.ietf.org/, May 2002.

[17] H.-Y. Hsieh and R. Sivakumar, "pTCP: An End-to-End Transport Layer Protocol for Striped Connections," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP 2002)*, (Paris, France), Nov. 2002.