Stochastic Relay Routing in Peer-to-Peer Networks

Xiaojun Hei
Department of Electrical & Electronic Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
heixj@ee.ust.hk

Haiqing Song
Lingnan College
Sun Yat-sen University
No.135, West Xin'gang Road, Guangzhou, China
songhq@mail.sysu.edu.cn

Abstract

Network Address Translation (NAT) commonly prevents nodes without globally valid IP addresses from establishing direct Internet paths. In peer-to-peer networks, peers may utilize intermediate nodes as relays for this NAT traversal. We develop a stochastic relay routing algorithm for selecting appropriate relay nodes. The proposed relay routing algorithm is constructed in a stochastic programming framework by leveraging the actual delay of local links and the statistical delay distributions of non-local overlay links. Single or multiple paths are established via relays between two peers for achieving packet delivery with low delay and small loss. The simulation results showed that the proposed stochastic single/multi-path routing algorithm achieved a much lower packet delay than deterministic shortest path algorithms, which utilize average link delays. We demonstrated the effectiveness of the path diversity provided by our algorithm in reducing packet loss significantly via simulations. Our algorithm is fully distributed and requires only accurate local information. The employment of our algorithm is beneficial for provisioning real-time streaming applications in peer-to-peer networks.

I. Introduction

Network Address Translation (NAT) commonly prevents two nodes behind NATs from direct communication [1]. A recent measurement study shows that a large fraction of KaZaA peers, as high as 30%, reside behind NATs [2]. However, in P2P networks, every peer should be capable of serving another peer. For example, if peer A wants to download a file from peer B behind NAT, peer A has difficulties in initiating a direct TCP connection to peer B. Though various NAT traversal techniques have recently been proposed to address this issue [1], tho most reliable method is to utilize intermediate nodes with public IP addresses as relays for traversing NATs. Peer-to-peer networks, which utilize relay nodes, include Skype [3], KaZaA [4] and Grid overlays [5], etc.. However, the performance of peer communication may suffer significantly if relay nodes are not selected appropriately.

Many real-time streaming applications are now running over P2P networks with stringent delay and loss requirements [3]. It is challenging to guarantee both loss and delay performance in relay routing for real-time streaming applications. The classic shortest path routing can be extended to be adaptive to network congestion by modifying link metrics, i.e., [6]. These dynamic routing algorithms have potential applications for relay selection in P2P networks. Each peer constantly measures the quality of overlay links to other peers and acquires link states, and then exchange these link states between peers. Nevertheless, an overlay link may consist of many physical links. Routing updates take non-negligible delay to be exchanged between peers; therefore, the state information maintained in peers is likely to be stale and inaccurate. With inaccurate state information, the performance of dynamic routing may degrade significantly [7].

Stochastic shortest path routing is resilient to inaccurate network state information. A broad overview can be found in [8]. Song [9] studied a new class of stochastic shortest path problems in which the routing decisions are made progressively based on both the current delay and the expected delays in the future. In this paper, we extend the K Stochastic Shortest Path (KSSP) algorithm, previously developed in [9], and address some practical issues for applying it in P2P networks to achieve fast packet delivery and small packet loss. In particular, we developed a label-correcting algorithm for computing the expected delays of the K stochastic shortest paths in generic networks.

Unlike conventional dynamic routing, our proposed algorithm requires much less signalling overhead for updating the system states in that it only requires state information exchange locally between one peer and its neighbors. Stochastic routing was also studied in [10] in ad-hoc networks; however, our work differs significantly from [10] due to distinct networking environments. Hence, the problem statements and the corresponding methodologies are largely different. Our algorithm is applicable for point-to-point communication channels between peers in P2P networks while [10] heavily exploits the broadcast link property in ad-hoc networks.

A single path may not have sufficient network resources for provisioning multimedia services. This may result in packet delay and loss. To combat packet loss, various coding techniques, i.e., Forward Error Correction (FEC), have been proposed to reduce loss by exploiting the path diversity provided by multi-path streaming [11]. In previous multi-path routing algorithms, loss is the major performance metric and delay is considered to be a less important metric due to buffering mechanisms commonly equipped in streaming applications. Our KSSP algorithm is delay-sensitive and may establish multiple paths to achieve path diversity; hence, we can utilize FEC coding techniques to improve the loss performance. Note that there is no theoretical difficulty to apply other coding techniques, such as Multiple Description Coding (MDC) and Digital Fountain Coding, et al., jointly with the KSSP algorithm.

The paper is organized as follows. First, we formulate the stochastic routing problem in Section II. Next, we propose the stochastic single-path routing algorithm in Section II-A and the stochastic multi-path routing algorithm in Section II-B. In Section III, the performance of the proposed routing algorithm is evaluated and compared with some existing routing algorithms via simulations. Finally, conclusion remarks are provided in Section IV.

II. STOCHASTIC ROUTING PROBLEM

In P2P networks, the quality of overlay links between peers in general is time-varying due to cross traffic competing for bandwidth along the path. Internet paths have stationary delay/loss performance in an hour time scale [12]. Peers may measure the link performance via active probing or passively monitoring. Multiple measurements provide stochastic properties of overlay links. Given stochastic link characteristics, a P2P network is able to react to network congestion and maintain a satisfactory QoS for end-users. For example, in Fig. 1, for a connection from S to D, the default Internet path is the dot-dash line "$S \rightarrow D$". Due to the NATs between two peers, this default Internet path may not be established between peer S and D. However, multiple overlay paths exist between this node pair (S, D) if other peers with public IP addresses can be used as relays. In addition, during network congestion period of the default Internet path, it is beneficial to transfer data using the overlay path "$S \rightarrow R_1 \rightarrow D$" or "$S \rightarrow R_2 \rightarrow D$", in order to maximize the quality perceived by end users.

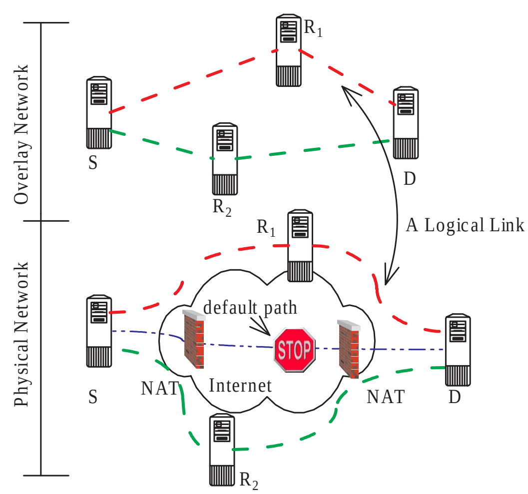


Fig. 1. A P2P network using relay nodes for NAT traversal.

Multimedia communication includes voice and video. Voice connections have small bandwidth requirements, i.e., a 10Kbps connection already has a good voice quality as long as the end-to-end packet delay can be bounded within 150ms; however, real-time video streaming is much more challenging in that it requires both high bandwidth and low delay. In Section II-A, we consider the stochastic single-path routing problem and leave the stochastic multi-path routing problem in Section II-B.

A. Stochastic Single-path Routing

Denote an overlay topology $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of peer/relay nodes and $\mathcal{A}$ is the set of overlay links. Denote D_{ij} as the delay of link (i, j). The probability distribution function of link delays can be estimated between peer nodes via measurements. We assume that the probability distribution functions of the delays of all links are known; however, we do not assume that one node knows the exact delay value of all overlay links at a particular time. We also assume that the distribution functions of link delays are all discretized. Note that this assumption does not bring critical limitations for deploying our algorithm, since we can always approximate a continuous probability distribution function with a discrete one with sufficient accuracy.

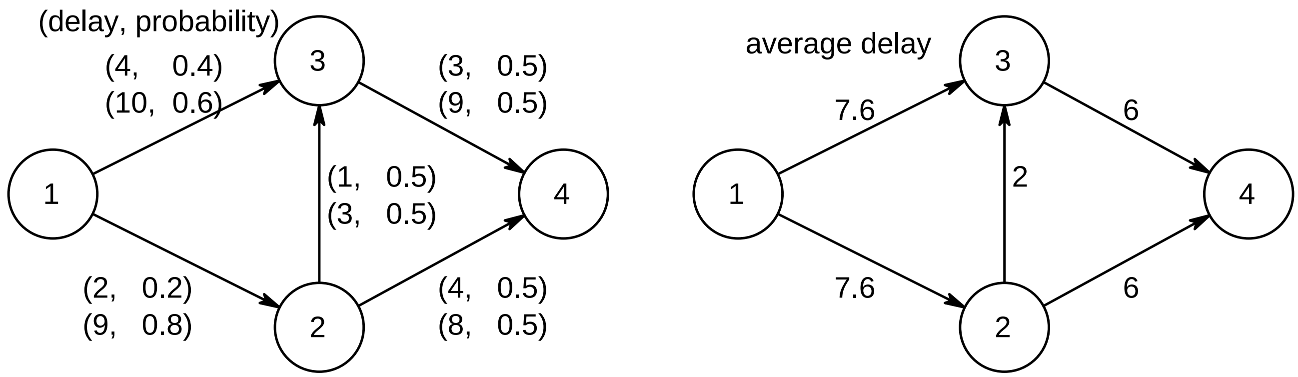
Define

$$\begin{aligned}
\mathcal{R}_{ij} &= \{1, 2, \ldots, r_{ij}\} = \text{the set of all possible delay states of link } (i, j), \\
d_{ij}^r &= \text{actual value of } D_{ij} \text{ under delay state } r,\ r \in \mathcal{R}_{ij}, \\
p_{ij}^r &= \text{probability that delay } r \text{ occurs.}
\end{aligned}$$

In our stochastic routing framework, packets are forwarded from the source to the destination as follows. Upon a packet arrival at node i, we assume that the delay values of the local links are known and accurate; however, the delay states of other non-local links are unknown. We define a decision rule of selecting a successor node at node i: the summation of the actual delay to that successor node and the expected delay from that successor to the destination is minimized. Note that local delay of one overlay hop is easy to measure and maintain correctly. Different delay observations of the local links may result in different selection of a successor node.

For the illustration purpose, we provide an example in Fig. 2(a). Each link has two delay states. Packets travel from node 1 to node 4 via relay 3 or relay 4. The issue here is to select a relay node to minimize the delay from node 1 to node 4.

Our computation procedure is in a reverse order from the destination node 4 to the source node 1. We start at node 3. If a packet arrives at node 3, it can be forwarded to the destination node 4 via link $(3, 4)$. The expected delay from node 3 to node 4 is $3 \times 0.5 + 9 \times 0.5 = 6$. Then we inspect node 2. Suppose that a packet arrives at node 2, this packet can be forwarded to node 4 directly via link $(2, 4)$, or be forwarded to node 3 for further delivery to node 4. The selection decision is based on the real-time observation of the delays of link $(2, 3)$ and link $(2, 4)$. When a packet arrives at node 2, there are four possible delay observations. The packet selects a different link to proceed based on the following delay observation cases.



(a) Stochastic routing (b) Deterministic routing

Fig. 2. An illustrative example

Case 1: $D_{23} = 1$ and $D_{24} = 4$. The packet selects link $(2, 4)$ because $D_{23} + E[D_{34}] = 1 + 6 > D_{24} = 4$. This leads to a delay of 4 units from node 2 to node 4.

Case 2: $D_{23} = 1$ and $D_{24} = 8$. The packet selects link $(2, 3)$ with the expected delay of 7 units (including the exact delay of 1 unit from node 1 to node 3 and the expected delay of 6 units from node 3 to node 4).

Case 3: $D_{23} = 3$ and $D_{24} = 4$. The packet selects link $(2, 4)$ because $D_{23} + E[D_{34}] = 3 + 6 > D_{24} = 4$. This leads to a delay of 4 units from node 2 to node 4.

Case 4: $D_{23} = 3$ and $D_{24} = 8$. The packet selects link $(2, 4)$ because $D_{23} + E[D_{34}] = 3 + 6 > D_{24} = 8$. This leads to a delay of 8 units from node 2 to node 4.

Assume that each observation occurs evenly with a probability of 0.25. Then the expected delay from node 2 to node 4 is 5.75. We call this delay as the expected delay of the stochastic shortest path from node 2 to node 4.

Finally, we inspect node 1. There are four possible delay observations of link $(1, 2)$ and link $(1, 3)$. The delay information at node 1 is shown in Table I. Following the same computation procedure at node 2, we compute the expected delay from node 1 to node 4 as 11.83 units.

TABLE I

STOCHASTIC ROUTING ALGORITHM AT NODE 1

Case	Delay observation at link $(1 \rightarrow 2)$	Delay observation at link $(1 \rightarrow 3)$	Decision (successor node)	Total delay (node $1 \rightarrow$ node 4)
1	$D_{12} = 2$	$D_{13} = 4$	node 2	7.75
2	$D_{12} = 2$	$D_{13} = 10$	node 2	7.75
3	$D_{12} = 9$	$D_{13} = 4$	node 3	10
4	$D_{12} = 9$	$D_{13} = 10$	node 2	14.75

We replace the delays of each link in Fig. 2(a) with the average link delay as shown in Fig. 2(b). We can compute the expected shortest path from node 1 to node 4 $(1 \rightarrow 3 \rightarrow 4)$ as $7.6 + 6 = 13.6$ units. This delay difference

$13.6 - 11.83 = 1.77$ units is the gain by jointly considering the real time delay observation of local links and the stochastic delay distribution of non-local links. Song in [9] proved that the expected delay of a stochastic shortest path from a node to the destination node is a lower bound of the delay of the corresponding deterministic shortest path using average link delays. Therefore, the proposed stochastic routing algorithm is guaranteed to outperform the deterministic shortest routing; nevertheless, the exact performance gain depends on the actual delay distribution.

We summarize the decision rules of stochastic single-path routing in a stochastic programming framework. At each node, one decision rule maps a delay observation value of all outgoing links to a successor node. Note that the number of observations may increase exponentially with an increasing node number. Therefore, it is not efficient to store the decision rules explicitly as shown in Table I. Instead, we implicitly express the decision rule with a much smaller storage size as follows.

Let v_j be the expected delay of the stochastic shortest path from node j to the destination and $S(j)$ be the set of all successors of node j. The decision rules at node i can be expressed: for any observation $\{D_{ij}^o : j \in S(i)\}$, we select the successor node j' which satisfies

$$j' = \arg\min_{j \in S(i)} \{D_{ij}^o + v_j\}. \tag{1}$$

Hence, we only need to store v_j for each node in the network. When a packet tries to find a next hop, it looks up the expected value v_j of the successor nodes at its current staying node and using Equation 1 to select the successor node with the minimum delay.

B. Stochastic Multi-path Routing

The path diversity in multi-path routing improves the loss performance in streaming applications. Each stochastic path specifies a path to the destination under any delay observations of the whole network. Two stochastic paths are distinct if for each possible observation of the link delays of the whole network, these two paths differ for at least one link. The stochastic shortest path is defined as the path that has the minimum expected delay. The k-th stochastic shortest path is the one that has the minimum expected delay among all stochastic paths that are distinct from the first $k-1$ stochastic paths. Let ω_i be the outcome (observation) of the delays of all links out of node i and Ω_i be the set of all these possible outcomes. Denote v_i^k as the expected delay of the k-th stochastic shortest path from node i to the destination node n. The K stochastic shortest path problem is formulated in Equation 2.

Definition 1:

$$v_n^k = \begin{cases} 0 & \text{if } k = 1, \\ \infty & \text{otherwise;} \end{cases}$$
$$v_i^k = \mathrm{E}\left[\min^k \left\{ D_{ij}(\omega_i) + v_j^\ell : \ell = 1, 2, \ldots, k, j \in S(i) \right\}\right],$$
$$k = 1, 2, \ldots, K, i \in \mathcal{N} \backslash \{n\}, \tag{2}$$

where $\min^k$ means the k-th smallest value of a set.

We illustrate the computation procedure of v_i^k as follows. Note that v_i^k is defined in a recursive manner. Given the expected delays of the K shortest stochastic paths from all successor nodes of node i to the destination, we compute the expected delays v_i^k, $k = 1, 2, \ldots, K$. In Fig. 3, suppose that $K = 2$ and the expected delays of the stochastic paths from node 2 and node 3 to the destination node 4 are already computed, $v_2^1 = 5.75$, $v_2^2 = 8.25$ and $v_3^1 = 6$. Let $V_i^k(\omega_i)$ represent the delay of the k-th stochastic shortest path from node i to the destination under outcome ω_i. Hence, $V_i^k(\omega_i) = \mathrm{E}\left[\min^k \left\{ D_{ij}(\omega_i) + v_j^\ell :\ \ell = 1, 2, \ldots, k,\ j \in S(i) \right\}\right]$. Table II shows all the possible outcomes and the corresponding delays of the two shortest stochastic paths. Given the uniform probabilities of all these four possible outcomes, we have

$$v_1^1 = \sum_{q=1}^{4} \mathrm{Pr}(\omega_1^q) \cdot V_1^1(\omega_1^q) = 11.83,$$
$$v_1^2 = \sum_{q=1}^{4} \mathrm{Pr}(\omega_1^q) \cdot V_1^2(\omega_1^q) = 14.45.$$

Since the number of possible outcomes for a node may grow exponentially with the number of links going out of that

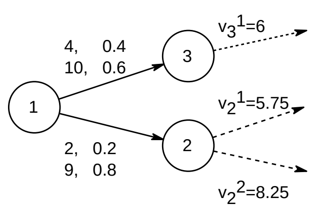


Fig. 3. A computation example of the expected delays of K stochastic shortest paths

TABLE II

STOCHASTIC SHORTEST PATH DELAYS UNDER ALL POSSIBLE OUTCOMES

ω_1	$D_{12}(\omega_1)$	$D_{13}(\omega_1)$	$D_{12}(\omega_1) + v_2^1$	$D_{12}(\omega_1) + v_2^2$	$D_{13}(\omega_1) + v_3^1$	Pr(ω_1)	$V_1^1(\omega_1)$	$V_1^2(\omega_1)$
ω_1^1	2	4	7.75	10.25	10	0.08	7.75	10.25
ω_1^2	2	10	7.75	10.25	16	0.12	7.75	10.25
ω_1^3	9	4	14.75	17.25	10	0.32	10	14.75
ω_1^4	9	10	14.75	17.25	16	0.48	14.75	16

node, the steps for computing v_i^k may also increase exponentially. Song [9] developed a polynomial-time algorithm for computing the expected delays of the K stochastic shortest paths in acyclic networks. The time complexity of the algorithm is $O(MRK\log(MRK) + MRK^3)$, where M is the number of links, R is the maximum number of possible values of link delays and K is the number of stochastic paths. Because a P2P network may contain cycles, we developed a label-correcting algorithm for computing the expected delays of the K stochastic shortest paths in generic networks, as sketched in Fig. 4.

Step 1: Initialization
  $v_n^1 = 0, v_n^k = \infty, k = 2, 3, \ldots, K,$
  $v_i^k = \infty, k = 1, 2, \ldots, K,$
  $Q^{(1)} = \{n-1\}, Q^{(k)} = \varnothing, k = 2, 3, \ldots, K.$
  Set $k = 1$.
Step 2: Remove a node j from $Q^{(k)}$.
Step 3: For each node $i \in B(j)$ ($B(j)$: the set of predecessor nodes of node j)
  Update $E\left[\min^k\{D_{ij} + v_j^\ell, \ell = 1, 2, \ldots, k, j \in S(i)\}\right]$.
  If $v_i^k < E\left[\min^k\{D_{ij} + v_j^\ell, \ell = 1, 2, \ldots, k, j \in S(i)\}\right]$,
  set $v_i^k = E\left[\min^k\{D_{ij} + v_j^\ell, \ell = 1, 2, \ldots, k, j \in S(i)\}\right]$,
  and add i to $Q^{(k)}, Q^{(k+1)}, \ldots, Q^{(K)}$ (if i does not belong to them).
  If $Q^{(k)} \neq \varnothing$, go to Step 2; otherwise, go to Step 4.
Step 4: Set $k \leftarrow k + 1$. If $k \leq K$, go to Step 2; otherwise, stop.

Fig. 4. Sketch of the label-correcting algorithm for computing the expected delays of the K stochastic shortest paths in generic networks

Given the values of v_i^k, we describe the procedure of constructing the stochastic multi-path routing algorithm using the example in Fig. 2(a). Suppose that we have computed $v_3^1 = 6.0$, $v_2^1 = 5.75$ and $v_2^2 = 8.25$. Packets are transferred from node 1 to node 4 using two stochastic shortest paths. Without loss of generality, we assume that packet 1 is transferred to node 4 using the first stochastic shortest path and packet 2 is transferred to node 4 using the second stochastic shortest path. Suppose that the delay observations for both packet 1 and packet 2 at node 1 are $D_{12} = 2$ and $D_{13} = 4$. Packet 1 is sent to node 2 and packet 2 is sent to node 3 because $D_{12} + v_2^1$ is the minimum value and $D_{13} + v_3^1$ is the second minimum value. If the observations at node 1 are that $D_{12} = 2$ and $D_{13} = 10$, both packet 1 and packet 2 are forwarded to node 2 because $D_{12} + v_2^1$ is the minimum value and $D_{12} + v_2^2$ is the second one. When these two packets arrive at node 2, packet 1 uses the first stochastic shortest path and packet 2 uses the second stochastic shortest path. Hence, these two packets depart at node 2. By the definition of K stochastic shortest paths, two packets, which traverse along two different stochastic paths, have at least one distinct link upon arrival at the destination. We summarize this stochastic multi-path routing algorithm in Fig. 5.

Step 1: *Initialization*

$current\_node \Leftarrow source\_node, path\_id \Leftarrow packet\_id \ mod \ K.$

Step 2: *Update the expected delays of the stochastic shortest paths*

Let $i = current\_node, k = path\_id$, find j' and ℓ' such that

$D^o_{ij'} + v^{\ell'}_{j'} = \arg\min^k \{D^o_{ij} + v^{\ell}_j : j \in S(i), \ell = 1, 2, \ldots, k\}$

Step 3: Set $current\_node \Leftarrow j'$, $path\_id \Leftarrow \ell'$. If $current\_node = destination\_node$, stop; otherwise, go to Step 2.

Fig. 5. The proposed stochastic multi-path routing algorithm

III. PERFORMANCE EVALUATION

In this section, we evaluate the delay and loss performance of the proposed KSSP algorithm via simulations. The whole simulation model was constructed based on SimLib 2.4 [13], developed at IKR, University of Stuttgart.

A. Loss and Delay Models of Overlay Links

Peer nodes are end-hosts on the Internet. The overlay links between node S, R_i, D are logical links. One logical link may actually consist of many physical hops; therefore, these logical links may exhibit quite different loss and delay characteristics from the physical links. An accurate model for packet loss over the Internet is quite complex. Nevertheless, a two-state Markov chain model, known as the Gilbert model can approximate the behavior of packet loss over the Internet reasonably accurately [14]. This model captures the dependence in consecutive packet losses. We assume that the loss process on an overlay link is modelled using the Gilbert loss model.

In the Gilbert loss model, a stationary continuous time Markov chain characterizes the potential correlations between consecutive losses on a path. The packet loss process along link k is described by a two-state continuous time Markov chain $\{X_k(t)\}$, where $X_k(t) \in \{0, 1\}$. If a packet is transmitted at time t when the state of link k is $X_k(t) = 0$, then no packet loss occurs. However, the transmitted packet is considered lost if $X_k(t) = 1$. Let p denote the probability of jumping from state 0 to state 1, and let q denote the probability of jumping from state 1 to state 0. The infinitesimal generator for this Gilbert model of link k is

$$Q_k = \begin{pmatrix} -\mu_0(k) & \mu_0(k) \\ \mu_1(k) & -\mu_1(k) \end{pmatrix}.$$

The stationary distribution of this Gilbert model is $\pi(k) = [\pi_0(k), \pi_1(k)]$, where $\pi_0(k) = \mu_1(k)/(\mu_0(k) + \mu_1(k))$ and $\pi_1(k) = \mu_0(k)/(\mu_0(k) + \mu_1(k))$. Unless stated explicitly, we use $\mu_0(k) = 20$ and $\mu_1(k) = 70$, for link k by default.

The delay statistics of overlay links can be measured by peer nodes. For simplicity, we assume the packet delay of each overlay link follows a uniform distribution. The lower bound of this uniform distribution is randomly generated between 5ms and 15ms, and the upper bound is randomly generated within 180ms and 220ms. In addition, we assume that the delay and loss are uncorrelated on overlay links.

B. Simulation Configurations

We evaluated the KSSP algorithm in two overlay network topologies, depicted in Fig. 6. In Fig. 6(a), packets are sent from node S to node D via only one relay node. For this network, 5 relay nodes exist between source S and destination R. In Fig. 6(b), packets traverse two relay nodes before reaching the destination R. In the second network, the network has 10 relay nodes arranged in two columns between the source-destination node pair. We

(a) One-tier relay (b) Two-tier relay

Fig. 6. The overlay topology used in simulations.

assume a streaming application which generates packets at a constant rate of 10 packets/sec. These overlay links are assumed to be independent. The packet assignment is carried out in a simple round robin fashion, e.g., if we use 3 paths for packet delivery, the packet sequence number on path 1 is $1, 4, 7, \ldots$.

C. Simulation Results

1) Loss probability of KSSP as a function of FEC parameters: We evaluated the effects of FEC parameters on the loss probability of KSSP. The FEC parameters were varied in two ways: (a) increase the degree of redundancy (e.g., for a given k, increase n); (b)increase n and k but maintain the same ratio of n/k.

We considered a simple FEC scheme, similar as in [11]. A streaming file is divided into groups of data packets such that each group consists of k data packets. A FEC packet group consists of k data packets and $(n-k)$ redundant packets for the FEC protection. If the number of lost packets within a FEC group is less than or equal to $(n-k)$, the original k data packets can be reconstructed within that FEC group. For simplicity, we did not differentiate data and FEC packets.

The effects of FEC parameters on the packet loss probability are illustrated in Fig. 7 for $n/k = 1.125$, 1.25 and 1.5 and different FEC group sizes. With increasing the amount of redundancy (e.g., from $n/k = 1.125$ to 1.5), the packet loss probability reduces significantly. However, increasing the number of data packets in a FEC group (while maintaining the same ratio of n/k) may not necessarily reduce the loss rate. These results confirm the effects of FEC parameters on the loss probability of multi-path routing, reported in [11]. Nevertheless, our work differs from [11] in that the proposed KSSP algorithm considers both the loss and delay performance while the loss performance is the only performance metric in [11].

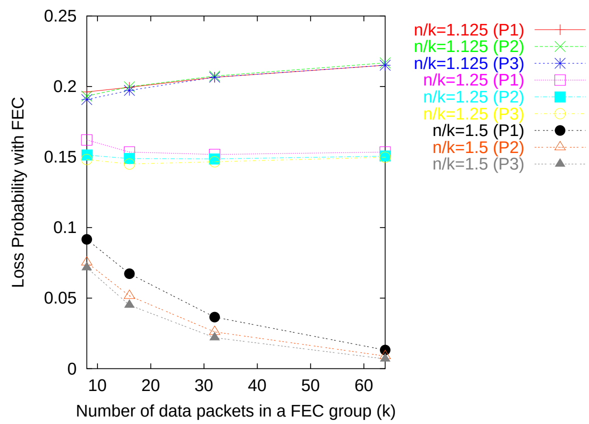
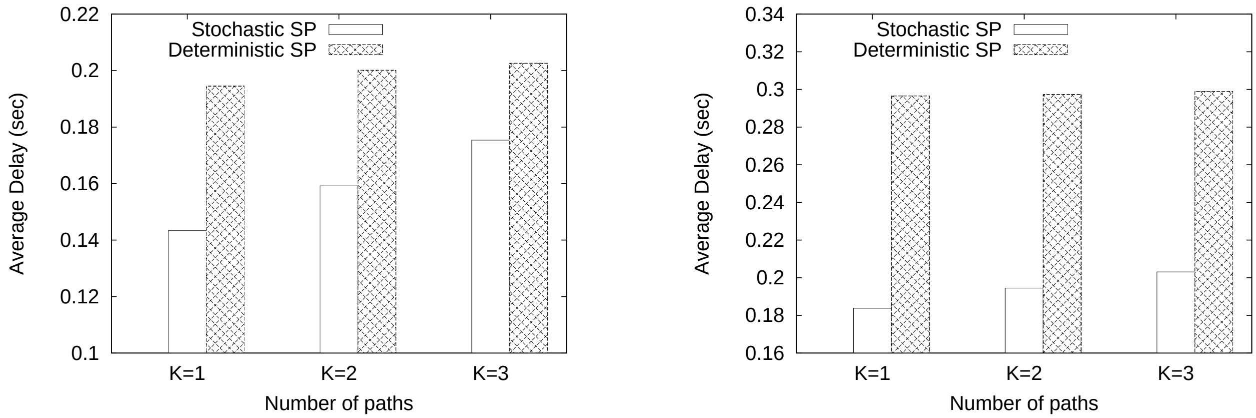


Fig. 7. Loss probability with FEC as a function of n/k and k using one-tier relay.

2) Performance comparison between KSSP and KDSP: In the section, we compare the proposed Stochastic Shortest Path (SSP) routing algorithm with the Deterministic Shortest Path (DSP) in terms of the average delay and loss probability. The FEC group is set as $(10, 8)$. Shown in Fig. 8, KSSP has significantly smaller average delays than KDSP. The average delay of multi-path routing is slightly worse than that of single shortest path routing. However, in Fig. 9 and Fig. 10, the multi-path routing jointly with FEC recovery reduces the loss probability significantly. In particular, the loss probability decreases with the increasing number of paths. When the number of paths increases further, the loss reduction rate slows down. In practice, 2 or 3 paths already reduce the loss probability sufficiently.

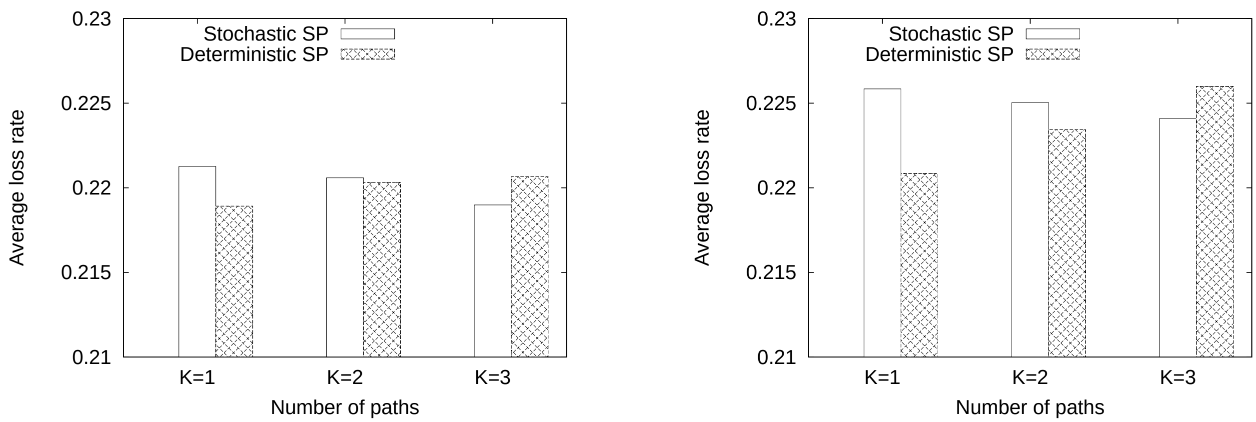
IV. Conclusion

The wide deployment of NAT brings forth difficulties for two nodes behind NATs to establish a direct Internet path. However, peer-to-peer networks require direct communication between peers. Peers may utilize intermediate nodes with public IP addresses as relays for achieving this NAT traversal. Dynamic routing algorithms can be used for selecting appropriate relay nodes; however, significant signal traffic are required to update the network state information. In this paper, we formulated the stochastic relay routing problem and developed a stochastic multi-path routing algorithm by leveraging both the real-time delay of local links and the statistical delay distribution of



(a) One-tier relay (b) Two-tier relay

Fig. 8. Average delay performance



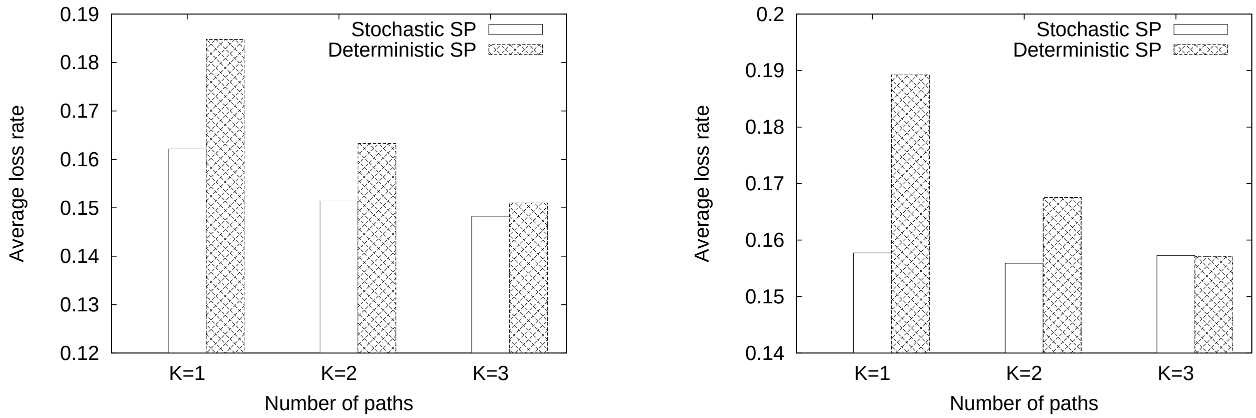
(a) One-tier relay (b) Two-tier relay

Fig. 9. Average packet loss probability without FEC recovery

non-local links to achieve packet delivery with small delay and low loss. We evaluated the proposed algorithm in terms of the average delay and the loss probability via simulations. The results demonstrate that the proposed KSSP algorithm achieves a much lower packet delay than the deterministic shortest path algorithm based on average link delays. In addition, the results indicate that the algorithm is effective in reducing packet loss. This KSSP algorithm is helpful in provisioning real-time streaming applications in P2P networks.

V. Acknowledgment

The authors would like to thank Keith Ross and Yong Liu in Polytechnic University for inspiring the authors to investigate the relay routing problem in peer-to-peer networks. They also thank anonymous reviewers for their constructive comments to improve the quality of the paper.



(a) One-tier relay (b) Two-tier relay

Fig. 10. Average packet loss probability with FEC recovery

REFERENCES

[1] S. Guha and P. Francis, "Characterization and measurement of TCP traversal through NATs and firewalls," in *Proceedings of Internet Measurement Conference*, Berkeley, CA, 19-21 Oct. 2005.

[2] J. Liang, R. Kumar, and K. W. Ross, "The FastTrack overlay: A measurement study," *Elsevier Computer Networks Journal*, vol. 50, no. 6, pp. 842–858, 13 April 2006, special issue on Overlay Distribution Structures and their Applications. [Online]. Available: http://cis.poly.edu/~ross/papers/KazaaOverlay.pdf

[3] "Skype." [Online]. Available: http://www.skype.com/

[4] "KaZaa." [Online]. Available: http://www.kazaa.com

[5] Y. Liu, Y. Gu, H. Zhang, W. Gong, and D. Towsley, "Application level relay for high-bandwidth data transport," in *First Workshop on Networks for Grid Applications (GridNets)*, San Jose, Oct. 2004.

[6] T. Bressoud, R. Rastogi, and M. Smith, "Optimal configuration for BGP route selection," in *Proceedings of INFOCOM*, vol. 2, San Francisco, USA, 30 March-3 April 2003, pp. 916 – 926.

[7] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 384–398, 2003.

[8] D. Bertsekas, *Dynamic programming and optimal control*, 2nd ed. Belmont, Mass. : Athena Scientific, 2001.

[9] H. Song, "Dynamic flow management problems in logistics networks," Ph.D. dissertation, Industrial Engineering and Engineering Management, Hong Kong University of Science and Technology, Jan. 2005. [Online]. Available: http://repository.ust.hk/dspace/handle/1783.1/2433

[10] C. Lott and D. Teneketzis, "Stochastic routing in ad-hoc networks," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 52 – 70, Jan. 2006.

[11] L. Golubchik, J. C. S. Lui, T. F. Tung, A. L. H. Chow, W.-J. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: what are the benefits?" *Performance Evaluation*, vol. 49, no. 1-4, pp. 429–449, 2002.

[12] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. of ACM SIGCOMM Internet Measurement Workshop 2001*, San Francisco, CA, Nov. 2001.

[13] IKR. University of Stuttgart, IKR Simlib library 2.4. [Online]. Available: http://www.ind.uni-stuttgart.de/INDSimLib/

[14] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for internet telephony," in *Proceedings of INFOCOM*, vol. 3, Mar. 1999, pp. 1453 – 1460.