Computer Networks 55 (2011) 1024–1035

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Introducing multipath selection for concurrent multipath transfer in the future internet

Jianxin Liao [a], Jingyu Wang [a,*], Tonghong Li [b], Xiaomin Zhu [a]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, PR China
[b] Technical University of Madrid, Madrid 28660, Spain

ARTICLE INFO

Article history:
Received 30 April 2010
Received in revised form 20 October 2010
Accepted 13 December 2010
Available online 21 December 2010

Keywords:
Future Internet
SCTP
Multipath selection
Path diversity
Shared bottleneck

ABSTRACT

It is essential for the Future Internet to fully support multihoming and select most appropriate paths for Concurrent Multipath Transfer (CMT). In real complex networks, different paths are likely to overlap each other and even share bottlenecks which can weaken the path diversity gained through CMT. Spurred by this observation, it is necessary to select multiple independent paths insofar as possible. However, the *path correlation* lurks behind the IP/network layer topology, so we have to fall back to end-to-end probes to estimate this correlation by analyzing path delay characteristics. In this paper, we present the first step towards a new topic of *correlation-aware multipath selection*, with formal and systematic problem definition, modeling and solution. Based on a well-designed delay probing, a Grouping-based Multipath Selection (GMS) mechanism is developed to avoid underlying shared bottlenecks between topologically joint paths. In addition, we further propose a practical functionality framework and define a novel *multihoming sublayer* for the exchange of the multipath capabilities. Extensive simulations demonstrate that the GMS under different network conditions performs much better than other selection schemes, even with burst background traffic.

1. Introduction

The research on the Future Internet [1] ranges from small, incremental evolutionary steps to complete redesigns (clean-slate) [2] and architecture principles, among which the evolutionary approaches supplement the existing Internet technologies, such as Mobile IP, HIP [3], SCTP [4], SHIM6 [5] and Multipath TCP [6,24]. One of the most challenging design goals of the Future Internet can be characterized by the support of multi-homing between various access networks with an aim of providing a broad range of services to the users anywhere, anytime. A host equipped with multiple interfaces can connect (potentially through different providers) to multiple networks simultaneously, and we call this functionality multihoming [7]. The use of several IP addresses on each end-host will become widely prevalent; this is a drastic architectural change compared to today's non-multihomed networks, where each host is typically identified by a single IP address that denotes both its identifier and its locator. When multiple access networks are nested together, to ensure that application requirements can be met effectively, it is essential that more than one path can be selected to transmit data simultaneously, which is concurrently called Concurrent Multipath Transfer (CMT) [8]. Doing so can provide: (1) more bandwidth and better resiliency for the user and (2) higher network utilization for network operators. Unfortunately, current IP/TCP/SCTP protocols are not readily capable for multihoming and CMT. The idea behind CMT is to make use of the additional paths that are ignored by the current routing system.

* Corresponding author.
E-mail addresses: liaojx@bupt.edu.cn (J. Liao), wangjingyu@bupt.edu.cn (J. Wang), tonghong@fi.upm.es (T. Li), zhuxm@bupt.edu.cn (X. Zhu).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.12.010

In CMT, a path is expressed as a pair of source–destination (S–D) addresses for reaching one destination. When a host has multiple parallel paths to send packets to its destinations, it must somehow make a decision of which path(s) to use for which connection(s) on a per-application basis. More specifically, the host needs a mechanism to select the source IP addresses and the destination IP addresses. We consider it a natural requirement that a selection scheme running on the end-hosts should choose a subset of paths to bear the requirements of upper applications, by providing a good balance between complexity and performance. The problem here is essentially how the paths can be suitably selected to bear a given data stream, when QoS, policy, security and reliability concerns are taken into account. Compared to other statements on path selection, our path selection actually is to select an S–D IP address pair rather than the entire route, and our multipath selection is to select multiple S–D IP address pairs between two multihomed hosts.

Most research about multipath transmission make the assumption that multiple paths are independent [6,8], but this assumption is rarely valid in real networks. For example, two different paths are likely to overlap one or more joint links somewhere in the network, even share the same bottleneck. So it is necessary to diminish this assumption and take into account the correlation between paths [9]. Furthermore, the benefits of path diversity do not just depend on whether paths are absolutely independent or dependent, but rather on their correlated degrees in actual networks. Evaluating correlation degrees of available paths and selecting relatively independent paths if possible is an important element in effective use of path diversity, which is partly motivated by the observation that packets sent over dependent paths are likely to suffer simultaneously from large packet delays, and otherwise not. Therefore, we can conclude that if the delay variation on different paths are strongly (or weakly) correlated, the internal shared congestion is more (or less) likely to occur. It is reasonable to model the path correlation based on the path delay variation, and what we need to do is to collect a history of one-way delay values of each forward path through external end-to-end measurements, without cooperation from the network routers.

Intuitively, we can view the selection of a highly reliable set of end-to-end paths as the problem of maximizing the effect of path diversity for a parallel-series network. Path bottleneck points are the most critical to impact the performance of the entire path, and their relative locations directly affect the degree of path correlation. Therefore it is crucial to identify bottlenecks in the large-scale network so as to evaluate path correlation. This paper makes the following three key contributions. Firstly, we focus on a new topic about how to select several paths from multiple available paths in a multihomed network environment, without any knowledge of the underlying network. Secondly, we propose a probing scheme capable of discovering shared bottlenecks among multiple paths simultaneously, and a subsequent Grouping-based Multipath Selection (GMS) strategy. Finally, we present a system implementation to enforce the multipath selection and transfer, and discuss the necessity of adding a *multihoming sublayer* below the transport layer to manage multiple paths.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 states the problem and main intuition behind our mechanism. In Section 4, we propose the path correlation model and probing method. Section 5 explains the proposed GMS, and assesses its computational complexity. An implementation framework is proposed in Section 6. Section 7 presents simulation results that show the advantages of our approach. Some discussion and open issues for future study are presented in Section 8. Section 9 concludes the paper.

2. Related work

The exploitation of path diversity has attracted much attention recently, and [9] provides a broad overview of the general area. We note that existence of multiple disjoint paths can result in many benefits including: (1) increased bandwidth, and (2) improved loss characteristics. There are a number of approaches [7–9] to accomplishing multipath data delivery, the path diversity-based approach is considered in this paper.

Multipath routing [10–12], especially for wireless ad hoc networks, focuses on how to leverage multiple complete paths through a network. In [10], Disjoint Pathset Selection Protocol (DPSP) is proposed for selecting a set of paths to achieve the best reliability. Mao et al. [11] further propose a meta-heuristic approach based on Genetic Algorithms to solve the routing selection problem. Wei and Zakhor [12] propose a different method for selection of two node-disjoint paths that takes into account the interference caused by the neighboring links.

Selecting optimal paths in overlay networks has also been an active research area recently [13–15]. Begen et al. study how to select multiple paths that maximize the video quality at clients on Internet overlay networks [13]. Given information about the underlying network graph, [14] proposes multipath routing heuristics for unicast and multicast scenarios along with a data scheduling algorithm. In [15], the authors propose to select two paths with minimal correlation for streaming over Internet overlay networks.

There are other similar works in interface [16], access network [17] and IP address [4,7] selection for multihomed wireless device. Historically, this was good, as the first link was usually the bottleneck which had the least bandwidth. Often now, however, it is a “backhaul” rather than the access link that has the most constrained bandwidth – an example of this could be a satellite or 3G link which connects a train WLAN to the internet. Therefore, the target should be how to select end-to-end complete paths instead of merely part of them. Another work in [18] aims at selecting the best path among several available end-to-end paths through the use of bandwidth estimation techniques, which is more suited to the single path selection.

Multipath selection needs to take advantage of the benefits of path diversity, so discovering the correlation characteristics of multiple paths is the key problem. It can be done either by internal nodes or by end systems. The aforementioned approaches attempt to learn about single path characteristics, but do not address directly the problem of identifying the correlations between multiple paths. Unlike others, Rubenstein et al. [19] attempt to detect whether two flows share the same bottleneck through end-to-end measurement. However, their goal is to exploit the relation between the flows rather than the paths.

The following features distinguish our work from other approaches in the literature: (1) the correlations between multiple paths are considered for multipath selection; (2) completely end-to-end, without the need of support at routers; (3) being suitable for all kinds of path relationship models, not just single source; and (4) fast evaluation, low load and high scalability to more than two paths.

3. Problem statement

Consider the multihomed networks are constituted by the multihomed end-devices (see Fig. 1). The source and the destination are connected via a network of communication links. An end-to-end path is a virtual link directly connecting two IP addresses which come from source and destination device respectively. It can be mapped to the IP path. For example, the PathP_{12} started from IP_s^1 and ended with IP_d^2 consists of the nodes N_S, N_m, N_k, and N_D. Characteristics of two end-to-end paths may be correlated because they may share some IP links or nodes. For example, the P_{12} and P_{13} share the IP links (N_S,N_m) and (N_m,N_k).

An *M-by-N* multihomed network topology can be abstracted as a directed acyclic graph $G = (V,E)$ between M source addresses in the source device and N destination addresses in the destination device, along with a given single-path routing policy that maps each source–destination pair to a single route from the source to the destination. Ignoring the topology and physical links of the network, we let P_{ij} simply denote any one path connecting source address IP_s^i and destination addresses IP_d^j. We assume that drops at congestion points are burst due to the Drop-tail nature of most routers, and the packets are dropped in an i.i.d. fashion. Moreover the packet drop processes in different links are independent of each other. We ignore quantization issues, data corruption or random delays.

Our goal is to select the number of paths required by the upper application and at the same time to minimize the correlation of selected path set. Nevertheless, the attempt to select the correlation-minimization path set directly is an *Integer Quadratic Programming* problem (NP-hard [20]), which is an exponential exhaustive search to select paths. In addition, we observe that path bottleneck points are the most critical to impact the performance of the entire path, and their relative locations affect the degree of path correlation directly. Thus, we introduce a pre-grouping process according to whether shared bottlenecks exist or not, and then perform multipath selection among groups which is solvable in a reasonable amount of time. The proposed GMS solution consists of the following four steps: (1) probing for the delay variation of all paths; (2) grouping based on whether these paths exist shared bottlenecks or not; (3) simple selection of the best path from each group; (4) precise selection of the required number of paths based on the paths obtained in step 3, if necessary. In GMS, we get rid of the strongly correlated paths and carry out the multipath selection on a smaller number of candidate paths, since the benefit of path diversity is never gained from paths with a shared bottleneck.

4. Path correlation evaluation

It is desirable to evaluate the correlation degree between two paths. This section gives the definition of path correlation and presents a novel probing scheme capable of determining whether any two paths are *strongly* correlated or not.

4.1. Path correlation model

R_{ij} and R_{xy} are the performance values for two given paths P_{ij} and P_{xy} observed at approximately the same times experienced by receivers. The *path correlation* is defined as

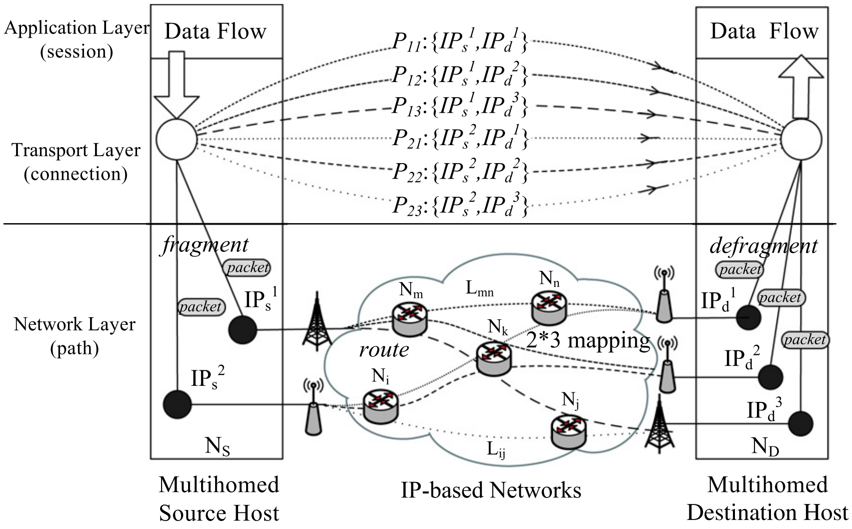


Fig. 1. An example path graph mapping to connectivity graph.

the sample correlation coefficient between P_{ij} and P_{xy} in a normal way as (1).

$$\rho(P_{ij}, P_{xy}) = \frac{Cov(R_{ij}, R_{xy})}{\sqrt{Var(R_{ij})}\sqrt{Var(R_{xy})}}. \tag{1}$$

This formula is based on the standard Pearson's correlation function of two random variables. Here, we utilize the one-way delay on the forward path as a metric to obtain R_{ij} and R_{xy} when calculating *path correlation* in (1). To ensure the accuracy of the calculation, we need a history of delay samples on each forward path. Without the need of network support, path delay can be obtained only through a standard timestamping mechanism, which uses the "Options" field in the TCP header to include the time when a packet is sent by the source, and the time when an ACK is sent by the destination. As the ACK sending time gives an approximate indication of the packet reception time, the one-way delay on the forward path can be estimated as the ACK sending time minus the packet sending time. Path delay properties have been observed to be stationary (on the order of a few minutes) [21]. This justifies our approach of computing (1) based on past behavior to select paths for future use.

Then, we can determine whether P_{ij} and P_{xy} share a common bottleneck or not. In [19], a *cross measure* is defined as the correlation coefficient of sample sets of two different variables, whereas an *auto measure* is defined as the correlation coefficient of two sample sets of the same variable. The *comparison test* between two paths learned from [19] is defined as follows: (1) Compute the *cross measure*, $M_x = \rho(P_{ij}, P_{xy})$, between pairs of packets on two paths P_{ij} and P_{xy}, spaced apart by time $t > 0$. (2) Compute the *auto measure*, $M_a = \rho(P^1_{ij}, P^2_{ij})$, between packets on the same path P_{ij}, spaced apart by time $T > t$. Here P^1_{ij} and P^2_{ij} represent two interleaving samples from the same path P_{ij}. (3) If $M_x > M_a$, then the paths share a common bottleneck, otherwise they do not. The intuition behind this test is that if two paths share a bottleneck, then the cross correlation coefficient should exceed the auto correlation coefficient provided that the spacing between packets on different paths at the bottleneck *is smaller than* the spacing between packets on the same path.

4.2. Correlation probing

In [19], their work can not be applied to more than two flows in which the probing load is too heavy, so it is essential to design a light probing scheme to infer the correlations of multiple paths simultaneously. One of the essential techniques in our construction is the extension of "*packet-pair*" technique [22]. A challenge associated with our approach is how to set probing packet spacing such that multiple pairs of *comparison tests* can be performed in parallel.

Definition 1. An *N-packets-pair* probe sequence S (D_{Sn}; λ; μ; Δ) is a sequence of block pairs with each block including N packets of the same size as shown in Fig. 2. The consecutive N packets are transmitted respectively to different destinations in destination address set D_{Sn} (there are N addresses). The inter-packet spacing within a block is λ time units, the inter-block spacing is at most μ time units and two adjacent N-packets-pairs are spaced by at least Δ time units.

The intuition behind the structure is to provide a baseline for the delay correlation over each path of the same source. The key insight is that because of their temporal proximity, as the quantity of addresses in a destination host (N) is limited to a small number, we expect packets within an *N-packets-pair* to have a high probability of experiencing a shared fate on the shared links. This well-designed structure ensures that if both paths from the same source share a bottleneck, then the spacing of packet-pair on the same path (T) at this bottleneck is larger than the spacing between packets on different paths (t). Thus, the precondition of *comparison test* can be satisfied. The values μ and Δ in Definition 1 are chosen empirically in order that the intra-pair and inter-pair packets highly experience correlated and dependent packet delays, respectively.

For a probing session initiated by a single source address, the set of destination addresses are treated as probing terminal points of a probe tree, in which each two branches forms the Inverted-Y topology, and the above *N-packets-pair* sequence is used for probing. For M source addresses, a similar probing sequence is sent from every source in parallel and simultaneously as in Fig. 3. As each source constructs one probe tree, there are M trees altogether. The branches from different trees converge at a certain junction, which forms the Y topology.

4.3. Correlation computation

To compute M_a and M_x, we should find a sequence of matched packets from two paths, which should arrive at that bottleneck at roughly the same time if both paths have a shared bottleneck. A key step in this process is synchronizing and matching the sample sets from two paths. The sampling process proceeds as follows. Assume path P_{ij} yields n_{ij} samples, and path P_{xy} yields n_{xy} samples. Without loss of generality, assume that $n_{ij} \leqslant n_{xy}$. Let $x(u)$ denote the timestamp (arrival time) of sample R_{ij} from P_{ij}, and $y(v)$ denote the timestamp of sample R_{xy} from P_{xy}, where $1 \leqslant u \leqslant n_{ij}$, and $1 \leqslant v \leqslant n_{xy}$. We merge two sets $x(u)$ and $y(v)$ and compute the mean spacing for all packets on the

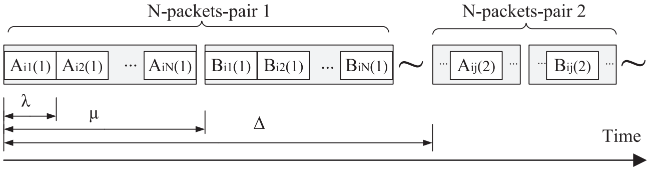


Fig. 2. *N*-packets-pair probe sequence.

two paths, t, between every two consecutive packets on P_{ij} and P_{xy}. That is $t = (\sum_{1\leqslant u\leqslant n_{ij},1\leqslant v\leqslant n_{xy}}|x(u) - y(v)|)/n_{pairs}$, where n_{pairs}denotes the number of packet-pairs generated, such that sample $x(u)$ in each packet-pair is paired with a peer sample $y(v)$ that minimizes $|x(u) - y(v)|$ for all v. After computing t, the M_a can be computed for any of the two paths. In this computation, we select samples from the path sample set such that inter-packet spacing is higher than t. Samples that are not used in the auto correlation test (due to packet spacing violation) are marked and are not used in the computation of M_x (for each particular test).

We use this comparative method to determine whether both paths share a bottleneck or not. In case that both paths do not have shared bottleneck, i.e., $M_x \leqslant M_a$, it does not mean that both of them are independent, and they are still likely to have some shared links or congestion. The absolute cross-measure correlation value M_x can reflect their *weak* correlation degree approximately, and we can simply exploit this specific value to quantify this kind of *weak* correlation.

5. Multiple paths selection

The multipath selection procedure is firstly to classify paths as different groups according to the result of *comparison tests*; then choose the best paths from each group, and if necessary, carefully find a required number of paths as the final result of selection procedure.

5.1. Grouping process

Algorithm 1 Grouping Process

```
Input: All paths full set $S_n$
Output: A list of path groups denoted as $\Phi$
$\Phi$ = Grouping ($S_n$)
    g = 0, $\Phi = \emptyset$
    for (i = 1; i $\leqslant$ M; i++) do
        for (j = 1; j $\leqslant$ N; j++) do
            for (k = 1; k $\leqslant$ g & & g > 0; k++) do
                if (representative path $P_{xy} \in G_k$ exists) then
                    if ($\rho(P_{ij}, P_{xy}) > \rho(P^1_{ij}, P^2_{ij})$) then
                        put $P_{ij}$ into $G_k$
                        break
                    end if
                end if
            end for
            if (k > g) then
                g++
                put $P_{ij}$ into the new created $G_g$
                $\Phi \leftarrow \Phi \cup \{G_g\}$
            end if
        end for
    end for
return $\Phi$
```

The grouping process takes as input a set of target paths S_n (with sufficient samples) to be grouped. We number P_{ij} as k_{th} path with $k = (i - 1)N + j$. We group each path according to its order. First, the first path P_{11} is to be grouped, then the second path P_{12}, etc. To group a new path P_{ij}, we designate a *representative path* P_{xy} in every group which has an identical source or destination address with P_{ij}, i.e., $x = i$ or $y = j$. A new path is only compared to its *representative path* of the group to determine whether it should join the group or not. This ensures that all paths that are grouped together are highly correlated. However, if there is no *representative path* in the group, the new path is not joined to that group. Finally, if the new path cannot be joined to any existing group, a new group is created. The overall grouping process is illustrated in Algorithm 1.

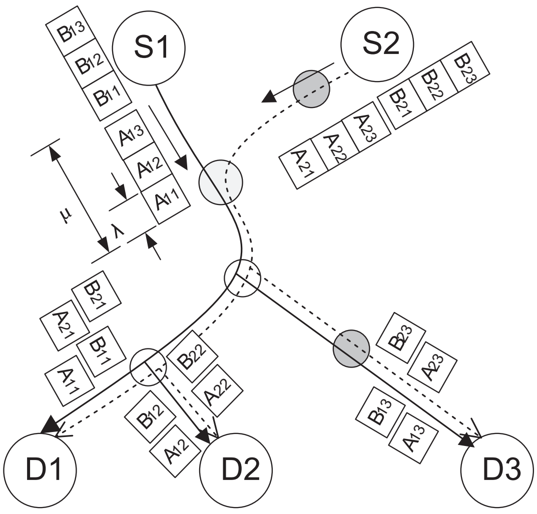


Fig. 3. Example of multi-source probing in one period.

5.2. Selection process

In the second step, it is necessary to find the best path [18] within each group firstly. The best path is the path which yields minimum expected transmission time for the requested data. Different from the correlation between paths used for multipath selection, the intrinsic performance of a path is more important for the single path selection within each group. The motivation behind this is to give preference to high-bandwidth and low-latency path, so the Bandwidth-Delay Product (BDP) is used as the selection metric here. This kind of selection, not restricted by the number of paths, upper layer application and so on, is called *free selection*.

Additionally, it may happen that the upper layer application or end system may impose specific requirements on the number of paths. In this case, we need to select the required number of paths as the selection output. To differentiate from *free selection*, this additional selection is called *restrained selection*. The candidate paths are the within-group optimal paths obtained in *free selection*. In fact, these paths have *weak* correlation between each other, and may be used for multipath transmission straight away to provide the maximum flow.

If the number of paths required (s) is greater than the number of candidate paths (k), i.e. $s \geqslant k$, more paths are needed to be selected as transmission paths. The actual strategy can depend on the specific scenario, and the final results can still use the output result of *free selection*, or append several other randomly selected paths. However, if $s < k$, further selection is needed to find fewer paths as required. These candidate paths do not have shared bottlenecks, but they are likely to share some ordinary congestion events, which still present a certain correlation. In *restrained selection*, we adopt the *cross-measure* value of $M_x = \rho(P_{ij},P_{xy})$ to quantify path correlation.

Here, the candidate path set is notated as S_k. Each path $P_{ij} \in S_k$ is associated with R non-negative and additive QoS values $W_r(P_{ij})$, $r = 1,2,\ldots,R$. There are R constraints D_r, $r = 1,2,\ldots,R$ for the combined capability over s paths. The correlation $\rho(P_{ij},P_{xy})$ between two paths P_{ij} and P_{xy} is defined in (1). Also we define the indication variable vector $X = (x_{ij})$ to indicate whether the path is selected ($x_{ij} = 1$), or not ($x_{ij} = 0$). This *minimum correlation selection* can be written as the following optimization problem with constraints:

$$\text{Minimize}: \quad \sum_{P_{ij}\in S_k} \sum_{P_{xy}\in S_k} x_{ij}\cdot x_{xy}\cdot \rho(P_{ij},P_{xy}) \tag{2}$$

$$\text{Subject to}: \quad \sum_{P_{ij}\in S_k} x_{ij} W_r(P_{ij}) \geqslant D_r (or \leqslant D_r) \quad \forall r = 1,2,\ldots,R, \tag{3}$$

$$\sum_{P_{ij}\in S_k} x_{ij} = s, \tag{4}$$

$$\text{over}: \quad x_{ij}, x_{xy} \in \{0,1\}, \quad \forall P_{ij}, P_{xy} \in S_k. \tag{5}$$

Here, (3) means R different QoS constraints, e.g., if W_r (P_{ij}) is the bandwidth value of path P_{ij}, then (3) specifies that "the sum of bandwidths for all selected paths should exceed the required value D_r". (4) and (5) are the constraints regarding the number of paths required. The solving approach is to enumerate all path pairs in any s paths from S_k that meet the constraint (3), sum their correlations, and choose the s paths with the minimum summation of correlations as the final result of *restrained selection*.

5.3. Computational complexity analysis

In order to avoid excessive probing cost while still measuring the correlation of any two paths, the first issue to be resolved is how to minimize the network load of the probing data. The foregoing sections describe a method of probing and collecting data at endpoints, to be used in calculating the *path correlations*. Assume that m denotes the number of path delay samples required for each R_{ij}. Considering the computation of M_a, each path needs $2m$ probing packets. For M source and N destination addresses, there are MN paths in total. In traditional probing schemes, to compute the M_a and M_x for any two paths, each path needs to transmit $2m$ probing packets and thus $4m$ probing packets are transmitted. Because there are $C^2_{M\cdot N}$ pairs of path, the total number of probing packets is $4mC^2_{M\cdot N}$. Clearly, this traditional approach does not scale well. In our proposed multi-source *N-packets-pair* probe mechanism, each source transmits m *N-packets-pair* probe sequence only once which is reused to compute the correlation degree with all other paths. As each *N-packets-pair probe* sequence has *2N* probing packets, the number of probing packets transmitted by each source is *2mN*. For M sources, the total amount of probing packets is *2mMN*, which is much fewer than that of traditional probe method.

The second issue is to reduce the computational complexity of the selection process. The brute force approach is to compare all possible pairs of paths to determine their correlation degrees, which is exponential in the number of paths MN. Our proposed GMS introduces the grouping process which uses the representative path from each group to make a comparison, instead of comparing with each path within the group. Thus, the computational complexity of GMS is $O(M\cdot N\cdot g)$, where g is the number of groups within a range of 1 to MN. The worst case occurs if all paths do not share any common bottlenecks and each is grouped separately, which would not occur often. This is due to the locality of address assignment, as well as traffic power-law characteristics.

6. Practical issues

In this section, we discuss some practical issues on the implementation framework and the necessity of adding a *multihoming sublayer* below the transport layer for the multipath selection.

6.1. Implementation framework

In order to deploy the proposed GMS in the Future Internet easily, we propose an implementation framework in the end-host system as shown in Fig. 4. The key elements are a decision point *Multipath Selection* and an aggregate point *Multipath State Management*, and other function modules responsible for reporting the information of different levels including the *Access Capability*, the end-to-end *Routing Capability*, the non-physical constraints *Policy* & *Preferences* of the operator(s) and user, the dynamic end-to-end path capacity through *Multipath Flow Control* and a series of *per-path Congestion Control* modules.

The basic flow of information is as follows:

- *Access Capability* detects a number of available accesses that can be used to support the application requirement. These available accesses are pre-filtered based on capabilities of the first hop, and potentially any information about local interface network capabilities.
- The set of remaining potential paths is then passed on to *Routing Capability*, which interacts with routing functionality to determine capabilities of the possible paths across the network between the source and the destination.
- *Multipath Flow Control* is responsible for managing the multiple paths uniformly, such as coordinated ARQ and load scheduling. The *per-path Congestion Controls* are responsible for limiting the transmission rate of each path by controlling its window size independently, and assessing the real-time capabilities of the end-to-end paths. All above functionalities may be implemented by the multipath transport protocol, such as

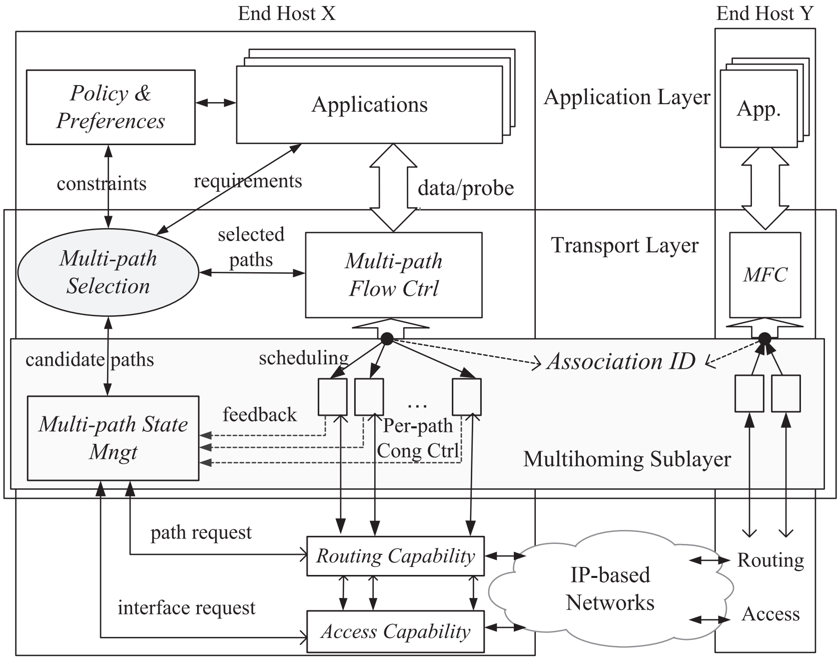


Fig. 4. Implementation framework for multipath selection.

cmpSCTP [23], Multipath TCP [6,24] and MRTP [25]. The feedback data (e.g. throughput, latency and packet loss rate) can be collected from the current traffic by most of transport control protocols. We can use the feedback information to optimize the multipath selection.

- When an application is initiated, it sends the application requirements and *Policy & Preferences* information to *Multipath Selection*. This decision element sends a request for paths towards the given destination to *Multipath State Management* and receives a list of candidate paths with associated capabilities and their correlations, and then combines with the information to select the most appropriate multiple paths for that application.

6.2. Multihoming sublayer

To allow the exchange of *multihoming* information between two hosts, it is necessary to define a novel *multihoming sublayer* as illustrated in Fig. 4. In contrast with current multihoming at the network layer (like in HIP [3], SHIM6 [5], SIMPLER [26]), this *multihoming sublayer* is located below the original transport layer. Here, we borrow the concept of "association" from SCTP [4,8], which is a generalization of a TCP connection. The unique association identifier is visible above the *multihoming sublayer*, which ensures that all upper layer protocols can operate unmodified in a multihoming environment even if some IP paths and wireless interfaces are changed. The mapping between the single association identifier and actual (several) address pair(s) is done by the new *multihoming sublayer*.

Normally, routing decisions and the selection of access networks are based on the information from the IP/network layer. However, this information is inadequate in multipath selection, since we need to take into account more factors such as the end-to-end path's throughput and latency. As a rule, the information about above characteristics is visible to the transport layer, but is typically hidden from the IP/network layer. Besides, how to select necessary paths is precisely what transport layers care about most, and the multipath selection can be thought of as the extension of the underlying single-path routing mechanism in the multihoming environment. Obviously, this layer division can minimize the related information (e.g. routes and specific policies) to be shared between different layers.

7. Evaluation and numerical results

In this section, we evaluate the effectiveness of GMS and quantify its gain in terms of the aggregating throughput with the number of paths. In addition, we also compare GMS with several strategies related to path selection.

7.1. Simulation configuration

The simulation platform used is the OPNET simulator [27]. The proposed GMS is implemented based on the proposed cmpSCTP [23] protocol and multipath scheduling strategy [28] in our previous work. Three topologies are used in our experiments to imitate complicated networks. The topologies provide a simplified connection of the physical routes which only contains some routers playing the role in branching or joining the paths of network flow. The source is provided with 2 or 3 addresses and the destination with 3 or 4 addresses, so there are 6 or 12 parallel paths between the source and destination host. 6 or 12 concurrent TCP-like flows are generated as foreground traffic, accompanied by the same number of multiplexed Pareto flows generated as background traffic. Our cross-traffic generator is a combination of 20 Pareto sources with an on–off period that takes values in the range [10 ms, 1 s]. Each simulation runs for 20 s where probe flows and background flows start at 0 s and cross-traffic flows start at 3 s. The path's share of the bottlenecks' bandwidth is affected by cross-traffic. The capacity and propagation delay of each link are indicated in the following figures. Table 1 summarizes the simulation parameters.

In the first topology (*Marginal model*), Fig. 5, the shared bottlenecks occur in the margin of the network. We generate cross traffic with bandwidth 8 Mbps between R1 and R6 to produce the first shared bottleneck SB1. The second shared bottleneck SB2 occurs in R5 due to the minimal capacity of 3 Mbs but shared by P_{13} and P_{23}. In the second topology (*Central model*), Fig. 6, the shared bottlenecks occur in the centre of the network. The centre shared links have limited bandwidth, while the link on the P_{12}, P_{12}, P_{21} and P_{22} are congested by high cross-traffic load. These two topologies simulate the environment where the shared bottlenecks occur either in the edge router, or in the common core router.

Table 1
Simulation parameters.

TCP flows	6 or 12 infinite FTP flows
Cross traffic	1 or 2 flows, CBR (8 Mbps and 6 Mbps)
Background traffic	to all links (1 Mbps Pareto)
Queue size	250 packets
Drop policy	Drop-tail
Mean size of packet	500 bytes

Fig. 7 depicts the third simulation topology (*Universal model*), where the shared bottlenecks can occur in any location of the network including the edge router and the core router. This topology is not as symmetric as the first two, which includes more paths and more complicated path relationships. The 12 paths produce 5 bottlenecks, which involve 4 shared bottlenecks and 1 unshared bottleneck. In these bottlenecks, the SB1 and SB2 are congested by high cross-traffic load, SB3 and SB4 are congested by limited bandwidth, and the only unshared bottleneck between R3 and R7 is uniquely possessed by the P_{34}.

In the simulation results presented next, SPS denotes the original single best path selection scheme [18]; RMS denotes the random multiple better paths selection scheme without the consideration of path correlation, which selects paths according to the priority of Bandwidth-Delay Product; GMS-Free denotes the *free selection* scheme of GMS algorithm; GMS-Restrained denotes the *restrained selection* scheme, which is subject to some restrictions and here only considers the number of paths required.

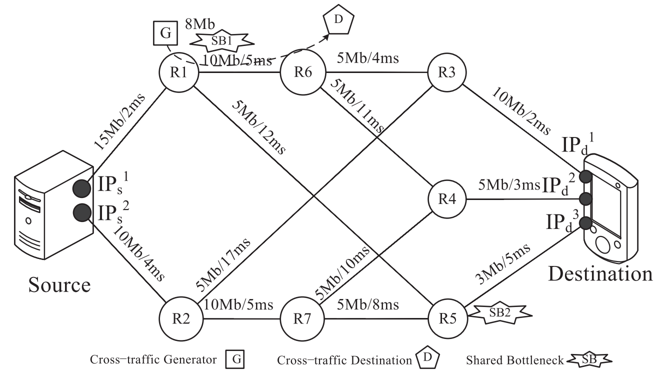


Fig. 5. Marginal model simulation configuration.

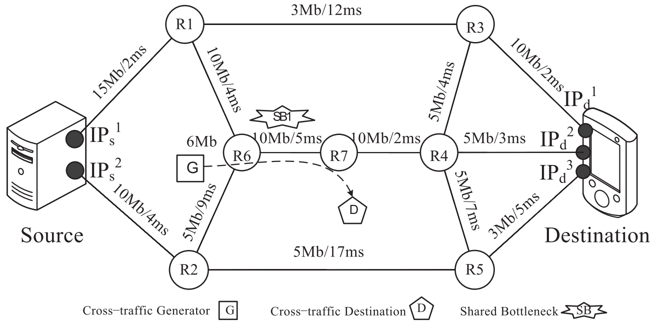


Fig. 6. Central model simulation configuration.

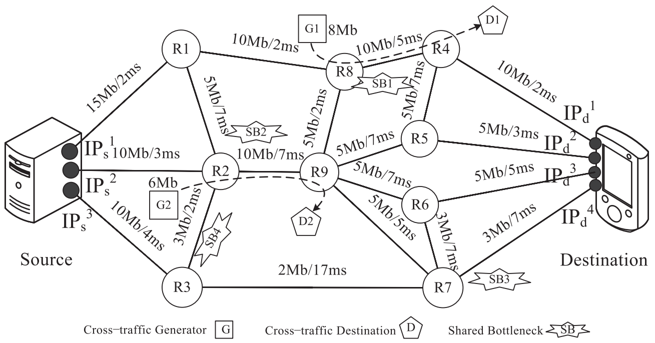


Fig. 7. Universal model simulation configuration.

7.2. Aggregating throughput

In this experiment, we evaluate the effectiveness of GMS-Free in terms of the aggregating throughput in all three topologies (Figs. 5–7). We use FTP applications as our foreground traffic, along with probing packets through all available paths, and trigger multi-path selecting at time 9 s. After the multiple paths are selected, the sender transmits the data over them.

Fig. 8 shows the aggregating throughput over simulation time. During the first 9 s, all 12 paths are used to send data, which includes several unnecessary paths. Thus, a large amount of congestion occurs and the total aggregating throughput is lower. With the use of multipath selection, less number of paths is used and the total aggregating throughput is increased. This is because these selected paths are nearly independent with each other and they just have sufficient bandwidth to transmit the data.

7.3. Number of paths required

Following above simulation topologies, we now evaluate the effectiveness of GMS-Restrained in terms of the aggregating throughput as the number of paths required changes. In this experiment, in case that the number of paths required is greater than the number of groups, more paths with better performance are appended to the output of GMS-Free scheme. In Fig. 9, we vary the number of paths required (s) and plot the aggregating throughput achieved. The ideal aggregating throughput is the sum of the throughput achieved for each selected path, which should increase as the number of selected paths increases.

We observe in Fig. 9 that the actual aggregating throughput curve does not monotonically increase, but begins to decline from a point. For instance, in the first topology the aggregate curve gradually increase in the beginning, but as s increases beyond 4, the aggregate curve is on the decline. For this topology, there are only 4 groups, meaning that selecting paths beyond the number of groups is useless. Therefore, it's necessary to make sure that the number of selected paths is no more than 4 in this case. If not so, the congestion will occur and the network bandwidth will be wasted unnecessarily. The right value of s depends on the number of groups (k). We suggest that the number of paths required be smaller than the number of groups, i.e. $s \leqslant$ k.

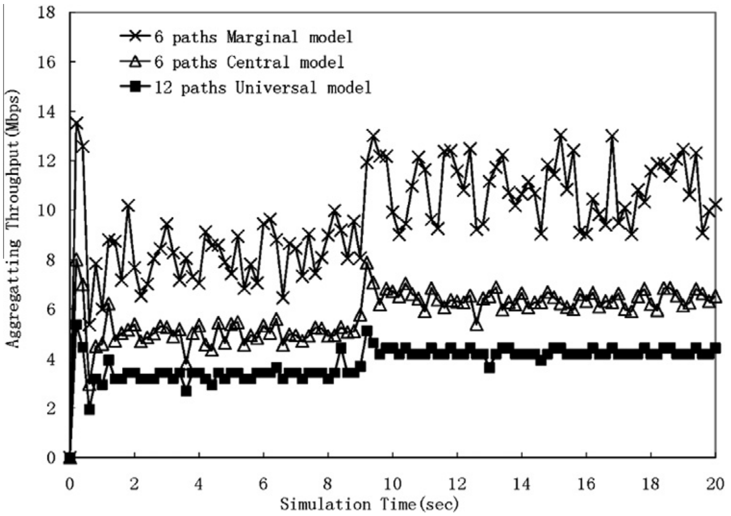


Fig. 8. Aggregating throughput (over simulation time) with various network environments.

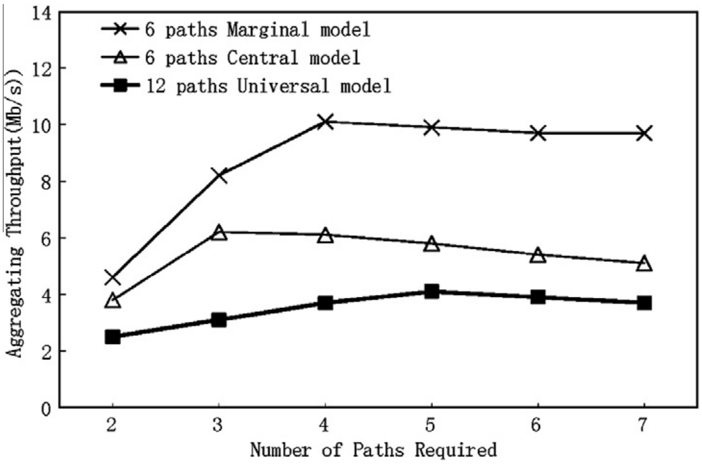


Fig. 9. Effect of the number of paths required.

7.4. Comparison with selection schemes

In this experiment, we compare the performance of GMS and other schemes in terms of the aggregating throughput. For GMS-Restrained in all three topologies, the number of paths required (s) is set to 3. For instance the first topology, one possible selected path sets are $\{P_{21}\}$ of SPS, $\{P_{12},P_{13},P_{23}\}$ of RMS, $\{P_{11},P_{13},P_{21}\}$ of GMS-Restrained, and $\{P_{11},P_{13},P_{21},P_{22}\}$ of GMS-Free.

The simulation results for different selection schemes, shown in Fig. 10, demonstrate that the aggregating throughput of GMS-Free is the highest, which is attributed to its exploitation of path correlation. Moreover, comparing the curves for RMS and GMS-Restrained, we observe that the aggregating throughput of RMS is significantly lower than that of GMS-Restrained, although both of them have the same number of selected paths. This is because RMS is a *correlation blind* scheme that cannot exploit the path diversity. In fact, we observe that the aggregating throughput of RMS is even worse than that of the single path transmission, as more congestion events occur in the network for RMS.

8. Discussion and open issues

In the previous sections, we have outlined our ongoing research on selecting multiple paths for CMT by leveraging the path diversity between two multihomed nodes. We have proposed a probing scheme and a subsequent GMS strategy. However, there are still a large number of open issues related to multipath selection. In this section we briefly discuss some of them.

8.1. Online re-selecting for time-varying network

In our current solution, multiple paths are selected before multipath transmission. But the network traffic is time-varying, and the actual route of each path is also

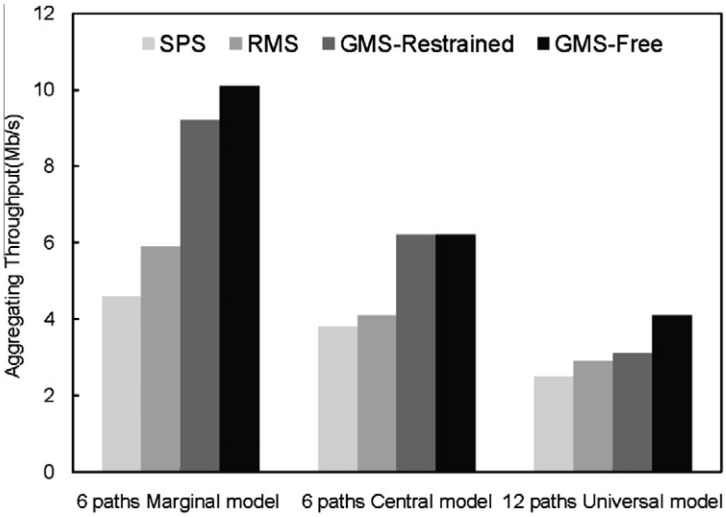


Fig. 10. Effect of different selection schemes.

volatile. So it is essential to monitor the change of the correlation degree between the paths, and update the selected path set according to the current network conditions. In order to achieve this, more research is needed to reduce the cost of probing correlation without sacrificing the accuracy of correlation computation. One option is to relieve the burden of sending probing packets, and replace the current active probing with the passive probing. However, how to obtain the inherent characteristics of multiple paths and the correlation with each other directly from the existing traffic is still an open issue.

8.2. Coordinated congestion control

One future work involves applying the concept of "*path correlation*" to multipath congestion control. The result of our grouping scheme can be used by any coordinated congestion control scheme. Most of congestion control schemes are triggered by packet loss. The packet losses occurred in independent paths are considered as relatively independent, so the congestion control just needs an independent strategy. For the packet loss occurring in correlated paths, however, the congestion control requires a coordination mechanism [29,30]. We design a simple coordination mechanism that works as follows. Each path maintains its own congestion window. When the loss is detected at any path of a group, all paths within the group reduce their window sizes to react to incipient congestion, but the paths belonging to different groups normally increases their window sizes to rapidly explore available network bandwidth. Additional future work is to study the impact of the *delay difference* [28] between paths on the multipath congestion control.

8.3. Supporting multipath routing

In our current solution, we assume that the underlying routing protocol only supports single-path routing. In the latest multipath routing technologies, including multipath routing in IP layer [10–12] and multipath routing in overlay layer [13–15], a pair of S–D addresses (called a *path* here) can be assigned multiple IP/overlay routes. It may happen that there is more than one bottleneck point between a pair of S–D addresses, so both our correlation model and probing scheme are invalid. As the situation of shared bottlenecks between two *path*s becomes more complex, it is necessary to redefine the path correlation and find a way to calculate it in order to support multipath routing. In addition, another open issue is to study the coordination between multipath selection and multipath routing in the future.

9. Conclusions

Compared to uni-path transmission, multipath transmission for the Future Internet can better utilize network resources and enhance aggregating throughput. More sophisticated network deployment means that there may be some topologically shared or joint links between different transport paths. Thus, how to select these paths for a given service, rather than interfaces or routes, seems to be a very interesting area of research. The selection should be unaware of the knowledge of the underlying network topology and the information should only be obtained through end-to-end network measurement.

In this paper, we propose a multipath selection strategy to exploit the path diversity by taking into account the potential path correlation. The probing and grouping mechanism enables the subsequent selection to avoid underlying shared bottlenecks easily. Besides, we propose an implementation framework in the end-host system, and discuss the necessity of adding a *multihoming sublayer*. The extensive simulations demonstrate that our scheme can select a desirable path set for CMT in various network environments, and outperforms other selection schemes in terms of the aggregating throughput.

Acknowledgments

This work was jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) National 973 Program (No. 2007CB307100, 2007CB307103); (3) National Natural Science Foundation of China (No. 61072057, 60902051); (4) the Fundamental Research Funds for the Central Universities(BUPT2009RC0505); (5) National Key Science & Technology Specific Project of China (No. 2011ZX03002-001-01, 2011ZX03002-002-01).

References

[1] <http://www.future-internet.eu/>.
[2] A. Feldmann, Internet clean-slate design: what and why, ACM SIGCOMM Computer Communication Review 37 (3) (2007).
[3] R. Moskowitz, P. Nikander, Host Identity Protocol (HIP) Architecture, RFC 4423, May 2006.
[4] R. Stewart, Stream Control Transmission Protocol, RFC 4960, September 2007.
[5] E. Nordmark, M. Bagnulo, Shim6: Level 3 Multihoming Shim Protocol for IPv6, RFC 5533, June 2009.
[6] A. Ford, C. Raiciu, TCP Extensions for Multipath Operation with Multiple Addresses, IETF Internet-Draft, draft-ford-mptcpmultiaddressed-03, 2010.
[7] P. Smith, BGP multihoming techniques, NANOG 23, October 2001.
[8] J.R. Iyengar, P. Amer, R. Stewart, Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths, IEEE/ACM Transactions on Networking 14 (5) (2006) 951–964.
[9] J. Apostolopoulos, M. Trott, Path diversity for enhanced media streaming, IEEE Communications Magazine Special Issue Proxy Support Streaming Internet 42 (8) (2004) 80–87.
[10] P. Papadimitratos, Z.J. Haas, E.G. Sirer, Path set selection in mobile Ad Hoc networks, in: Proceedings of ACM MobiHoc, Lausanne, Switzerland, June 2002.
[11] S. Mao, Y. Hou, X. Cheng, H. Sherali, S. Midkiff. Multipath routing for multiple description video in wireless Ad Hoc networks, in: Proceeding of IEEE INFOCOM, Miami, FL, March 2005.
[12] W. Wei, A. Zakhor, Path selection for multi-path streaming in wireless ad-hoc networks, in: Proceedings of IEEE ICNP, January 2006.
[13] A. Begen, Y. Altunbasak, O. Ergun, Multi-path selection for multiple description encoded video streaming, in: Proceedings of IEEE ICC, May 2003.
[14] J. Chen, S. Chan, V. Li, Multi-path routing for video delivery over bandwidth-limited networks, IEEE Journal of Selected Areas Communications Special Issue Design, Implement. Anals of Communication Protocols 22 (10) (2002) 1920–1932.
[15] Z. Ma, H. Shao, C. Shen, A new multi-path selection scheme for video streaming on overlay networks, in: Proceedings of IEEE ICC, June 2003.

[16] C. Casetti, C. Chiasserini, R. Fracchia, M. Meo, Autonomic interface selection for mobile wireless users, IEEE Transactions on Vehicular Technology 57 (6) (2008) 3666–3678.

[17] M. Alkhawlani, A. Ayesh, Access network selection based on fuzzy logic and genetic algorithms, Advanced Artificial Intelligence (AAI) 8 (1) (2008) 1–12.

[18] R. Fracchia, C. Casetti, C. Chiasserini, M. Meo, WiSE: best-path selection in wireless multihoming environments, IEEE Transactions on Mobile Computing 6 (10) (2007) 1130–1141.

[19] D. Rubenstein, J. Kurose, D. Towsley, Detecting shared congestion of flows via end-to-end measurement, IEEE/ACM Transactions on Networking 10 (3) (2002) 381–395.

[20] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman Company, 1979.

[21] Y. Zhang, N. Duffield, V. Paxson, S. Shenker, On the constancy of internet path properties, in: Proceedings of ACM SIGCOMM Internet Measurement Workshop, November 2001.

[22] S. Keshav, A control-theoretic approach to flow control, in: Proceedings of ACM SIGCOMM, Zurich, Switzerland, September 1991.

[23] J. Liao, J. Wang, X. Zhu, A multi-path mechanism for reliable VoIP transmission over wireless networks, Computer Networks 52 (13) (2008) 2450–2460.

[24] A. Kostopoulos, H. Warma, T. Levä, B. Heinrich, A. Ford, L. Eggert, Towards multipath TCP adoption: challenges and opportunities, in: Proceedings of Sixth Euro-NF Conference on Next Generation Internet, Paris, France, June 2–4, 2010.

[25] S. Mao, D. Bushmitch, S. Narayanan, S.S. Panwar, MRTP: A multiflow real-time transport protocol for ad hoc networks, IEEE Transactions on Multimedia 8 (2) (2006) 356–369.

[26] R. Gummadi, R. Govindan, Practical routing-layer support for scalable multihoming, in: Proceedings of IEEE INFOCOM, March 2005.

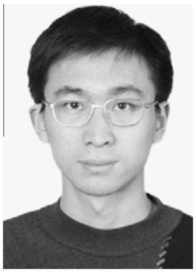
[27] OPNET simulator URL:<http://www.opnet.com/>.

[28] J. Wang, J. Liao, X. Zhu, On preventing unnecessary retransmission with optimal fragmentation strategy, in: Proceeding of IEEE ICC, Beijing China, May 2008.

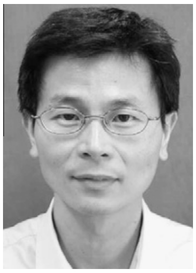
[29] P. Key, L. Massoulié, D. Towsley, Path selection and multipath congestion control, in: Proceedings of IEEE INFOCOM, Anchorage, AL, USA, 2007, pp. 143–151.

[30] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, H. Tokuda, Multipath congestion control for shared bottleneck, in: Proceedings of Seventh International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT), Tokyo, Japan, May 21–22, 2009.

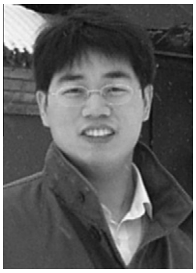
Jianxin Liao was born in 1965, obtained his Ph.D. degree at University of Electronics Science and Technology of China in 1996. He is presently a professor of Beijing University of Posts and Telecommunications. He has published hundreds of papers in different journals and conferences. His research interests are mobile intelligent network, broadband intelligent network and 3G core networks.



Jingyu Wang was born in 1978, obtained his Ph.D. degree from Beijing University of Posts and Telecommunications in 2008. Now he is an assistant professor in Beijing University of Posts and Telecommunications, China. His research interests span broad aspects of performance evaluation for Internet and overlay network, traffic engineering, image/video coding, multimedia communication over wireless network.



Tonghong Li was born in 1968, obtained his Ph.D. degree from Beijing University of Posts and Telecommunications in 1999. He is currently an assistant professor with the department of computer science, Technical University of Madrid, Spain. His main research interests include resource management, distributed system, middleware, wireless networks, and sensor networks.



Xiaomin Zhu was born in 1974, obtained his Ph.D. degree from Beijing University of Posts and Telecommunications in 2001. Now he is an associate professor in Beijing University of Posts and Telecommunications. His major is Telecommunications and Information Systems. His research interests span the area of intelligent networks and next-generation networks with a focus on 3G core network and protocol conversion. He has published over 110 papers, among which there are more than 30 first-authored ones, in different journals and conferences.