Topology-aware Kademlia based on Distributed Clustering in Self-organizing Mode

Qiang Xu
Dept. of Network Engineering
Electronic Engineering Institute
Hefei, China
e-mail: yfnm126@126.com

Lechang Sun
Dept. of Network Engineering
Electronic Engineering Institute
Hefei, China
e-mail: sunlechangeei@yahoo.com

Jingju Liu
Dept. of Network Engineering
Electronic Engineering Institute
Hefei, China
e-mail: jingjul@yahoo.com

***Abstract* — Targeting the detouring problem led by the mismatch between logical topology and physical topology in structured peer-to-peer (P2P) networks, a novel topology-aware Kademlia is proposed, in which nodes are classified into self-organized clusters using distributed algorithm. A novel clustering-based mechanism is designed to correlate two topologies, so that the routing procedure is rationalized hop by hop. Theoretical analysis proves that this advanced Kademlia is also O(logN) running but more efficient than the original in physical topology. Simulation results show its outstanding performance on minimizing latency and improving stretch by nearly 15% through avoiding the mismatch.**

Key words-DHT; Kademlia; topology-aware; mismatch; detouring;

I. INTRODUCTION

The structured P2P has recently emerged as a candidate infrastructure for building large-scale and robust network applications. The core component of the structured P2P is the distributed hash table (DHT). Many classical DHT networks have been proposed in [1-5], among which Kademlia[4] is widely utilized in practical system for its simple mechanism and low maintenance overhead.

Since DHT networks create a virtual topology over the physical topology, the only relation between two layers exists in the Hash algorithm, which makes a node's logical ID independent of its physical location. Consequently, the routing algorithm decreasing hops in logical topology can't shrink physical latency. This mismatch between the two topologies results in a serious problem known as detouring. To solve this problem, a topology-aware Kademlia base on distributed clustering in self-organizing mode is proposed in this paper. First, a distributed clustering algorithm is presented to classify all the nodes into self-organized clusters according to their physical proximity. Then, a NodeID (NodeID is the identifier to mark a node) assignment mechanism based on the constructed clusters is designed to correlate two topologies. Both theoretical analysis and simulation results prove that the improved structure can boost the efficiency of system remarkably by rationalizing the routing procedure.

The remaining sections of this paper are organized as follows. After a survey of related studies in Section 2, a clustering algorithm in self-organizing mode is proposed in Section 3. Section 4 details the topology-aware Kademlia. Simulation results and performance analysis are presented in Section 5. Finally, Section 6 concisely summarizes this paper.

II. RELATED WORK

Topology-aware techniques for P2P have been extensively studied in recent years. Three main approaches are widely used to construct topology-aware structured P2P overlay: geographic layout, proximity routing and proximity neighbor selection [6].

Geographic layout is to reflect physical location of a node by the value of its NodeID. As NodeID assignment mechanism is always determined by the architecture itself, the geographic layout method is always conceived for a certain P2P protocol. [7] presents an effective implementation in CAN [1] with an achievement of a less delay stretch. In addition, [8] proposes a method of getting an appropriate NodeID by considering the nodes' physical position in Chord [2].

Proximity routing approach attempts to select a relatively near node from a set of candidates as the next hop of routing. Following this approach, clustering is usually considered to produce the set of candidates. In [9-11], overlay networks are constructed on centralized clusters in which supernodes incline to be bottleneck of system. Although some distributed clustering algorithms are proposed in [12-14], there is still much room to improve the efficiency and reduce the expense.

Proximity neighbor selection is implemented by means of establishing and maintaining a routing table comprising proximity neighbors. The classical example is Pastry [5], as well as researchers try to apply this approach to other DHTs in [15-16].

As far as special protocols are concerned, some new methods are proposed for IPv6 or Ad-hoc in [17-22]. However, they are not suited to current Internet circumstance.

Both advantages and drawbacks exist in three approaches mentioned above. This paper concentrates on combining and highlighting their advantages to investigate a methodology for Kademlia.

III. DISTRIBUTED CLUSTERING ALGORITHM

Traditional clustering relies on the entire knowledge of topology, which is impossible to acquire for P2P system due to its dynamic. Therefore, a distributed algorithm is proposed to classify nodes, according to their physical locations, into self-organized clusters with properties of certainty and symmetry.

978-1-4244-6349-7/10/$26.00 ©2010 IEEE

A. Basic Definitions

Before illustrating the clustering algorithm, four basic definitions are introduced to show the method on the physical topology-aware overlay construction and the cluster identification.

Definition 1) The ***reference frame of physical topology*** is defined as n different landmarks [10] in Internet; each landmark stands for one dimension of physical topology. In such reference frame, a node locates itself according to the Round Trip Time (RTT) by sending ICMP echo message (Ping) to each landmark. Out of consideration for accuracy, the landmarks are supposed to be distributed uniformly in Internet.

Definition 2) The ***landmark permutation*** indicates a node's location in physical topology. Before joining the system, a new node N_i measures the RTT to all landmarks and permutated them following the Shortest Latency First principle to construct $Permutation_i$.

Definition 3) The ***cluster identifier*** ClusterID labels a cluster uniquely. In a reference frame with n landmarks, the length of ClusterID is $n\lceil \log_2 n \rceil$ bits. For a certain node $N_i \in Cluster_i$, $ClusterID_i$ equals to $Permutation_i$, when landmarks are interpreted in binary. The relation $<ClusterID_i, NodeID_i>$ is published in network as resource .

Definition 4) The ***resolution of reference frame*** indicates how many different clusters can be discriminated in the reference frame. Since the cluster and the landmark permutation are in one-one correspondence, n! is defined as the resolution of a reference frame with n landmarks.

B. Clustering Algorithm

Supposing $N_i \in Cluster_i$ whose cluster identifier is $ClusterID_i$, Algorithm Ⅰ details the clustering algorithm.

According to Algorithm Ⅰ, the reference frame divides the physical topology into n! independent regions and nodes fallen into the same region form a cluster. In addition, these clusters have two favorable properties as follows for designing routing algorithm.

Property 1) ***Certainty***: for a node N_i, there must be a unique cluster which N_i belongs to.

Proof sketch: According to steps 6-7 in Algorithm Ⅰ, $ClusterID_i$ equals to $Permutation_i$ denoted in binary code, besides the existence and unicity of $Permutation_i$, so Property 1is proved.

Property 2) ***Symmetry***: if $N_j \in Cluster_i$ is true，then $N_i \in Cluster_j$ holds.

Proof sketch: According to steps 13-15 in Algorithm Ⅰ, N_i exchange contact information with any $N_j \in Cluster_i$ and N_j combine $\{N_i\}$ into $Cluster_j$ afterwards. When the network is steady, if $N_j \in Cluster_i$, then $N_i \in Cluster_j$ holds.

IV. TOPOLOGY-AWARE KADEMLIA

In previous studies [12-17], nodes in the same cluster are selected to be the next hop to improve routing efficiency. However, they omit the way how to choose a node in different clusters. Hence, a novel definition of distance between clusters is introduced in this section, along with a NodeID assignment mechanism correlating physical topology with logical topology in Kademlia.

A. Basic Scheme

Unconventionally, each landmark has different capability of locating a certain node in the reference frame defined—the closer nodes are more capable than the further ones. If W_i^{α} weighs such capability of the αth nearest landmark in $Permutation_i$, W_i^{α} is in reverse ratio to α . Thus, for N_i and N_j, the ordinal of inconsistent landmarks between $Permutation_i$ and $Permutation_j$ can be used to measure their distance in physical topology.

Definition 5) The ***mapping distance*** $Dis(ClusterID_i, ClusterID_j)$ is defined as the group-wise exclusive or (XOR, $\otimes$) result between $ClusterID_i$ and $ClusterID_j$. It is calculated from equation (1), in which *group*() operation means dividing the cluster identifier into segments of $\lceil \log_2 n \rceil$ bits in a reference frame with n landmarks,

$$Dis(ClusterID_i, ClusterID_j) = group(ClusterID_i) \otimes group(ClusterID_j) \quad (1)$$

Revealed by step 7 in Algorithm Ⅰ, all nodes in the same cluster have an identical landmark permutation, so that they are close enough to ignore their tiny distance. Therefore, the definition above just depicts the distance between clusters rather than nodes. For instance, if $ClusterID_i$ = 100101011000001010 and $ClusterID_j$ = 100101011010001000, then *group*($ClusterID_i$) = 100|101|011|000|001|010 and *group*($ClusterID_j$) = 100|101|011|010|001|000 .So the mapping distance between $Cluster_i$ and $Cluster_j$ is 000101 in binary or 5 in integer.

Algorithm I. **procedure Clustering()**

Require: N_i has a contact to an already participated node N_k ($k \neq i$).
$L=\{ landmark_j | 1 \leqslant j \leqslant n \}$ is the set of landmarks interpreted in binary.

1: $Cluster_i = \{ N_i \}$;
2: **for** every $landmark_k \in L$
3: Test the RTT to $landmark_k$ by Ping;
4: **endfor**
5: Construct $Permutation_i$
6: $ClusterID_i = Permutation_i$;
7: J=FindResource(N_k, $ClusterID_i$);
8: **for** every $N_j \in J$
9: **if** N_j is reachable
10: $Cluster_i = Cluster_i \cup \{ N_j \}$;
11: **endif**
12: **endfor**
13: **for** every $N_j \in Cluster_i$
14: Exchange its contact information with N_j
15: **endfor**
16: Publish $< ClusterID_i , NodeID_i >$ as resource;
17: **endprocedure**

Original Kademlia implements the proximity neighbor selection in logical topology, nevertheless the detouring problem still remains as a result of the mismatch of two topologies. The cluster identifier will be integrated into NodeID to handle it in this paper.

Algorithm II. **Function Routing()**

Require: N_i is looking up the target N_j. i is initialized to 0.

```
1: distance = NodeID_i ⊗ NodeID_j ;
2: k'=⌊log2 distence⌋ ;
3: while i < K
4: if the contact information of N_j is in k'th bucket
5: break;
6: else
7: NodeSet = α closet nodes without being requested;
8: for every N_m ∈ NodeSet
9: FIND_NODE(N_m, NodeID_j);
10: if Receive(N_m)=true
11: i++;
12: Update the routing table;
13: endif
14: endfor
15: endif
16: endwhile
17: if i ≤ K
18: return the contact information of N_j;
19: else
20: return 0;
21: endif
22: end function
```

B. *Logical Topology*

The logical topology of original Kademlia is an Incomplete Binary Tree, where nodes are determined as leaves by unique prefix of 160 bit hash quantities. The notion of distance is defined to be bit-wise XOR between NodeIDs.

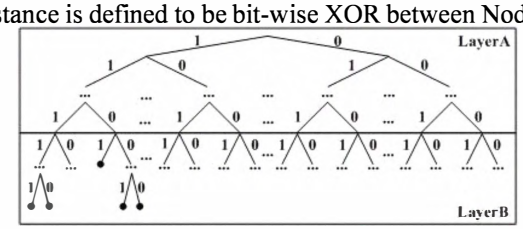

Figure 1. Logical topology of the proposed Kademlia

Fig. 1 illustrates the logical topology of the proposed Kademlia which is still an Incomplete Binary Tree but comprises two layers.

- LayerA constructs $n\lceil \log_2 n \rceil$ bit cluster space, in which the proximity relation of subtrees reflects the mapping distance between corresponding clusters.
- LayerB categorizes nodes belonging to the same cluster into an identical subtree where the representing leaves are determined by unique prefix of 128 bit hash quantities.

NodeID is constituted of two parts in the proposed structure: the $n\lceil \log_2 n \rceil$ bit cluster identifier and the 128 bit hash quantities.

C. *Routing Algorithm*

The routing table is an improvement to traditional *k*-buckets [4] in the proposed Kademlia. Every node stores a list of <*NodeID, IP address, Port, Dis, Time*> records for neighbors whose XOR distance fall into the range between 2^i and 2^{i+1}. If $N_i \in Cluster_i$ and $N_j \in Cluster_j$, *Dis* stands for the mapping distance between $Cluster_i$ and $Cluster_j$. In each *k*-bucket, records are sorted by *Dis*—the nearest node at the head, the farthest node at the end; thus the nearer neighbors have the priority to be requested. *Time* registers the latest contact time of neighbor, which is the basis to update the routing table. The farthest XOR distance between nodes is $2^{n\lceil \log_2 n \rceil+128}$ and there are $n\lceil \log_2 n \rceil+128$ *k*-buckets totally, since NodeID is a $n\lceil \log_2 n \rceil+128$ bit quantities.

The original Kademlia defines four RPCs: PING, STORE, FIND_NODE and FIND_VALUE. In the proposed structure, they are also adopted in routing algorithm which is detailed in Algorithm II.

D. *Theoretical Analysis*

The comparisons of two structures are listed in Tab. 1 to demonstrate the effective function of the proposed Kademlia. N is the total number of nodes in P2P system. $\overline{T}$ represents the average latency between nodes in the original structure, whereas $\overline{T}_1$ and $\overline{T}_2$ represent respectively the average latency between nodes within the same cluster and among different clusters in the proposed structure. Evidently, $\overline{T}_1$ is less than $\overline{T}_2$. The statistics in table is proved by three assertions.

TABLE I. Comparisons of Two Structures

Structure	Capacity	Exp. of Hop	Exp. of Latency
Original	2^{160}	$O(\log N)$	$\overline{T}O(\log N)$
Proposed	$n!*2^{128}$	$O(\log N)$	$(p\overline{T}_1+(1-p)\overline{T}_2)O(\log N)$

Assertion 1) The capacity of proposed Kademlia built on n landmarks is $n!*2^{128}$.

Proof sketch: n! is the resolution of the reference frame with n landmarks. Moreover, there can be 2^{128} nodes in a single cluster at most. Thus the capacity of network built on n landmarks is $n!*2^{128}$.

Assertion 2) The proposed Kademlia is as efficient as the original in logical topology.

Proof sketch: The difference between Algorithm II and lookup [4] procedure in original Kademlia is in the way how to select the next hop. However, they share the common routing mechanism. Therefore, they are equally efficient in logical topology as $O(\log N)$ running system proved in [4].

Assertion 3) The proposed Kademlia is more efficient than the original in physical topology.

Proof sketch: Define the probability that FIND_NODE arises within the same cluster to be p. In this case, the expectation of latency in proposed structure is $(p\overline{T}_1+(1-p)\overline{T}_2)O(\log N)$. On the other hand, the latency in original structure is $\overline{T}O(\log N)$. Because $\overline{T}_1 < \overline{T}_2 \le \overline{T}$ is true, $((p\overline{T}_1+(1-p)\overline{T}_2)O(\log N))<(\overline{T}O(\log N))$ always holds. In conclusion, the proposed Kademlia has more excellent physical efficiency than the original.

V. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

The simulation is carried out on P2PSim [23] platform, in which a new protocol named as A-Kademlia realizes the proposed structure, derived from the existed Kademlia protocol. King data set [24] and E2E Graph are used as the topology data and the topology model respectively. RedHat9.0 is the operating system used. All of the data are collected from output files by programs.

To evaluate its performance, A-Kademlia is compared with Chord, Tapestry and Kademlia in four metrics of request: logical hops, physical latency, latency per hop (stretch) and success rate. The total number of simulated nodes in platform is tuned to be 1K. As nodes join the network gradually, the four kinds of data are recorded. The average values are computed when the percentage of nodes reach to 10%, 50% and 100%, while the mean of these values is calculated at last.

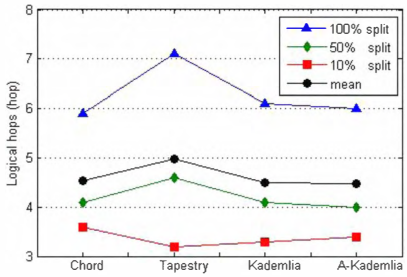

Figure 2. Percentage contrast in logical hops

Fig. 2 shows the percentage contrast in logical hops among four protocols. Notice that request in Tapestry undergoes the longest path in logical topology. Furthermore, the other three protocols are almost equal in this aspect of performance, which is consistent with Assertion 2).

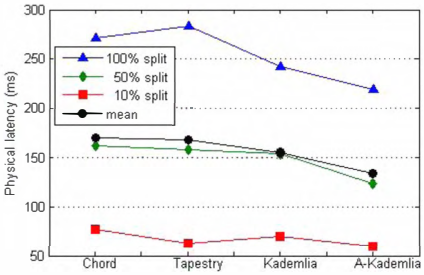

Figure 3. Percentage contrast in physical latency

Fig. 3 shows the percentage contrast in physical latency among four protocols. It is observed that A-Kademlia performs best. Comparing with Kademlia, A-Kademlia has shortened the latency by nearly 15%. This result accords with what is proved in Assertion 3).

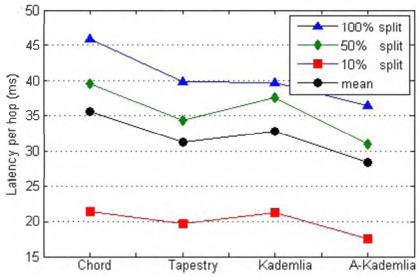

Figure 4. Percentage contrast in latency per hop

Fig. 4 shows percentage contrast in latency per hop among four protocols. The best result is obtained in A-Kademlia. It is indicated that proposed method can remarkably improve the rationality of each hop in routing procedure.

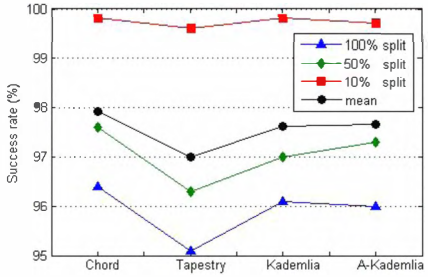

Figure 5. Percentage contrast in success rate

Fig. 5 shows the percentage contrast in success rate among four protocols. Evidently, Tapestry has the lowest rate, while the other three are comparable with one another.

The simulation results shows that Chord has the most serious detouring problem owing to its unidirectional clockwise routing along ring. Moreover, Tapestry has the poorest scalability. Its performance decays at the highest speed, while the node number increases. On the other hand, A-Kademlia performs similarly as Kademlia in logical topology but shortens the physical latency remarkably by rationalize the routing procedure.

VI. SUMMERY

In DHT networks, detouring problem caused by topology mismatch degrades the system efficiency severely. In this paper, a topology-aware Kademlia is studied to solve it. Based on a presented distributed clustering algorithm, a novel NodeID assignment mechanism is designed to correlate physical topology and logical topology, so that the routing procedure is rationalized. Theoretical analysis proves that the proposed Kademlia is also a $O(\log N)$ running system but more efficient than the original in physical topology. Simulation results show that the proposed structure can reduce the latency by nearly 15% through avoiding the mismatch of two topologies.

ACKNOWLEDGMENT

We are indebted to our colleagues: Hong Shan, Ting Zhao and Haomiao Zhou, for their constructive criticism and helpful suggestions for improving the overall quality of this paper. We also would like to express our appreciation to Doctor Yongying Gao for her helpful comments on this paper.

REFERENCES

[1] S. Ratnasamy, P. Francis and M. Handly, "A scalable content-addressable network," Proc. ACM SIGCOMM, ACM Press, 2001, pp. 161-172, doi:10.1145/383059.383072.

[2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup service for internet application," Proc. ACM SIGCOMM, ACM Press, 2001, pp. 149-160, doi:10.1145/383059.383071.

[3] Y. B. Zhao, J. Kubiatowicz and A. D. Joseph, "Tapestry: an infrastructure for fault-tolerant wide-area location and routing,"

Technical Report: CSD-01-1141, University of California Berkley, 2001.

[4] P. Maymounkov and D. Mazieres, "Kademlia: a Peer-to-peer information system based on the XOR metric," Proc. International workshop on Peer-to-Peer Systems, Springer-Verlag, 2002, pp. 53-65.

[5] A. Rowstron and P. Druschel, "Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems," Proc. the 18th IFIP/ACM International Conference on Distributed Systems Platforms, Springer-Verlag, 2002, pp. 329-350.

[6] K. Gummadi, R. Gumrnadi, S. Gribble, S. Ratnasamy, S.Shenker and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," Proc. ACM SIGCOMM, ACM Press, 2003, pp. 381-394, doi:10.1145/863955.863998.

[7] S. Ratnasamy, M. Handley, R. Karp and S. Shenke, "Topologically-aware overlay construction and server selection," Proc. IEEE INFOCOM Conference, IEEE Press, 2002, pp. 1190-1199.

[8] Y. S. Yu, Y. B. Miao and C. K. Shieh, "Improving the lookup performance of Chord network by hashing landmark clusters," Proc. IEEE International Conference on Networks, IEEE Press, 2006, pp. 1−4, doi:10.1109/ICON.2006.302674.

[9] B. Y. Zhao, Y. Duan, L. Huang, A. D. Joseph and J. D. Kubiatowicz, "Brocade: landmark routing on overlay networks," Proc. International Workshop on Peer-to-Peer Systems Springer-Verlag, 2002, pp.34-44.

[10] B. Krishnamurthy, J. Wang, and Y. Xie, "Early measurements of a cluster-based architecture for P2P systems," Proc. ACM SIGCOMM Internet Measurement Workshop, ACM Press, 2001, pp. 105-109, doi:10.1145/505202.505216.

[11] G. Kwon and K. D. Ryu, "BYPASS: topology- aware lookup overlay for DHT-based P2P file locating services," Proc. International Conference on Parallel and Distributed Systems, IEEE Computer Society, 2004, pp. 297-304, doi:10.1109/ ICPADS.2004.24.

[12] F. Hong, M. Li and J. D. Yu, "PChord: improvement on Chord to achieve better routing efficiency by exploiting proximity," Proc. IEEE International Conference on Distributed Computing Systems Workshops, IEEE Computer Society, 2006, pp. 806−811, doi:10.1109/ICDCSW.2005.108.

[13] Y. Liu, P. Yang, Z. Chu and J. G. Wu, "TCS-Chord: an improved routing algorithm to Chord based on the topology-aware clustering in self-organizing mode," Proc. International Conference on Semantics, Knowledge, and Grid, IEEE Computer Society, 2005. pp. 25−25, doi:10.1109/SKG.2005.121.

[14] Y. Liu and P. Yang, "An advanced algorithm to P2P semantic routing based on the topologically-aware clustering in self-organizing mode," Journal of Software, 2006, vol.17, part 2, pp. 339-348.

[15] Z. C. Xu, C. Tang and Z. Zhang, "Building topology-aware overlays using global soft-state," Proc. International Conference on Distributed Computing Systems, IEEE Computer Society, 2006, pp. 500.

[16] H. J. Wang and Y. T. Lin, "Cone: a topology-aware structured P2P system with proximity neighbor selection," Proc. Future Generation and Networking, IEEE Computer Society, 2007, pp. 43-49, doi:10.11091/FGCN.2007.91.

[17] J. Q. Cui, Y. X. He and L. B. Wu, "More efficient mechanism of topology-aware overlay construction in application-layer multicast," Proc. International Conference on Networking, Architecture, and Storage, IEEE Computer Society, 2007, pp. 31−36.

[18] L. H. Dao and J. W .Kim, "AChord: topology-aware Chord in anycast-enabled networks," Proc. International Conference on Hybrid Information Technology, IEEE Computer Society, 2006, pp. 334−341, doi:10.1109/ICHIT.2006.47.

[19] J. P. Xiong, Y. W. Zhang, P. L. Hong and J. S. Li, "Reduce Chord routing latency issue in the context of IPv6," IEEE Communications Letters, vol. 10, Jan. 2006, pp. 62-64, doi:10.1109/LCOMM.2006.1576571.

[20] J. P. Xiong, Y. W. Zhang, P. L. Hong and J. S. Li, "Chord6: IPv6 based topology-aware Chord," Proc. International Conference on Networking and Services, IEEE Computer Society, 2005, pp. 4.

[21] S. G. Wang, H. Ji, T. Li and J. Q. Mei, "Topology-aware peer-to-peer overlay network for Ad-hoc," The Journal of China Universities of Posts and Telecommunications, vol. 16, Feb. 2009, pp. 111–115.

[22] R. Winter, T. Zahn and J. Schiller, "Random landmarking in mobile, topology-aware peer-to-peer networks," Proc. IEEE International Workshop on Future Trends of Distributed Computing Systems, IEEE Press, 2004, pp. 319-324.

[23] The P2PSim Project, http://pdos.csail.mit.edu/p2psim/, July, 2008.

[24] P. G. Krishna, S. Stefan and D. G. Steven, "King: estimating latency between arbitrary internet end hosts," Proc. ACM SIGCOMM Workshop on Internet Measurement, ACM Press, 2002, pp. 5-18, doi:10.11451/637201.637203.