Establishing efficient routes between personal clouds

Ercan Ucan and Timothy Roscoe

Systems Group,
Department of Computer Science, ETH Zurich

Abstract. We address the problem of establishing efficient routes between nodes in disjoint peer-to-peer overlay networks, motivated by the case of personal overlays, each consisting of an ensemble of fixed, mobile, and virtual devices belonging to an individual user. We argue that the problem of route optimization between such systems is different from both routing between single hosts and inter-domain internet routing – in particular, scale and heterogeneity play a significant role, and the peer networks may wish to hide their topology for privacy reasons. We show that there is a significant tradeoff between efficiency and the degree of network information exposed to one peer network by the other, and present an approach that allows users to flexibly advertise desired information about their networks to one another. In this paper, we focus on optimizing the routes for latency and infer the potential to do the same for various other metrics such as bandwidth, monetary cost and energy consumption.

1 Introduction

In this paper, we address the problem of establishing efficient routes between a pair of personal overlay networks, where each network has incomplete knowledge of the other's topology. We are motivated by the scenario of *personal clouds* [21], ensembles of devices (phones, tablets, PCs, rented virtual machines, etc.) owned by single users, and which interact in a peer-to-peer fashion as an alternative to centralized cloud services such as Facebook, Dropbox and Apple's iCloud.

The problem we address is as follows: how can we establish efficient routes between pairs of nodes in two such personal clouds based on the current network state and user preferences? We argue that this problem is new, and different from both routing between single hosts, and inter-domain internet routing, for several reasons.

Firstly, a number of different routing metrics (latency, bandwidth, monetary cost per byte, energy budget for an object transfer, etc.) are important, sometimes simultaneously. In addition, these properties are likely to change frequently.

Secondly, the overlay nodes themselves, and the available connectivity between them, are highly diverse in these metrics: a 3G wireless link has very different cost, power consumption, and latency from a wired Ethernet, for example. Moreover, some nodes (such as phones) may have limited resources.

Thirdly, the small scale of the networks involved (around 10 devices per user) allows more computationally sophisticated reasoning about the best route for a given operation, exploiting detailed information about the diverse set of options available.

Finally, information about the set of devices in a user's personal cloud – their location, address, type, connectivity, and even in some cases existence – is potentially sensitive data the user may not wish to fully share with peers.

We consider this an important problem to tackle within the context of peer-to-peer device ensembles, not least because resource usage matters a great deal. Money is a scarce resource, and metered 3G/2G data connections or virtual machines rented from cloud providers can incur significant monetary cost, depending on usage. Time is a scarce resource, and a careless data transfer and routing approach in a personal data replication system can overload a single node or a single link for transferring data, even though the transfer could be carried out in a more time-efficient way using other links. Energy is also a scarce resource: a Nokia N900 on a 3G connection sending a file at 150 kbit/s draws 375 mA, and when receiving at 200 kbit/s draws 275 mA [15]. This is more power than consumed by continuously playing an MP3, activating the camera with a preview image, or vibrating the phone battery continuously. Efficient use of resources is critical.

Our contributions in this paper are threefold. First, we introduce and motivate the problem of inter-overlay routing in personal clouds. Second, we show simulation and early system implementation and deployment results which illustrate a significant trade-off between route efficiency between overlays and the degree of information exposed to one peer network by the other. Third, we present our work on an approach that allows users to flexibly advertise information about their networks to one another and optimize routes for metrics such as latency, bandwidth, monetary cost and energy.

In the next section, we present the background and related work. In Section 3 we further motivate the problem using concrete scenarios. In Section 4, we present our initial approach to tackle the problem. Section 5 describes the initial implementation of our research prototype. We present a preliminary evaluation of our approach via simulations in Section 6. In Section 7, we present our real system implementation and experimental results. Finally, we conclude and discuss the ongoing and future work in Section 8.

2 Background and Related Work

Our work on personal clouds is inspired by work on personal storage systems [19, 21–23] which aim to support personal data and content-based partial replication, without the need for centralized online services. In these systems, a user generates new data items by taking photos and videos, downloading music and documents. These items are replicated across the devices according to a system policy, and vary in size from a few kilobytes up to a few gigabytes. Most of the devices in the personal cloud offer multiple options for communication, such as Ethernet, WiFi, UMTS, Bluetooth, and USB.

Perspective [22] is a storage system designed for home devices. Cimbiosys [19] is designed for users to be able to selectively distribute data across their devices by associating content filters through opportunistic peer-to-peer synchronization. Eyo [23] is a personal media collections storage system. Anzere [21] is a personal storage system aimed for policy-based replication showing how to flexibly replicate data in response to a complex, user-specified set of policies in a dynamically changing environment. All of

these systems encounter the inter-overlay routing problem as soon as two instances need to exchange data. Despite influential naming and architecture work in this area such as UIA [8], as far as we know the problem of enabling efficient routes and optimizing data object transfers among different personal clouds has received little attention to date.

Dexferizer [26] presents an approach to optimizing the transfer of data objects *within* a user's collection of computation devices, subject to a variety of user-defined quality metrics such as cost, power consumption, and latency. Dexferizer employs techniques from declarative networking together with application-defined transfer policies and priorities to select appropriate transfer mechanisms and schedules.

We are also influenced by ideas from large-scale network architectures. SCAFFOLD [9] aims at better support for widely-distributed services, and argues that rather than today's host-based unicast, the main abstraction of future networks should be service-based anycast. Content-Centric Networking (CCN) [11] proposes replacing the host-to-host communication scheme of the Internet, arguing that *named data* is a better abstraction for today's network applications than *named hosts*. Similarly, DONA [13] proposes a service-centric approach, arguing that most Internet usage is data retrieval and service access, and this requires clean-slate design of Internet naming and name resolution. The eXpressive Internet Architecture (XIA) [1] argues that elevating only one principal type above others hinders communication between other types of principals and, thus, the evolution of the network. Instead, XIA provides native support for multiple principal types.

Our work is related to the problem of cross-layer visibility [12]. Strong layer abstractions hinder network management tasks, failure diagnosis, and ultimately the reliability of the network [3]. A similar observation has been made Plutarch [5] and Metanet [28]. Plutarch suggests making heterogeneity among different networks explicit through the notion of *contexts* which are linked via *interstitial functions*. Metanet proposes a new architectural object called a *region* as a first-class component of future networks. Our approach is based on similar general observations.

Research on MANETs has greatly contributed to the problem of routing in personal computing environments. MANET routing protocols [4, 6, 18] cannot rely on a centralized component or infrastructure, and must quickly react to device and link failures, changes in topology, and network partitions. MANETs can also be seen as an extension to the Internet: if a node in the MANET has Internet connectivity, it can function as a gateway [7, 24]. However, handling heterogeneity does not appear to be a primary focus of MANETs.

Our research is also related to the work on Content Distribution Networks (CDNs) [10, 16, 27], which aim to address similar problems, in a different context. CDNs aim to choose the *best* replica to deliver data to end users. In this paper, we have a focus on enabling selective advertisement of the network information, depending on the trust level, between the peer overlay networks that are owned by different users. Moreover, our work is in a smaller and more heterogeneous context as compared to CDNs.

The approach we have developed in tackling the problem is partly inspired by BGP [20]. However, BGP addresses a significantly larger and different problem scenario that involves large networks acting as traffic carriers.

3 Motivation and Challenges

In this section, we first motivate the problem further with three use-cases, and outline the design space, trade-offs and the challenges involved.

3.1 Use cases

As a first example, consider the scenario in which Alice meets Bob in a cafe and they want to exchange a short movie file of 100MB. The file is initially replicated on Alice's phone, home computer and office PC. Bob wants to add this item to his collection so that he can watch it later. The phone is running on a 3G connection whereas the home computer and the office PC are on faster broadband connections. In such a scenario, ideally, the phone should be simply an initiator of the data transfer: the file should be moved between machines with better connectivity.

As a second example, consider the scenario illustrated in Figure 1, where Alice and Bob want to make their devices (phone and laptop) talk to each other. One option is to establish a communication through the cloud. Another option which may not always be possible is that, whenever the two devices happen to be on the same WiFi or Bluetooth network (even though TCP/IP over Bluetooth is not very mainstream) and they are aware of each other's available network interfaces, they may be able to establish a much more efficient route. Hence, exposing more network information can significantly improve routing.

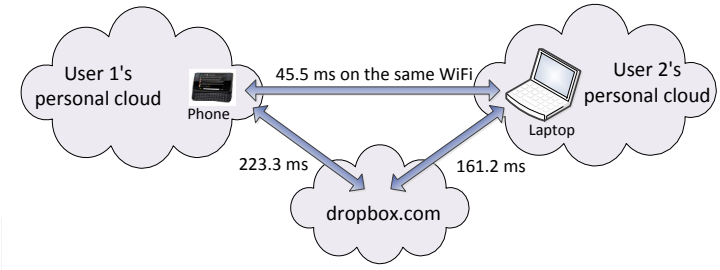

Fig. 1. Exposing more network information can significantly improve routing.

As a third example, imagine that Alice wants to receive a large movie file from Bob who keeps two replicas of the movie, one on his home computer and one on his cloud storage. If Alice happens to own cloud storage from the same provider as Bob and if Bob exposes this information about his network to Alice, then they may be able to perform a very fast copy of the large file in the cloud. Alternatively, Alice could simply flag the file as *shared with Bob* on the cloud storage, without actually having the data to be moved from one location to another. Of course, in practice, the realization of sharing on

the cloud can mean a variety of things such as sharing a URL (*e.g.*, in Dropbox's case). Moreover, it may also require implementation of additional mechanisms/interfaces depending on the cloud provider's mode of operation.

3.2 Design space, Trade-offs and Challenges

As we show later in this paper, there is a trade-off between how much information the users expose about their personal clouds to each other, and the efficiency of subsequent routes and transfers. The more information is available across the peer networks about topology, node locations and capacities, and current network conditions, the better the resulting connectivity can be. In addition, here are some of the technical challenges involved in solving this problem:

- Dynamic nature of connectivity: many of the devices used in a user's personal cloud are mobile. These devices do not have permanent connectivity to the rest of the cloud.
- Routing challenges (NATs, firewalls, mobile devices, etc.): the current Internet architecture (that is, based on end-hosts) poses challenges for routing and data transfer among multiple personal clouds.

4 Design

We start this section by first describing the scenario we target, with disjoint peer overlays and their respective network information and routes. We then illustrate the problem of establishing efficient routes, and present our initial approach to tackle this problem.

4.1 Setting

We can characterize the challenge we address in inter-overlay routing as follows:

- Initially the two disjoint networks do not know any information about each other regarding the nodes they contain, the internal network topology, connectivity interfaces, item distribution, the current state of the network, etc.
- There may be multiple routes and multiple connectivity interfaces between the nodes.
- Links and the topology of the network are dynamic. We might or might not have the schedules of when certain links will appear or disappear.
- The network links have associated costs. This may be monetary cost, power, bandwidth or latency. Moreover, this information may not be known to other peer overlays.
- For privacy reasons, users may want to expose only certain parts of their network topology to each other.

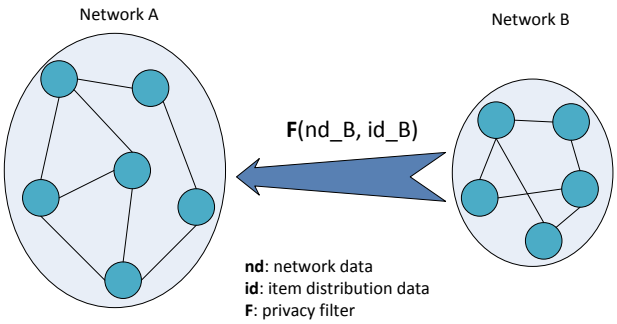

Fig. 2. Overview of our approach at an abstract level

4.2 General approach: Selective advertisement

The overall idea of our approach is inspired by BGP advertisements. We aim to provide the users of personal clouds with the means to flexibly advertise their network information to the users of other peer personal clouds, as they wish, depending on the network, transfer circumstances and user preferences.

Figure 2 illustrates at an abstract level, the general approach of advertising network information between personal clouds. The main idea is that the user (owner of network B in this case) applies his own *privacy filter* to the data of his network and sends this information to the peer overlay. Here we describe further the elements that are shown in the illustration.

- *nd_B*: is the network data that belongs to the network B. It consists of the information such as devices (their hostnames/IP address, port numbers to connect to), the internal network topology, link status information (ping latency, bandwidth estimation etc.). We focus on this type of information in the context of this paper.
- *id_B*: is the item data belonging to network B. Item data consists of the tuples showing where an item is replicated within the personal cloud. This information becomes relevant in the context of scheduling and optimizing object transfers.
- *F*: is the privacy filter applied by the owner of network B. This filter can be adjusted differently by the users of the systems depending on the trust level to each other.

4.3 Calculating efficient routes between two networks

In this section, with the help of an example scenario shown in Figure 3, we describe how we calculate efficient routes between two networks using network advertisements and employing Dijkstra's shortest path routing algorithm in a variant of link-state routing. The figure shows two disjoint networks, network A and B, with their representative weights to represent the network latency. At this point, we do not claim that these topologies and the numbers we show are entirely realistic. Our aim here is to rather

have a means to illustrate our approach and to have an initial reference point for our simulations.

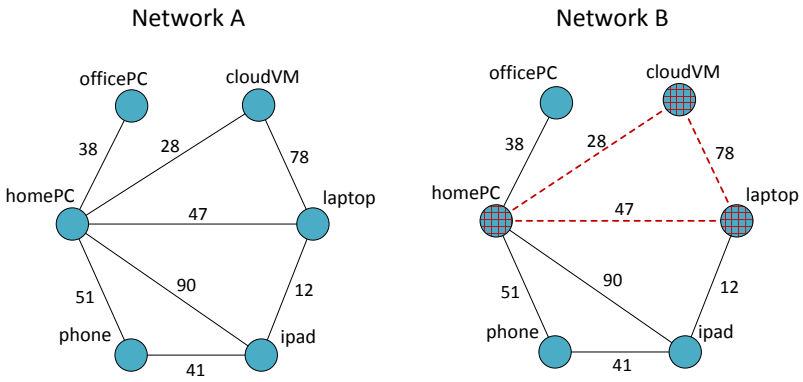

Fig. 3. The example network scenario used in our simulations

Following are the steps we use for the detection and the calculation of efficient routes between the members of two disjoint personal clouds via flexible advertisements of the network information:

1. Network B sends a network advertisement to network A. An advertisement message consists of a list of reachable nodes and topology information between these advertised nodes. In this example, the network B may choose to advertise information about its 3 nodes (homePC, cloudVM and laptop) and part of the connectivity information among them. The information about a node in this case consists of a combination of its hostname/IP address (or a gateway) and the port number to use in order to reach that node. The topology information consists of the edges between the devices together with the weights.
2. Network A then adds the network information it received from network B into its own network knowledge base.
3. Network A tries to establish connections (preferably as many as possible) between each of its nodes and the nodes advertised by the network B. Successfully formed connections are also added as edges to the topology graph of the Network A.
4. Network A then calculates all-pairs-shortest-paths using Dijkstra's algorithm on this extended set of nodes and edges.

If a node is not exposed from the network B, the shortest paths to that node are calculated with the assumption that the exposed nodes of the network B will act as network gateways to the non-exposed node. At the end of the last step, each node in network A knows the shortest paths to the advertised nodes of network B.

5 Implementation

The work we present in this paper is done in the context of a broader project: Anzere [21], a data storage and replication system that we are developing, aimed for personal clouds, integrating personal computers, mobile phones, tablets and virtual machines acquired on demand from cloud providers such as Amazon EC2 and Planetlab. Anzere is a system in which the objects are replicated according to declarative replication policies specified by the users. The policies written in Anzere do not need to refer to specific devices, but are based on device properties. Anzere obtains its replication actions by solving a constrained cost-based optimization problem derived from the set of replication policies. Anzere employs a replication subsystem to replicate user data as well as information necessary for the system operation (*e.g.*, overlay and sensors information). It builds on existing replication techniques, in large part on PRACTI [2]. In order to achieve consistency, Anzere employs Paxos [14]: all nodes in the overlay reach consensus on the total order of updates. The combination of these protocols provides the basis for a broad range of consistency possibilities in Anzere, even though we have not yet worked with update scenarios in the context of this paper.

Anzere currently runs a little over 32,000 lines of Python. The implementation makes heavy use of the Python Twisted framework [25], which is an event-driven networking engine. The software architecture of Anzere is modular, which was initially inspired by the OSGi [17] module management system. The main motivation of this modular architecture is to make the system maintenance easier and also to enable us to customize the functionality and the libraries running on each device based on its hardware architecture and OS platform. The most relevant module of Anzere regarding this paper is the *overlay network*, which includes *network sensors* that run on every node in the ensemble and continuously monitor link and device status. In the context of this paper, the network sensors perform ping latency measurements (such as every 5 seconds) between the devices and keep track of both the instantaneous and the exponentially smoothed RTT values. Anzere also employs models to estimate the expected bandwidth, energy consumption and the throughput of the network links using the measurements performed.

The *storage* module is our primary application driver for this work: it contains *storage sensors* that monitor the status of the data items in the system and partially replicated content according to user-supplied policies. The inter-personal cloud communication architecture we are currently developing makes use of the continuous information flow generated by these two modules in order to compute the optimal routes and object transfer schedules. At the initial simulation stage, the route establishment algorithm was driven using a prototype simulator implemented in Python, with the goal to first establish the potential benefits of our approach. At the system implementation and deployment stage, we employed two disjoint Anzere instances. We present the details of these networks in the following sections.

6 Simulation

Our evaluation consists of two parts: Simulation and system implementation. In this section we present initial simulation results that are aimed at evaluating our approach.

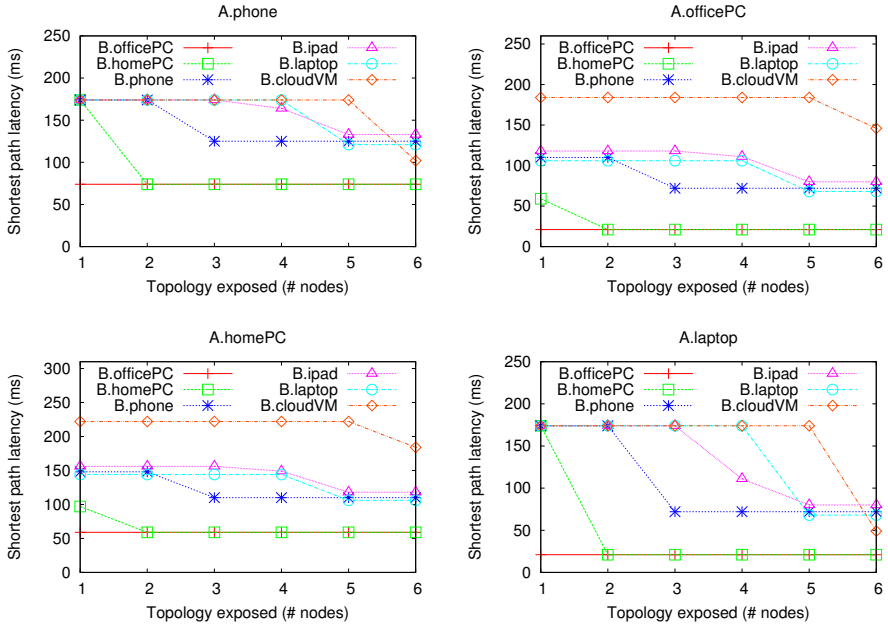

Fig. 4. Shortest path latencies to each node of network B from A.phone, A.officePC, A.homePC and A.laptop

Since personal clouds are newly emerging and they may vary quite significantly from one user to another, currently we do not have extensive data about how a typical personal cloud and its connectivity topology looks like. Therefore, at the simulation stage of our research, we have been trying out our ideas using hypothetical model graphs which in practice may resemble personal clouds. In general, during the set of simulations to evaluate this approach, we identified 3 different parameters which we think are important.

- *Topology exposed*: This parameter denotes how much of the network topology is exposed to the peer network. In these simulations, the network topology is exposed at the granularity of number of nodes, together with *all* the edges that belong to that particular set of nodes.
- *Connecting edges*: This parameter denotes how many connecting edges are present in between the two networks.
- *Weights*: This parameter denotes the weights of the connecting edges.

In the context of our simulations, we have been experimenting with the first item in the list: The network topology that is exposed to the peer overlay.

6.1 Potential benefits of the approach

In this section we show initial simulation results aimed at illustrating how much our approach can be beneficial to a user. At the moment, we experiment with metrics such

as end-to-end latency (msec) and shortest path to the peer overlay, but we conjecture that other metrics such as bandwidth (bytes/sec), cost (price/byte), power consumption (energy/byte) can also benefit from such an approach.

Pairwise end-to-end routes In this simulation we look at the potential benefits our approach can provide in terms of improving pairwise routes between the nodes that belong to two different personal clouds.

For our simulations, we used the two example peer overlays shown in Figure 3. As we mentioned earlier in the paper, we realize that these topologies and the numbers may not be entirely realistic, but they still give us an initial reference point. Our initial experience with a real system follows in the next section of this paper. In this simulation, the network B advertises its network topology to network A and then we look at how the pairwise latencies between the nodes of the two networks are affected depending on the amount of network information exposed. At each stage of the experiment, Network B gradually increases its exposed network topology by one node using the following order of the nodes: officePC, homePC, phone, ipad, laptop, cloudVM. In all the cases (except for the case in which there is only one node advertised from network B), there are 3 connecting edges between the two networks: A.officePC-B.officePC, A.laptop-B.officePC, A.officePC-B.homePC. The weights of all the connecting edges are the same (21) for this simulation.

Figure 4 shows the effect of varying the amount of exposed topology information on the shortest paths to nodes in network B. It shows the shortest path latencies to the nodes of network B from the phone, homePC, laptop and officePC of the network A. The numbers shown here are obtained from a single run of the complete simulation. For the sake of brevity we do not show the latencies originating from the remaining two nodes (ipad and cloudVM) of network A.

These initial figures we present here suggest that maybe not for all, but for some of the node pairs in these networks, changing the amount of exposed network information can significantly improve the pairwise end-to-end route quality.

7 System Experiments

In this section, we present the initial results of our experience with the deployment of our approach in the context of a real system implementation. As mentioned before in Section 5, we implemented our ideas within the Anzere personal storage system [21].

So far, we have investigated two aspects of the approach in our system implementation. Firstly, we investigated the effect of increasing the number of advertised nodes from one Anzere instance to another. Secondly, we looked at the the effect of exposing the internal topology information of an Anzere instance on the shortest paths achievable between the members of two disjoint Anzere instances.

Figure 5 illustrates the network elements and the internal network topologies of Anzere instances we employed in our experiments. The current experimental setup consists of two (initially disjoint) Anzere instances. The first network consists of five devices while the second one contains four devices. Other than the locations noted in parentheses in the figure, the rest of the devices reside in Switzerland.

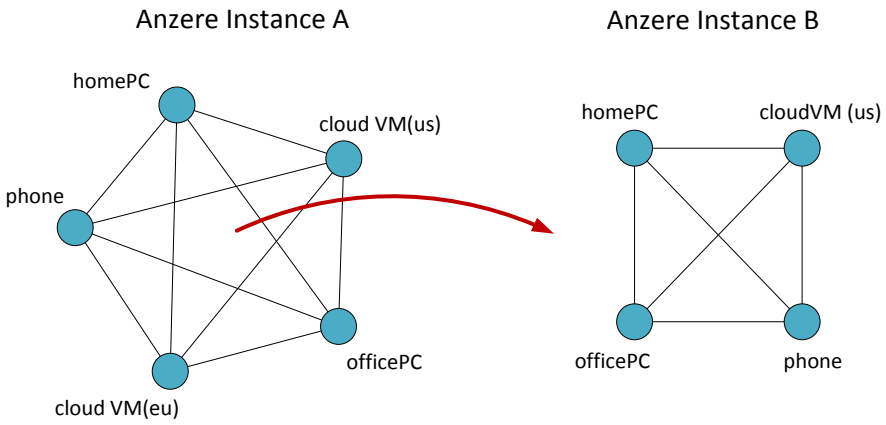

Fig. 5. The network and the topologies of the two Anzere instances used in our system experiments.

Anzere's current network and routing layer is designed to establish as many connections as possible between each member of the different instances when an advertisement is received. Therefore, the current topologies in both of the systems can be visualized as fully-connected graphs. The smart-phones run on a wireless connection and are residing behind a NATs. The rest of the devices have publicly accessible IP addresses. Typically, the homePCs are also behind NATs but the ones we used in this experiment had public IP addresses. In Anzere, the devices that reside behind the NATs utilize hole punching techniques in order to establish connections to the other members of the Anzere instances.

For the experiments performed in this section, the link measurements were taken every 4 seconds and 0.125 (the same as the value used by TCP) was chosen for the value of α, which is used for the exponential smoothing of the measured data. The advertisement messages were sent from the officePC of the instance A to the officePC of the instance B.

7.1 Effect of increasing the number of advertised nodes

In this experiment, we try to investigate whether increasing the number of advertised nodes from one Anzere instance to another changes the shortest paths between the all the members of each instances. The Anzere instance A increases the number of nodes it exposes one by one, with every iteration of the experiment. The information exposed about a node is a combination of the hostname/IP address and the port number to be connected to.

The advertisement scheme in this experiment works as follows: The coordinator node (officePC) of the instance A sends its advertisement to the coordinator node of

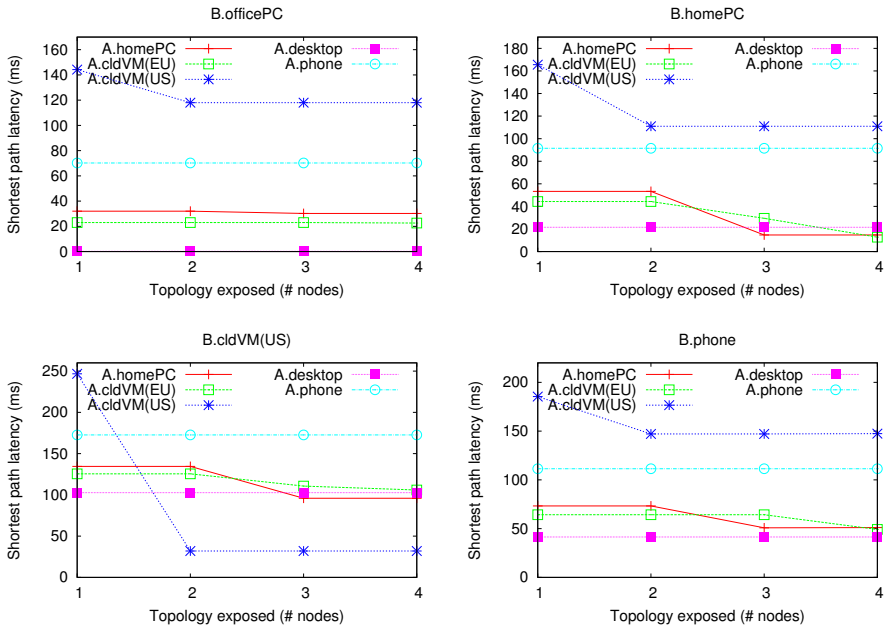

Fig. 6. Latencies to each node in network A originating from B.officePC, B.homePC, B.cloudVM (US) and B.cloudVM (EU)

the instance B. The advertisement consists of the exposed nodes' hostnames (or IP addresses) and the port numbers, in addition to the connectivity and topology information (the links and their weights) that exists between the advertised nodes.

If a node is not exposed from the network A, the shortest paths to that node is calculated by using the assumption that one of the exposed nodes will act as a gateway to this non-exposed node.

The nodes from the instance A to instance B are advertised incrementally in the following order: OfficePC, cloudVM (US), homePC, cloudVM (EU). In other words, at each iteration of the experiment, one more node is added to the advertisement message. As soon as an an advertisement is received, all the nodes belonging to the instance B try and establish connections to the set of exposed nodes of instance A and then start measuring their link properties, in case the connection establishment attempt has been successful.

Figure 6 shows the change in the shortest paths between the pairs of nodes as the number of exposed nodes from instance A to instance B increases. Similar to the case of simulations, the numbers shown here are obtained from a single run of the complete experiment. As illustrated by this figure, our initial experience with the implementation in a real system supports the figures we have presented in the preliminary simulations. The main message of these plots is that, while it may not affect some of the shortest paths between some pairs of the nodes in disjoint personal clouds, depending on the topology and the network configuration, the routes between some pairs devices can be

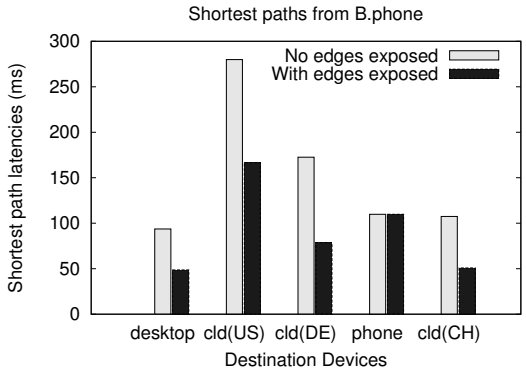

Fig. 7. The effect of exposing internal network topology information on the shortest paths achievable by B.phone to the members of the Anzere instance A

improved significantly by increasing the number of exposed nodes between the two Anzere instances.

7.2 Effect of exposing the topology between the advertised nodes

The advertisement messages in the previous experiment included the nodes exposed as well as the internal network topology information in between the advertised nodes of the instance A. In this experiment we try to investigate whether or not exposing the internal network topology information in addition to the advertisement of the nodes makes a significant difference in terms of the shortest paths achievable between the pairs of devices.

Figure 7 shows the shortest paths from B.phone to the members of the instance A. The shortest path values are shown for both the case in which the internal network topology information (the set of edges and their weights) is shared to the peer Anzere instance, and also the case in which this information is not shared. The four nodes that are exposed from the instance A in this experiment are the following: officePC, cloudVM (US), homePC and cloudVM (EU). The set of devices and the topologies of both the personal clouds employed in this experiment are the same as in Figure 5 except for the location of the cloud VM in Europe. A VM in Switzerland was employed instead of a VM in Germany due to the failure of the instance in Germany. Again, the numbers shown here are obtained from a single run of this experiment.

As shown by Figure 7, exposing the internal network topology of an Anzere instance can actually reduce the shortest path latencies to some of the instance A nodes significantly.

8 Conclusion

Establishing efficient routes between personal clouds and the higher-level problem of optimally transferring data objects among personal clouds are new and important problems. In this paper we have presented a technique which provides users with a means to selectively advertise their network information to each other and still arrive at efficient routes. Current results show the benefits for one metric, latency.

In our ongoing work, we are integrating this technique into our personal cloud platform, and incorporating other metrics such as bandwidth, power consumption and monetary cost. Our initial observations show that one can really get more or less benefit from exposing more or less information. Hence, the tradeoff is significant. As an immediate future work, we are planning to extend our approach and apply the idea to the larger problem of optimizing data object transfers between personal clouds. This involves implementation of advertising item distribution data between the two personal clouds in addition to advertising network information.

References

1. A. Anand, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, and P. Steenkiste. XIA: an architecture for an evolvable and trustworthy internet. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets '11)*, pages 2:1–2:6, New York, NY, USA, 2011. ACM.
2. N. M. Belaramani, M. Dahlin, L. Gao, A. Nayate, A. Venkataramani, P. Yalagandula, and J. Zheng. PRACTI Replication. In *Proceedings of the 3rd symposium on Networked Systems Design & Implementation (NSDI '06)*. USENIX Association, 2006.
3. E. A. Brewer, Y. H. Katz, Y. Chawathe, S. D. Gribble, T. Hodes, G. Nguyen, M. Stemm, and T. Henderson. A network architecture for heterogeneous mobile computing. *IEEE Personal Communications*, 5:8–24, 1998.
4. T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). *Internet RFCs*, RFC 3626, 2003.
5. J. Crowcroft, S. Hand, T. Roscoe, R. Mortier, and A. Warfield. Plutarch: An argument for network pluralism. In *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA '03)*, pages 258–266, 2003.
6. S. R. Das, C. E. Perkins, and E. M. Belding-Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00).*, pages 3–12, March 2000.
7. Y. S. Elizabeth, Y. Sun, E. M. Belding-Royer, and C. E. Perkins. Internet connectivity for ad-hoc mobile networks. *International Journal of Wireless Information Networks*, 9(2):75–88, 2002.
8. B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. Persistent personal names for globally connected mobile devices. In *Proceedings of the 7th symposium on Operating Systems Design and Implementation (OSDI '06)*, pages 233–248. USENIX Association, 2006.
9. M. J. Freedman, M. Arye, P. Gopalan, S. Y. Ko, E. Nordstrom, J. Rexford, and D. Shue. Service-centric networking with SCAFFOLD. Technical Report TR-885-10, Department of Computer Science, Princeton University, September 2010.

10. M. J. Freedman, E. Freudenthal, and D. Mazières. Democratizing content publication with Coral. In *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation (NSDI'04)*. USENIX Association, 2004.
11. V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT '09)*, pages 1–12. ACM, 2009.
12. R. R. Kompella, A. Greenberg, J. Rexford, A. C. Snoeren, and J. Yates. Cross-layer Visibility as a Service. In *Proceedings of 4th Workshop on Hot Topics in Networks (HotNets IV)*, 2005.
13. T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communication (SIGCOMM '07)*, pages 181–192. ACM, 2007.
14. L. Lamport. The part-time parliament. *ACM TOCS*, 16(2):133–169, 1998.
15. Nokia N900 Hardware Power Consumption. http://wiki.maemo.org/N900_Hardware_Power_Consumption.
16. E. Nygren, R. K. Sitaraman, and J. Sun. The Akamai network: a platform for high-performance internet applications. *SIGOPS Oper. Syst. Rev.*, 44(3):2–19, Aug. 2010.
17. OSGi Alliance. *OSGi Service Platform, Core Specification Release 4, Version 4.1, Draft*, 2007.
18. C. E. Perkins and E. M. Royer. Ad-Hoc On-Demand Distance Vector Routing (AODV). In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pages 90–100, February 1999.
19. V. Ramasubramanian, T. L. Rodeheffer, D. B. Terry, M. Walraed-Sullivan, T. Wobber, C. C. Marshall, and A. Vahdat. Cimbiosys: a platform for content-based partial replication. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI '09)*, pages 261–276, 2009.
20. Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). *Internet RFCs*, RFC 4271, 2006.
21. O. Riva, Q. Yin, D. Juric, E. Ucan, and T. Roscoe. Policy expressivity in the Anzere personal cloud. In *Proceedings of the 2nd ACM symposium on Cloud Computing (SOCC '11)*. ACM, 2011.
22. B. Salmon, S. W. Schlosser, L. F. Cranor, and G. R. Ganger. Perspective: semantic data management for the home. In *Proceedings of the 7th conference on File and storage technologies (FAST '09)*, pages 167–182, 2009.
23. J. Strauss, C. Lesniewski-Laas, J. M. Paluska, B. Ford, R. Morris, and F. Kaashoek. Device-Transparency: a New Model for Mobile Storage. In *Proceedings of the Workshop on Hot Topics in Storage and File Systems (HotStorage '09)*, October 2009.
24. P. Stuedi, M. Bihr, A. Remund, and G. Alonso. SIPHoc: Efficient SIP Middleware for Ad Hoc Networks. In *Proceedings of the 8th ACM/IFIP/USENIX International Middleware Conference (Middleware 2007)*, pages 60–79, November 26-30 2007.
25. Python Twisted. http://twistedmatrix.com/trac/.
26. E. Ucan and T. Roscoe. Dexferizer: a service for data transfer optimization. In *Proceedings of the 19th International Workshop on Quality of Service (IWQoS '11)*, pages 33:1–33:9. IEEE Press, 2011.
27. L. Wang, K. S. Park, R. Pang, V. Pai, and L. Peterson. Reliability and security in the CoDeeN content distribution network. In *Proceedings of the annual conference on USENIX Annual Technical Conference (ATEC '04)*. USENIX Association, 2004.
28. J. T. Wroclawski. The Metanet. White Paper. In *Proceedings of Workshop on Research Directions for the Next Generation Internet*, 1997.