CARMA: A Distance Estimation Method for Internet Nodes and its Usage in P2P Networks

Gennadiy Poryev
National Technical University of Ukraine "KPI"
Kiev, Ukraine
core@barvinok.net

Hermann Schloss
Logica Deutschland
GmbH & Co. KG
Hennef, Germany
hermann.schloss@logica.com

Rainer Oechsle
Trier University of Applied Sciences
Trier, Germany
oechsle@fh-trier.de

***Abstract*—Topological distance estimation is the key to the efficiency in distributed systems and peer-to-peer networks. Contrary to many existing or proposed methods, which usually require the exchange of messages between the nodes, we have developed a metric, which is computed purely within a node, and which is based on the preloaded and precomputed topological structure of the Internet. Many distributed systems and applications may benefit from this metric, since it estimates the topological distance between any arbitrary pair of nodes in the Internet. As a proof of concept we have first shown the correlation between our metric and a few established distance indicators, such as hop count or round trip time of a message. Then, we employed this metric as an "edge weight" representing the connection quality between two network nodes and we used it for the construction of a multicast overlay network based on a Minimum Spanning Tree approximation. According to the evaluation results, this metric corresponds fairly well to the actual measured distances. By using this metric, our approach minimizes communication costs and avoids extraneous communication needed for latency measurements.**

***Keywords*-Internet, Topology, Distance Estimation Method**

I. Introduction

Since the beginning of the 21st century the usage, scale and diversity of *peer-to-peer* (P2P) networks widened significantly, and the application scope of P2P systems has been notably extended. Being previously considered as a means for file sharing or instant messaging, today's P2P networks serve as a basic infrastructure for a wide range of innovative application scenarios such as VoIP, multimedia on-demand, software delivery, massive multiuser environments or online games.

The initial driving motivation behind P2P systems was to relieve load stress from centralized server farms. However, the intrinsic asymmetry of end-user broadband data links have caused *Internet Service Providers* (ISP) to increase maintenance and upgrade cost of the "last mile" hardware in order to keep quality of service steady. Some ISPs had also introduced controversial measures to detect and forcibly shape end-user bandwidth for traffic recognized as P2P, affecting file sharing networks in particular.

For this reason, researches in the area of P2P systems are aiming to optimize P2P traffic and consider the inherently clustered nature of the Internet as a potential leverage mechanism. The general idea is to maximize network throughput inside the particular network clusters while minimizing the traffic usage between such clusters. The scope of the cluster is not defined clearly, and there is usually more than one clustering layer.

In this paper, we propose a locally computed approach for topological distance estimation that does not rely on third-party non-guaranteed external infrastructures and consider its usage in P2P networks.

The rest of this paper is organized as follows. In Section II we first take a look at related work dealing with traffic optimization in P2P networks and consider then several application-level multicast approaches, based on a *Minimum Spanning Tree* (MST) or on its approximation. Afterwards in Section III, we describe the computation of the *Combined Affinity Reconnaissance Metric Architecture* (CARMA) metric [1]. In Section IV we consider two application scenarios, which benefit from the utilization of the CARMA metric: a) the selection of peers – network nodes that may serve the requested content – and b) the construction of the ***CARMA**-based **M**ultica**st** **In**frastructure* (*CARMIn*), based on the MST approximation method. Then in Section V we discuss the experimental validation of the CARMA metric and explain the flavor distribution, anomalies and statistical characteristics. As the evaluation results show (Section V), our CARMIn multicast tree achieves a good MST approximation with respect to a communication cost metric and avoids – due to utilization of CARMA – extraneous communication needed for latency measurements. In Section VI we provide a brief overview of our contribution and discuss our future plans.

II. Related Work

Usenet [2] was introduced 30 years ago as one of the first P2P networks. However, only at the end of the 1990s P2P applications have achieved a breakthrough and become very popular because of the widespread use of file sharing platforms like *Napster* [3]. Nowadays, there is a wide variety of P2P file sharing networks. Among them are *Gnutella*

[4], *eDonkey2000* (ED2K) [5] and *BitTorrent* [6]. In [7] a multicast P2P overlay is described that is used for content distribution in large-scale enterprise networks. The proposed approach reduces the completion time compared to BitTorrent without wasting additional resources.

Various surveys suppose that 30% to 50% of today's end-user-generated traffic is caused by P2P applications. In [8] the authors claim that most P2P systems use application-level routing based on the overlay topology and completely neglect the topology of the underlying transport network. Because of this, P2P systems cause a lot of extraneous traffic. In order to avoid this traffic, the authors propose the ISP-aided neighbor selection by considering the node proximity in the underlying network at the application-level.

The authors of [9] have recently described the design, deployment and evaluation of an approach, minimizing the expensive cross-ISP traffic. The authors show that the application of their approach significantly reduces the latency delays. The *P4P* architecture [10] also aims towards the minimization of the network traffic. In order to achieve their objective, the authors take into account the conditions of the underlying network layer during the overlay construction.

According to [11] the consideration of a node's topological locality is the key to efficient communication in P2P systems. It improves performance and increases availability, since the probability of transmission failures increases with the distance and depends also on bandwidth conditions.

Modern network modeling environments that deal with network topology rarely take locality into account. Most of them use either the network latency metric measured in time units between request and response (ping), or the hop count metric measured as the number of nodes between source and destination hosts [12]. We deem the ping method as generally unreliable as it heavily depends on link speeds and bandwidth conditions. For example, a zero-loaded end-user ADSL link can produce slower pings than an almost fully loaded Gbps link. As shown in [13] standard routing trace methods may also be unreliable and affected by bandwidth conditions or indicating non-existent links due to traffic switch-overs.

A number of researches have proposed schemes that involve building the external (in relation to the P2P overlay) infrastructures dedicated to keeping track of the condition of intra-network and inter-network links, remembering explicitly measured routing paths and delivering path prediction on their basis. Such schemes, for instance, include P4P [10] and *iPlane* [14]. Other proposals, such as [15] are concerned with an active intervention into the P2P exchange protocols to augment traffic usage patterns in accordance with ISP policies.

Contrary to the methods employing external infrastructures for distance estimation, we propose an approach based on the preloaded and precomputed topological structure of the Internet and running locally on client machines. By using our distance estimation method, clients are able to create a multicast overlay at the application-level without relying on a central instance or external infrastructures.

While the problems of scalable data localization have been exhaustively addressed, the problem of reducing *multicast costs* in very large, global scale environments still remains inadequately considered. In [16] the authors state that multicast has become an important communication primitive in P2P networks. The authors note that the consideration of communication cost caused by multicast overlays is a critical issue in P2P networks due to dynamic and rapidly changing network topology conditions.

In [17] the authors argue that nowadays the network infrastructure itself becomes a precious resource. They state that the construction methods of multicast trees considerably influence the network load and that current available strategies often waste too many network resources. In order to adjust the multicast tree infrastructure to the physical network conditions the authors propose the use of transmission delays between peers as a performance metric. On the basis of this metric, the authors construct a network friendly multicast tree. However this metric cannot be applied in advance without transmitting a message between two peers.

To address this issue of reducing multicast costs in our work, we propose the construction of an application-level multicast infrastructure, based on an MST approximation. The MST problem is one of the most popular and important problems in the research area of graph theory, distributed computing and networks. In opposition to the theoretical models where we usually have a global knowledge of all nodes and the corresponding distances for the MST construction, in a realistic network (e.g., the Internet) a node neither knows all other nodes involved in the same application scenario nor exact distances between these nodes.

The ALMI (Application Level Multicast) project [18] uses a central instance for MST computation. In [19] the authors propose a MST-based multicast cluster for P2P video streaming and show that the utilization of the MST approach reduces the network traffic. However, since here all network nodes are considered for MST construction, high management and maintenance costs can be expected in large scale networks. In our CARMIn approach we avoid the additional communication by using the CARMA metric and address the question of a distributed approximation of an MST that is, constructing a suboptimal spanning tree whose communication cost is near-optimal.

Similar to our approach, [20] have considered the problem of the construction of suboptimal spanning trees. In [21] the authors propose the construction of a *Nearest Neighbor Tree* (NNT) instead of an MST. To ensure both acceptable multicast costs and latency delays JXTA [22] nodes always connect to the nearest node (in terms of latency) achieving an MST approximation, too. However, the quote in [23] "There is no satisfactory approximation algorithm known for the

MST problem" encouraged us to address this problem in our work.

In [24] the authors propose a *binning* scheme by adjusting adjacent nodes to certain bins depending on their *Round Trip Time* (RTT) distance to certain landmark severs. To be more exact, a node measures its round-trip time to each of these landmarks and orders these landmarks in ascending order. Nodes having the same order of landmarks are considered closer than nodes having different order. This approach significantly reduces the amount of communication necessary for the capturing of node distances. However, the communication with the landmark servers for the RTT measurements is required.

Several MST or NNT-based approaches readjust their infrastructures when nodes are joining or leaving the infrastructure. However under churn (i.e., peers arriving and departing at a high rate) this readjustment makes them useless for large scale application scenarios. Thus in our CARMIn approach only direct neighbors are involved into joining and leaving of peers in this way avoiding the readjustment. Similar to our approach in [25], the authors propose the Orchard algorithm for building and maintaining application-level multicast trees taking into account the problem of churn.

III. CARMA METRIC

The Internet is in no way a uniform structure. There are large backbone networks involved in international and intercontinental links, national-tier ISPs, end-user-servicing ISPs, hosting companies and end-users. Network latency and quality of service are accordingly very different depending on the link speed from tens of Gbps to less than 56 Kbps for dial-up modems. On the scale of a country, the Internet structure used to be organized rather sporadically – individual ISPs established arbitrary links among themselves and to foreign upstream ISPs. This had lead to peering conflicts and situations in which a message to a neighboring house traveled halfway the continent.

To alleviate this problem, *Internet Exchange Points* (commonly abbreviated as IX or IXP) were introduced. Usually, a number of national telecom operators create the dedicated facility to which all national ISPs then connect. Thus, consumer traffic within the scope of an IX does not travel expensive international or satellite links. This helps balance mutual peering and to ensure lower maintenance costs per ISP, allowing lower prices for end-users. Developed countries are used to having more than one nationwide IX. From the customer point of view, it is generally assumed that traffic within a single IX scope flows faster and is cheaper than between different IXes. The presence of an IX can also provide for a lower hop count in the packet path. Figure 1 depicts an Internet segment consisting of some networks, which are grouped by *Autonomous Systems* (AS) [26]. Some of these ASes are connected to a single IX.

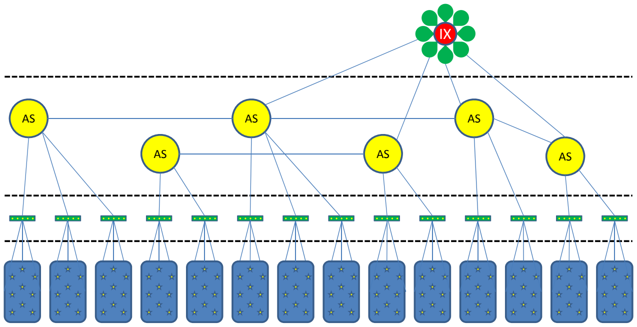


Figure 1. Example of an Internet segment covered by an IX

In network-based applications, we often need global knowledge of all network nodes and distances between these nodes. This information is usually managed by a central instance or may be inferred from external infrastructures. By having this knowledge, nodes in a network are able to construct complex infrastructures and achieve efficient communication at the application as well as at the network layer.

Without relying on a central instance or external infrastructures, clients usually apply either ping or hop count methods to estimate node distances. The problem hereby is that even if the ping or hop count methods would provide reasonable and reliable results, there is no way to apply these methods to a pair of foreign IP addresses. That is, it is easy to measure the ping or hop distance from the node a to the node b or to the node c. But there is no way to measure the ping distance between the nodes b and c from the node a. That means, if clients are interested in this kind of information, they have to explicitly request this information from corresponding nodes, which causes a significant communication overhead.

To sum up, the construction of complex structures requires either additional communication between nodes (in decentralized P2P systems), or is not scalable due to the existence of a *Single Point of Failure* (in P2P systems with a central instance) managing relevant information, which is a big drawback in global scale networks.

In order to sidestep these drawbacks, in our work we employ decentralized P2P systems and propose the combined affinity metric, which is calculated locally on each node. This metric is calculated given the remote IP address of the peer and all information then can be implicitly inferred from it. The "combined" adjective in the CARMA acronym means that despite our work's prevailing focus on only the first layer of the proposed metric, its design does, however, contain several additional components that can be included in a metric calculation in the future, as follows:

1) average response time to keep-alive requests;
2) average hop count to the destination, including the possibility of its change during communication [13];

3) bandwidth and average consumption at the moment of decision, including preset constraints;
4) "gratitude" and "greed" values calculated as the amount of traffic the remote party had provided and consumed respectively.

Generally speaking, we consider CARMA as three-layered, with the first layer being the locality awareness expressed in flavors (see their explanations below), a second layer that utilizes additional traffic but does not involve actual P2P communication (see Section V-A), and a third layer that requires active communication to remote parties over a compatible protocol.

CARMA works by initially preloading structural information from publicly accessible services called *Regional Internet Registries* (RIRs) and converting it into an internal graph-like data structure. Unlike solutions based on the PlanetLab infrastructure or those using RouteViews, the RIR services and databases are mandatorily public, essential for the functioning of the Internet and therefore much more reliable. The pieces of information important to CARMA include the *delegated-latest-** databases of registered IPv4 ranges, *Autonomous System Numbers* (ASNs) and various WHOIS databases of registered subranges and *Autonomous System Sets* (ASSETs).

Since the RIR database expansion rate is relatively slow and the IPv4 address space is nearing its exhaustion, CARMA only needs to update its locally cached data from those databases once every few days. Once loaded, CARMA builds a model to approximate the Internet topology with some simplifications, resulting in 4 structural layers as follows: a) IPv4 ranges are divided into b) subranges but at the same time they also belong to c) Autonomous Systems (ASes), which are joined into sets called d) AS-SETs or ASSETs. It is assumed that lower layer entities are explicitly connected through their common upper-layer entity, and AS-SETs are arbitrarily connected to each other. It is understood that such assumptions in the model are more optimistic than what happens in reality, as there may be ASes that spread worldwide, for example. However, exceptions like this are not numerous and pertain only to the departments of the few telecom operators specializing in providing transoceanic and transcontinental links. Therefore, Internet end-users are unlikely to be encountered in the ranges assigned to such ASes.

Let's take a closer look at IPv4 ranges and subranges, ASes, and ASSETs (with examples from the RIPE registry, which is responsible for Europe):

IPv4 range – a subset of an IPv4 address space defined by the first address of the range and a host count. Note that the host count is not necessarily a power of 2 as implied by the *Classless Inter-Domain Routing* (CIDR) rules now commonly used for Internet routing. There are records that specify an arbitrary number of nodes, but for practical reasons such definitions are subsequently augmented by CARMA to contain a power of 2 number of nodes. IPv4 ranges are defined in the *delegated*-file (see Listing 1).

```
ripencc|EU|ipv4|143.65.0.0|65536|19900326|assigned
ripencc|EU|ipv4|143.93.0.0|65536|19940413|assigned
ripencc|NO|ipv4|143.97.0.0|65536|20070104|assigned
ripencc|EU|ipv4|143.99.0.0|65536|19900907|assigned
```

Listing 1. Excerpt from a database file with IP ranges

```
ripencc|EU|asn|2857|1|19931227|allocated
ripencc|EU|asn|2858|1|19940112|allocated
ripencc|SE|asn|2859|1|19940127|allocated
ripencc|EU|asn|2860|1|19940118|allocated

route:        143.93.192.0/18
descr:        FH-RPL-NET
origin:       AS2857
mnt-by:       AS2857-MNT
changed:      weiss@uni-mainz.de 20001220
source:       RIPE
```

Listing 2. Excerpt from database files with AS definitions and relations

AS – a registered Autonomous System. Every AS has a numerical identifier known as *Autonomous System Number* ASN. AS definitions are also listed in the *delegated*-file along with the ISO country code and the date of allocation. However, this file does not specify a relationship between IPv4 ranges and ASes. For this relationship CARMA uses the *ripe.db.route.gz* file (see Listing 2). The latter file contains definition blocks, each block specifies an IPv4 range (this time in proper CIDR notation), and related ASes. This information is used to establish relationships between ranges and ASes listed in the *delegated*-files. Note that a relationship between an IPv4 range and an AS is not unambiguous: The same range can be announced under different ASes; some ASes or ranges listed in the *delegated*-file may not be linked at all, and some relationships specified in *ripe.db.route.gz* may contain ASes and IPv4 ranges, which are unspecified in the *delegated*-file. The incidence of such inconsistencies is low, however.

IPv4 subrange – a subset of the IPv4 address space defined by the addresses of the first and last address of the subrange. These definitions are listed in the *ripe.db.inetnum.gz*-file (see Listing 3). The subranges differ from ranges in that they are not explicitly related to an AS. Subranges are generally smaller in terms of address space. A vast majority of them are derived from splitting up ranges. It is therefore possible to establish a relationship between one or more subranges and a single range, although not all ranges are split into subranges. When parsing information from this file, one should take care to check for sanity of the subranges specified. For instance, a subrange may specify an entire IPv4 address space, or a subrange may even have a netmask length such as 3 bits and may thus be much larger than an IPv4 range. Such cases are dictated by the internal workings of the WHOIS server software but are obviously invalid for

```
inetnum:        143.93.32.0 - 143.93.63.255
netname:        FH-RPL-NET
descr:          Fachhochschule Trier
descr:          Rechenzentrum
descr:          Schneidershof
descr:          D-54293 Trier
country:        DE
admin-c:        KM624-RIPE
tech-c:         RB373-RIPE
status:         ASSIGNED PI
mnt-by:         TRANSKOM-MNT
changed:        hostmaster@transkom.net 20050207
source:         RIPE
```

Listing 3. Excerpt from a database file with IPv4 subranges

```
as-set:         AS-DECIX-CONNECTED
descr:          ASN of DE-CIX members
descr:          DE-CIX, the German Internet Exchange
admin-c:        AN6695-RIPE
tech-c:         WT6695-RIPE
tech-c:         DM6695-RIPE
tech-c:         SJ6695-RIPE
notify:         notify@de-cix.net
mnt-by:         DECIX-MNT
source:         RIPE
changed:        auto-upd@de-cix.net 20091011
members:        AS42
...
members:        AS2828
members:        AS2857
members:        AS2914
...
members:        AS65333
```

Listing 4. Excerpt from a database file with ASSET definitions

CARMA and therefore filtered out of the model.

AS set or *ASSET* – a topological junction point that may declare an arbitrary number of ASes and other ASSETs and facilitate connectivity among them. It is assumed that the information flow between two ASes belonging to the same ASSET does not take a route via other ASSETs. Unlike ASes, ASSETs have alphanumeric identifiers. In terms of CARMA, an IX point is an ASSET with a significant number of member ASes (usually hundreds), although, technically, every ASSET can be considered as a kind of IX as there is usually no explicit requirement in terms of member count. The definitions for ASSETs can be found in the *ripe.db.as-set.gz* file (see Listing 4).

When all database files are processed, the resulting incomplete graph reflects the Internet topology as close as it could be done without having access to *Border Gateway Protocol* (BGP) information. It is not necessary to devise any graph-walking algorithm to calculate the affinity value subsequently called *flavor*, because the purpose of CARMA is to estimate the affinity of two given nodes, not calculating an exact hop count. The proposed flavors of the remote node in relation to the originator node are given below in the order of corresponding tests undertaken by CARMA:

1) *Subrange* identifies the presence of the remote node's IP address in the same IPv4 subrange specified in the *ripe.db.inetnum.gz* database file dealing with administrative IP subranges. However, if such a presence is found, CARMA does not immediately return this flavor, because subranges may overlap with different netmask lengths, which in turn may happen to be shorter than that of the corresponding range (see below). This flavor identifies the presence of the remote node most likely within the scope of operation of a single router or the same network operations center. For example, this could be for end-users connected to the same point of presence of a telecom operator, or nodes within a university network, which usually have single upstream ISP.
2) *Range* identifies the presence of the remote node's IP address in the same IPv4 range specified in the *delegated*-file or the *ripe.db.route.gz* WHOIS excerpts dealing with ASNs and IPv4 delegations. If the subrange lookup yielded any results, the ranges found are examined and compared in terms of netmask length. In this case, the range flavor is only returned if the shortest range netmask is shorter than or equal to that of a subrange, otherwise the subrange flavor is returned. This ensures that the subrange flavor is never returned for allocations larger than the corresponding range, even if they overlap. This flavor means that both nodes most likely reside within the scope of the same department or the same small organization, and that the traffic between these nodes is unlikely to travel outside of the single business network of their ISP.
3) *AS* identifies the presence of the remote node's IP address within the address space allocated to the same AS as defined in the *ripe.db.route.gz* file. Although this fact does not guarantee such an immediate connectivity as the previous flavors, packets are unlikely to travel networks outside this AS, since an AS is the basic Internet routing entity [26]. This flavor suggests that the traffic between two nodes is handled by the ISP internally, and that incoming traffic from outside of the Internet destined to one node undergoes the same routing rules as traffic to the other node.
4) *ASSET/IX* states that both nodes belong to different ASes announced by the same ASSET, which may happen to be an IX if the number of member links is large enough (not every ASSET is an IX, but all IXes are ASSETs). The immediate advantage of this knowledge is not obvious, but in developing countries the difference in quality of service may largely depend on this flavor to the extent that network speed and latency differ by some *orders of magnitude* for nodes within and outside of the same IX. In such national Internet configurations ISPs often implement mandatory traffic shaping policies to limit the packet flow to and from outside of the IX.
5) *ASSET-link* indicates that the node addresses belong to different ASes, which belong to different ASSETs,

and at least one ASSET includes the other ASSET.

6) *Backbone* indicates that the node addresses belong to different ASes, which belong to different ASSETs, and both ASSETs are declared within the scope of a third ASSET.
7) *Distant* identifies that all previous CARMA affinity tests had failed to yield a positive match, and the relative locality of the originator and target node cannot be reliably estimated. Therefore they are assumed to be located topologically far away.

By using these flavors any arbitrary pair of IP addresses can be assigned to an affinity cluster, which in the most cases corresponds to the real topological distance between these nodes.

IV. CARMA-based Application Scenarios

In this section we take a look at two application scenarios, which benefit from the CARMA distance estimation method. The first scenario considers a preselection of peers in a P2P file sharing application. The second scenario discusses the construction of a multicast infrastructure in P2P networks based on an MST approximation.

A. *CARMA-based Peer Selection*

Regardless of the differences in their protocols and implementations, there is something common in all file sharing networks. That is, after the request for a published entity is processed by either the indexing server or other nodes, a response is obtained in the form of a list of peers. Whether this is done using Distributed Hash Tables (DHT) such as *Kademlia* [27], indexing servers or message flooding [4], the result will contain at least a list of IP addresses and ports.

¿From this point on, it is completely up to the client software to decide which nodes should be queried and in what order. From our previous experiences of analyzing the ED2K and BitTorrent network traffic from a single node, we found out that the client software usually performs queries in the order, which was initially reported by the network or index servers.

By their design ED2K clients will query every known source and will attempt to place themselves in the download queue of every source they managed to successfully negotiate with. The other (receiving) side will organize the download queue initially according to the FIFO principle. Modern clients (eMule [28] and its numerous clones) also feature a reward system, which advances inbound clients in the queue according to the amount of related traffic they had provided to the node. This is supposed to discourage leeching but also has obvious drawbacks in delaying new nodes that do not have any part of the content yet.

Although eMule provides a few tuning methods such as queue rotation, speed and chunk management based on the popularity of the file, none of them takes into account anything related to connectivity (client bandwidth, network latency etc.), let alone the geographical location or topological affinity.

In the popular BitTorrent network, the number of peers for highly-demanded content can easily reach tens of thousands, whereas for most end-user nodes it is quite impractical to initiate more than a hundred connections simultaneously, even when having high-speed links.

The BitTorrent protocol is simpler than ED2K. It does not feature any reward system, and due to the per-content swarm isolation BitTorrent is generally faster. Also, a tracker may not report all peers to the client initially. However, this is usually circumvented later by the peer exchange and DHT mechanisms.

Recently there have been some advances in the locality awareness for BitTorrent networks. Popular nationwide trackers (rutracker.org, for instance) have introduced so-called "retrackers" – dedicated secondary servers. These servers are optionally connected to a primary database, but mainly supposed to only return a peer list local to a specific network scope. This scope usually consists of an IP address pool allocated to customers of a particular ISP, or, more frequently, contains the private unroutable IP ranges of a local intranet. This provides for a significant speed burst for affected ISP clients, but it is a very simple method that only allows for a two-tier locality awareness.

We believe that it is essential to not leave the peer selection process to pure luck. In our previous research in the area of file-sharing networks [29] we figured out a method, which could be used to improve the performance of these networks. The key to the performance improvement is the consideration of CARMA distance estimation flavors for the arrangement of the peer query order. That is, choosing the peers with the lowest flavors would reduce the network latency and increase the exchange speed. In Section V-A we evaluate the quality of the CARMA metric and its impact to peer selection in file sharing applications.

B. CARMIn - ***CAR**MA-based **M**ulticast **In**frastructure*

In this section we propose a multicast infrastructure based on an MST approximation. For a large scale number of network nodes the construction of an MST as a communication tree T will lead to unacceptable high network maintenance costs in the case of joining, leaving or failing of nodes. Hence, we have to find a tradeoff between the minimization of multicast costs and latency delays on the one hand and acceptable network maintenance costs on the other hand.

One problem of constructing an MST in real networks is the fact that we do not know exact distances between the nodes (latency delays) as we do in a graph theoretical setting. Measurements of the round trip latency between nodes for the purpose of distance acquisition by sending extraneous ping messages induce an inacceptable high communication overhead in large scale networks and hence have to be avoided. As mentioned before, CARMA flavors indicate the

```
Algorithm 1: Join Operation
  Input: node $n$, spanning tree $T = (V, E_T, w)$ and
         bootstrapping set $BS \subseteq V$ ;
  Output: $T' = (V', E'_T, w)$ including $n$;
  begin
      if $V \neq \emptyset$ then
          Arrange $v \in BS$ such that
          $\forall\ v_i \in BS : w(v_i, n) \leq w(v_{i+1}, n)$ holds;
          $E'_T = E_T \bigcup \{n, v_1\}$, where $v_1$ is the first node
          in $BS$;
      $V' = V \bigcup \{n\}$;
```

node locality by telling whether a remote peer belongs to the same subnet, the same AS, the same IX, and so on. Therefore, in our approach we utilize the CARMA flavors as a distance substitute for a spanning tree approximation.

Another problem that we have to address, is the lack of global knowledge needed for a spanning tree construction. Most of the existing P2P networks designed for provision of application-level multicast use a bootstrapping process, which returns a list of nodes identified by their IP addresses that are presumed to be online. That is, the initial knowledge of a node is limited to these nodes delivered from the bootstrapping process. In our work we assumed this list to contain between $\log N$ and $\sqrt{N}$ entries, where N is the number of network nodes.

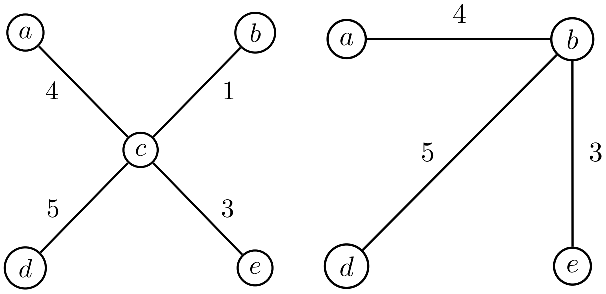
In our CARMIn approach we rely on the NNT principle, where a new node connects to the nearest (in terms of topological distance) known network node.

In our approach we model the Internet as an undirected and connected graph $G = (V, E, w)$. Hereby V stands for the set of vertices v_i, $1 \leq i \leq |V|$, representing network nodes, E is the set of edges $e_{i,j} = \{v_i, v_j\}$ representing the logical connections between nodes, and $w : E \rightarrow \mathbb{N}$ is a weight function assigning a weight to an edge. Generally, the weight function w returns latency time (in milliseconds) of the edge e. However, in this special case it represents a CARMA flavor. On the basis of this graph, we have to create a near-optimal approximation of an MST $T = (V, E_T, w)$ where $E_T \subseteq E$. T is per definition a *connected* graph without *cycles*. In the following we describe the key operations of our approach.

On *joining* (Algorithm 1), the new node first arranges nodes from the bootstrapping set BS depending on their CARMA flavor, and then connects to an arbitrary node with a flavor identifying best network conditions to this node.

Figure 2(a) shows a CARMIn multicast overlay with five nodes. Hereby the nodes b and c belong to the same subrange, the nodes c and e to the same AS, the nodes a and c to the same ASSET, and the nodes c and d belong to the same ASSET link.

```
Algorithm 2: Leave Operation
  Input: node $n$, spanning tree $T = (V, E_T, w)$ and
         neighbor set $N \subseteq V$;
  Output: spanning tree $T' = (V', E'_T, w)$ without $n$;
  begin
      if $degree(n) > 1$ then
          Identify $v \in N$ such that the condition
          $\forall\ v_i \in N : w(v, n) \leq w(v_i, n)$ holds;
          Advise all $v_i \in (N \setminus \{v\})$ to connect to $v_i$;
      $E'_T = E_T$;
      forall the $v_i \in V$ with $\{v_i, n\} \in E_T$ do
          $E'_T = E'_T \setminus \{v_i, n\}$;
      $V' = V \setminus \{n\}$;
```



(a) Before leaving of node c (b) After leaving of node c

Figure 2. CARMIn multicast overlay

On *leaving* (Algorithm 2), the leaving node identifies the node v with the lowest CARMA flavor from all of its neighbors N and then advises all remaining neighbors $N \setminus \{v\}$ to create a connection with v.

Figure 2(b) represents the above depicted CARMIn overlay after the leaving of node c. As proposed, all remaining nodes create connections with the *nearest neighbor* of c which is node b.

These definitions of *join* and *leave* operations ensure that our approximation is *connected* and *cycle-free* (key characteristics of a tree) at any stage of the overlay network construction. However, if a hub node – maintaining a significant number of connections – fails, these characteristics may be violated.

In order to guarantee that our MST approximation always satisfies these characteristics independent of node failures, we introduce a *backup* routine. According to this routine, each node notifies its direct neighbors about its connections and the corresponding connection quality. That is, a network node always knows all of its direct neighbors and all their neighbors including the corresponding CARMA flavors (2-hop neighborhood).

If the node n_f fails, nodes in its neighborhood N will be aware of n_f's nearest node v_i, and thus are able to create connections to v_i in analogy to the regular *leave* operation.

The special feature of the proposed CARMIn approach is the fact, that only the local knowledge of 2-hop neighborhood (no global knowledge) is required for maintaining a multicast infrastructure. Although CARMIn can be used with any other distance estimation or measurement method, it benefits from the utilization of CARMA, since in this way, no additional communication is required for distance estimation. Therefore, CARMIn may be considered as a potential application of the CARMA metric in practice.

We evaluate the quality and cost of our CARMIn MST approximation in Section V-B by comparing it with other multicast approaches.

V. Evaluation Results

In this section we evaluate the quality of the CARMA-based peer selection and of the CARMA-based MST approximation by taking a closer look at the quality of the CARMA distance estimation, and by comparing our CARMA-based multicast approach with other multicast infrastructures.

A. Evaluation of the CARMA-based Peer Selection

The extensive test-runs of CARMA were conducted from a site residing in the customer's address pool of the ISP UkrTelecom for two IPv4 address pools obtained from two of the most popular public trackers of BitTorrent swarms in the Ukraine, namely RuTracker [30] and TorrentsNetUA [31]. These trackers differ significantly in one key aspect, which is important to highlight and validate the CARMA advantages, namely the different percentages of nodes within the same national IX as the vantage point from which the experiments were conducted. From observing the outcome of the "country resolution" feature (which is done by simply querying WHOIS servers for the "country" field) in the popular BitTorrent client μTorrent, we estimate that RuTracker has roughly one-fifth Ukrainian users while TorrentsNetUA harbors about 95% active users from within the Ukrainian exchange point (UA-IX) at any given time. If CARMA is able to confirm such a prevalence, it would be a good sign, prompting the validity of its mathematical model.

Due to established technological and business practices of member ISPs participating in UA-IX, the bandwidth and price for the traffic inside and outside of UA-IX may differ significantly, up to some orders of magnitude. In spite of this, CARMA has large optimization potential. If CARMA-based peer selection rules were to be implemented in, for example, popular BitTorrent clients operating under UA-IX or a similar national Internet setup, far fewer nodes would have to connect outside of their exchange points and many more nodes would be able to choose their peers among those with a more likely higher bandwidth availability.

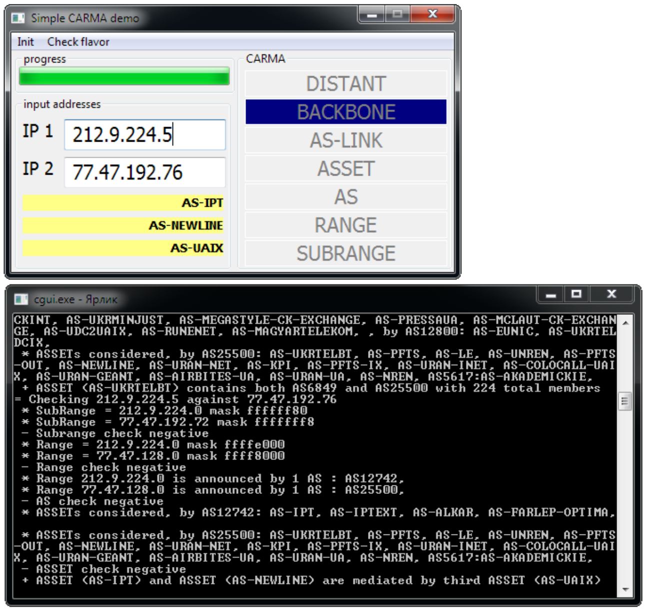
We decided to use a modified ICMP *traceroute* method in our software. It is assumed that the first IPv4 address from a given address pair is the address of the node where the software runs. The software performs a series of special ICMP *traceroute* requests towards the second address. The modified ICMP *traceroute* method differs from the standard version of the *traceroute* tool as follows:

- *ICMP protocol* – much like the Windows version of traceroute ICMP is used rather than UDP, which is common in its GNU counterpart. This is mostly because the primary runtime environments for CARMA are Microsoft Windows x86 and x64, where easy to use ICMP ping functions are part of the programmer-friendly IP Helper API.
- *No DNS queries* – IP addresses of intermediate nodes are not resolved into their host names, neither their IP addresses are indicated, because our evaluation method is only interested in the number of nodes.
- *Speedy and Smart Verification* – unlike the traceroute command-line tool, the reply timeout is set to one second. If the last responded node is not the intended target, or the reply timeout occurs anywhere on the path, the whole query is restarted. This restart can only happen three times. If the target node is not reached on the second and third attempt, the hop count is assumed to be the largest number of intermediate nodes found in all three passes. This effectively eliminates the influence of accidental network lags, which may cause premature ping timeouts of more than one second.

This modified ICMP *traceroute* method was tested from an asymmetric end-user ADSL connection within the UA-IX, characterized by an average response time of around 50 milliseconds from the nodes of its immediate IX neighborhood and of less than 300 milliseconds from the nodes abroad. The tests indicated that an average measurement session for an address pair lasts anywhere from less than half a second, if the target node responds to ICMP requests, to no more than 5 seconds, if it does not respond because of timeout, and no more than 3 seconds, if it does not respond because of reported network or host unreachability. Nevertheless, the sampling of each of the thousand address pairs requires about 1 hour to complete.

The ICMP measurement module was integrated into the CARMA batch-processing software such that for every processed address-pair the real hop count can be measured and written together with the computed CARMA flavor value, unless an ICMP loophole is detected (which was the case for about 2 address pairs per thousand). We consider the gathered statistics later in the course of this section as well.

To obtain a broad spectrum of CARMA affinity flavors as well as hop counts, both trackers were used to gather sampling swarms. However, the volumes of swarms differ significantly: RuTracker was able to produce a swarm of 3610 peers while TorrentsNetUA struggled to achieve 900. The reason for this was that RuTracker features the mandatory enabling of the DHT and *peer exchange (PEX)*



IP 1 212.9.224.5

IP 2 77.47.192.76

AS-IPT

AS-NEWLINE

AS-UAIX

```
CKINT, AS-UKRMINJUST, AS-MEGASTYLE-CK-EXCHANGE, AS-PRESSAUA, AS-MCLAUT-CK-EXCHAN
GE, AS-UDC2UAIX, AS-RUNENET, AS-MAGYARTELEKOM, , by AS12800: AS-EUNIC, AS-UKRTEL
DCIX,
 * ASSETs considered, by AS25500: AS-UKRTELBT, AS-PFTS, AS-LE, AS-UNREN, AS-PFTS
-OUT, AS-NEWLINE, AS-URAN-NET, AS-KPI, AS-PFTS-IX, AS-URAN-INET, AS-COLOCALL-UAI
X, AS-URAN-GEANT, AS-AIRBITES-UA, AS-URAN-UA, AS-NREN, AS5617:AS-AKADEMICKIE,
 + ASSET (AS-UKRTELBT) contains both AS6849 and AS25500 with 224 total members
= Checking 212.9.224.5 against 77.47.192.76
 * SubRange = 212.9.224.0 mask ffffff80
 * SubRange = 77.47.192.72 mask fffffff8
 - Subrange check negative
 * Range = 212.9.224.0 mask ffffe000
 * Range = 77.47.128.0 mask ffff8000
 - Range check negative
 * Range 212.9.224.0 is announced by 1 AS : AS12742,
 * Range 77.47.128.0 is announced by 1 AS : AS25500,
 - AS check negative
 * ASSETs considered, by AS12742: AS-IPT, AS-IPTEXT, AS-ALKAR, AS-FARLEP-OPTIMA,

 * ASSETs considered, by AS25500: AS-UKRTELBT, AS-PFTS, AS-LE, AS-UNREN, AS-PFTS
-OUT, AS-NEWLINE, AS-URAN-NET, AS-KPI, AS-PFTS-IX, AS-URAN-INET, AS-COLOCALL-UAI
X, AS-URAN-GEANT, AS-AIRBITES-UA, AS-URAN-UA, AS-NREN, AS5617:AS-AKADEMICKIE,
 - ASSET check negative
 + ASSET (AS-IPT) and ASSET (AS-NEWLINE) are mediated by third ASSET (AS-UAIX)
```

Figure 3. Screenshot of CARMA concept demonstration tool

mechanisms. Both are features of the BitTorrent protocol [6] allowing a list of new peers from already connected peers to be gotten. DHT and PEX essentially provide a large peer list within a few seconds, instead of the usually limited number of bootstrap peers provided otherwise by tracker alone.

The sampling swarm from RuTracker consisted of 3610 peer nodes of which a small, but notable percentage was believed to belong to the UA-IX address space. TorrentsNetUA provided a sampling swarm of 891 peer nodes of which the vast majority was expected to belong to the UA-IX. Each peer was processed by the CARMA software against the address of its own host machine (also within the UA-IX, see Figure 3), and then the apparent network distance from the host machine to the peer node was measured in terms of a hop count. Figure 3 shows a screenshot of the CARMA concept demonstration tool with a pair of IPv4 addresses as input and their computed flavor with a portion of the calculation logic logfile.

The experiment took about 5 hours to complete. Due to its extended duration, the test-runs were performed in the time period between 01:00 and 06:00 UTC, during which the average Internet traffic volumes generated by the end-users in Russian and Ukrainian segments are at their lowest. This was done to ensure the most favorable conditions for our modified ICMP *traceroute* tool, including the low occurrence of traffic switchovers. We also have observed that nighttime does not cause the numbers of the peers participating in the BitTorrents swarms to drop. This is because the majority of active BitTorrent users either operate so-called seedboxes (dedicated servers customized for BitTorrent) or keep their computers turned on during the night to gain the advantages of the lower nighttime traffic prices.

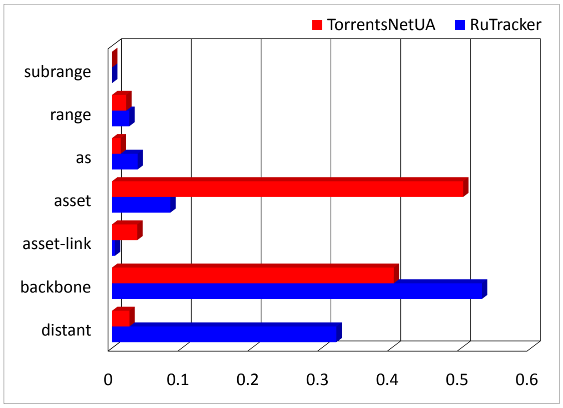


Figure 4. Normalized flavor breakup for the sampling swarms

The preliminary analysis of the obtained data revealed several key insights into the functioning of CARMA. The visible abundance of ASSET-flavored peers is due to the fact that RuTracker historically harbors a large Ukrainian user base. CARMA was able to identify almost 19% of participating users in all tested swarms as Ukrainian, as most of the end-users of Ukrainian ISPs are topologically "under" two major IX points, namely UA-IX and DataGroup-IX [32]. This would trigger the ASSET flavor for most of them if tested by CARMA against a Ukrainian address.

Figure 4 represents the normalized percentage breakup of the sampling swarms obtained from the TorrentsNetUA (upper bar) and RuTracker (lower bar) trackers. First, the TorrentsNetUA flavor breakup in Figure 4 shows that the quantity of distant nodes is 2.5% of the whole swarm space, which is consistent with our predictions. It should be noted however, that in rare occasions this flavor could denote an IPv4 address space registered within the Ukraine and actually operating under UA-IX, if, for some reason, the stored information of its linkage is wrong. Conversely, not all backbone-flavored nodes belong to UA-IX either, as, for example, many Russian IPv4 addresses were flavored as backbone because of an intermediate link between the Ukrainian and Russian major IX points by TeliaSonera AB.

Secondly, the notable low percentage of ASSET-link flavored nodes in both cases may indicate the similarly low likelihood of encountering an arbitrary cross-AS link not mediated by the higher-level ASSET, or a general trend towards building hierarchical routing policies within the national Internet exchange setups, which is consistent with the conclusions drawn in [32].

Also surprising is the fact that none of the nodes had fallen into the subrange flavor. This may have happened due to either the small sample size, or because the subrange announced for the host machine address matched the same range, in which case CARMA chooses the latter.

Meanwhile, our primary goal for this validation was to ensure that the affinity flavor predicted by CARMA corresponds to the topological distance in the network. It is well understood that the nature of these two parameters (flavor and hop count) is completely different and that the numerical representation of the resulting CARMA flavor has no physical meaning unlike the hop count. Still they have to correspond with enough accuracy to prove the effectiveness of CARMA as a traffic-less, purely computational distance estimation metric. To prove the point, we decided to employ two well-known methods of mathematical statistics, such as the chi-square criterion and one-way analysis of variance (abbreviated one-way ANOVA). Strictly speaking, the latter is formally not suitable for analyzing data of digital or otherwise discrete nature, as it was designed for normally distributed data. But we decided to use it anyway due to the significantly large sample size.

However, before calculating the statistical criteria, the results must undergo a sanity test. To give an impression of the nature of the results, the significant portion of them is shown on Figure 5 depicting the accordance between CARMA flavors and hop counts for the RuTracker sampling swarm. The horizontal axes correspond to CARMA flavors (symbolic names) and traceroute hop counts (numeric), while the vertical axis corresponds to the number of occurrences. It is apparent from Figure 5 that within each flavor the hop count distribution is more or less gradual and dome-like (increasing and decreasing slowly along the hop count axis). But at the farthest corner of the graph we see one distinct verge reaching about 100 occurrences for a small hop count with backbone and distant flavors. Figure 6 reveals anomalous occurrences in the hop count flavor distribution: underlined values are those flagged by CARMA as topologically very far while having traceroute hop counts extremely low (2 and 3).

To determine the causes of such anomalies, we conducted manual traceroute runs on the addresses, which yielded specific combinations, such as backbone:3, distant:2 and distant:3. Traceroute unexpectedly stopped at the second and third hop and no subsequent nodes replied at all. Since this behavior was observed only from our vantage point (many LookingGlass servers traced the route to these nodes without any problem), future versions of the CARMA batch processing software [1] should include mechanisms to filter out bogus results caused by temporary malfunctions of ISP routers.

The parameters relevant to the chi-square criterion were automatically calculated by the CARMA batch processing software for both passes, see Table I. As the probability levels for both samples are far below 0.001, we conclude that the correspondence between the predicted CARMA flavor and the actual hop count does exist and is certain. We now proceed to apply the one-way ANOVA method to determine the influence level. We define the influence level as the ratio of the Sum of Squares (SS) between groups and the total SS, see Table II. As can be seen from the table, the correspondences between the chosen parameters (hence, influence levels) are 31.425% and 8.684% for RuTracker and TorrentsNetUA, respectively. The lower influence level for the sample obtained from TorrentsNetUA swarm could be explained by its topologically constrained nature, in the sense that distant flavored nodes were much less frequent.

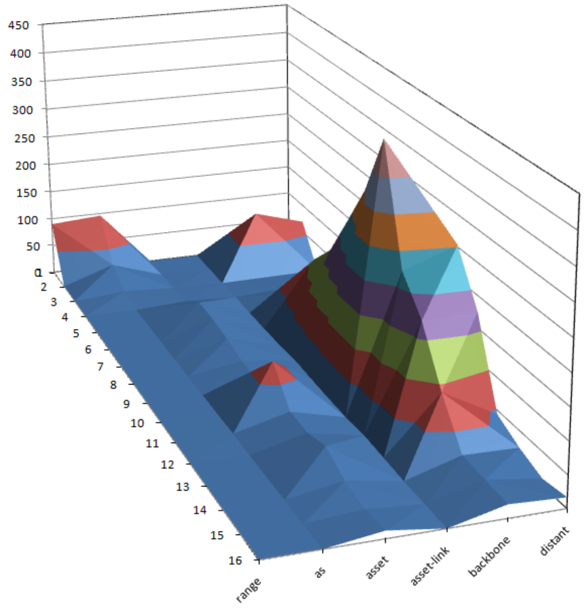
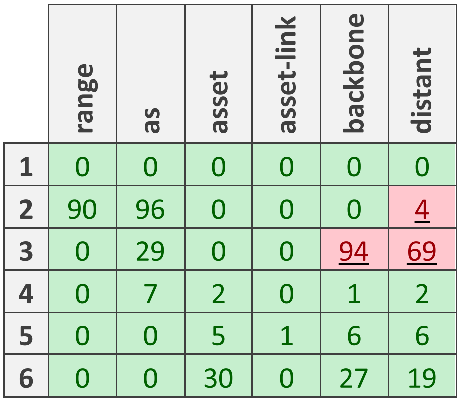


Figure 5. Flavors and hop counts of the RuTracker swarm



	range	as	asset	asset-link	backbone	distant
1	0	0	0	0	0	0
2	90	96	0	0	0	4
3	0	29	0	0	94	69
4	0	7	2	0	1	2
5	0	0	5	1	6	6
6	0	0	30	0	27	19

Figure 6. Anomalous occurrences in the hop count flavor distribution

	RuTracker	TorrentsNetUA
Degrees of freedom (d)	120	75
chi-square (χ^2)	3722.86	413.20
Probability (p)	$p < 0.001$	$p < 0.001$

Table I
STATISTICAL PARAMETERS RELEVANT TO THE CHI-SQUARE CRITERION

	RuTracker	TorrentsNetUA
Range variance	0	0.941
AS variance	1.029	2.083
ASSET variance	9.949	1.585
ASSET-Link variance	6.466	4.933
Backbone variance	8.576	3.802
Distant variance	10.937	0.177
SS between groups	14785.941	215.419
SS among groups	32265.495	2265.020
SS total	47051.436	2480.439
Level of influence	31.425%	8.684%

Table II
STATISTICAL PARAMETERS RELEVANT TO VARIANCE ANALYSIS

Destination	Flavor	Hops	Ping [ms]
143.93.54.111	1	3	4
136.199.199.105	2	9	7
131.246.120.51	3	11	8
82.165.77.114	3	13	12
143.169.9.245	3	15	-
193.1.101.61	3	16	39
193.232.113.151	3	17	47
141.20.5.188	3	17	**26**
163.1.13.189	3	20	**24**
130.92.253.230	3	22	-
131.180.77.26	5	12	**17**
217.21.43.11	5	18	51
169.229.131.81	5	26	**166**
131.130.70.8	6	14	-
217.173.193.11	6	15	-
77.47.133.2	6	16	53
128.112.132.86	6	20	106

Table III
CARMA VS. PING COMPARISON

However, both levels of influence are quite optimistic.

In order to get an impression of the value of the CARMA distance estimation method returning a distance flavor and the corresponding hop count, we compared it with the results of the ping method returning the round trip time of a message in milliseconds. Therefore, we first measured the CARMA distance (flavor and hop count) between the University of Applied Sciences in Trier (Germany) and 17 other universities. Then we sorted these results in ascending order. Afterwards we measured the ping distance to these IP addresses. As the results in Table III show, for 4 of 17 positions (less than 25%) the ping order differs from the CARMA order. Based on these results we claim that CARMA provides a feasible approximation for node distances in the Internet. Moreover, as the results show, 4 of 17 IP addresses were not accessible by the ping method. This behavior indicates a significant drawback of the ping and traceroute methods already described in Section II: Some ISPs are filtering those requests. Therefore ping or traceroute requests cannot guarantee that a valid result will be returned. This fact may be considered as a definite advantage of CARMA compared to ping and traceroute, since CARMA always returns a result.

It is understood that the results obtained in this set of the experiments are rather preliminary and are somewhat lacking the concrete proof of explicit performance improvement in case CARMA is implemented in BitTorrent client software. The purpose of this paper, however, is to evaluate the feasibility of the CARMA model in P2P applications in general by comparing it with traditional network distance metrics.

B. Quality and Cost of the CARMIn MST Approximation

The problem of minimizing communication costs can be reduced to the problem of finding a *Minimum Communication Cost Spanning Tree* (MCT) [33][34] known to be NP-hard. This problem is formalized in [35] as follows: having a set of peers $V = \{v_1, \dots, v_n\}$ there is a matrix $R_{n\times n} = (r_{i,j})$ of communication requirements where $r_{i,j}$ represents the expected communication from v_i to v_j. The distances between peers are stored in a distance matrix $W_{n\times n} = (w_{i,j})$, where $w_{i,j}$ represents the latency time for sending a message from v_i to v_j. [35] denotes the distance $dist(v_i, v_j, G)$ between two arbitrary nodes v_i and v_j in a network graph G as the minimum sum of the edge weights from W along any path connecting v_i and v_j in G. For every two peers v_i and v_j a spanning tree $T = (V, E_T, w)$ ($E_T \subseteq E$) contains a unique path of length $dist(v_i, v_j, T)$. The communication cost over the network tree is defined as:

$$C(T) = \sum_{i,j} r_{i,j} \cdot dist(v_i, v_j, T)$$

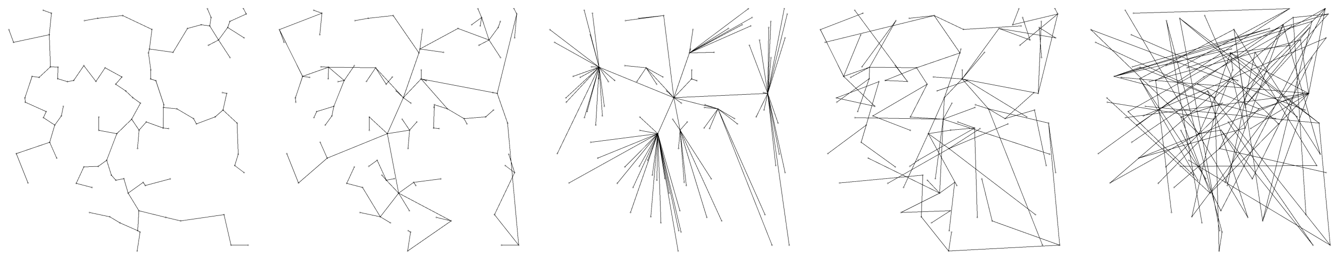
The algorithm proposed in [35] guarantees a $O(\log^2 |V|)$ approximation of the considered problem.

However, in a P2P system we cannot exactly specify the expected amount of communication $r_{i,j}$ between two arbitrary nodes v_i and v_j. By assuming that $r_{i,j} = 1$, $\forall\ 1 \leq i,\ j \leq |V|$, [36] proposes the reduction of multicast costs by using an MST approximation. Hereby the multicast cost $C(E_T)$ is denoted as the cost for propagating a message to all recipients in the group, which is the sum of all edge weights in the tree representing latency delays along any path taken by the message:

$$C(E_T) = \sum_{e \in E_T} w(e)$$

We use the $C(E_T)$ metric as the quality function for the comparison of different approaches.

In order to provide meaningful results, we compare our CARMIn approach with some of the existing P2P approaches supporting application-level multicast such as ALMI, JXTA, and HiOPS [36]. To extend the range of our evaluation, we have also considered a *RANDOM* infrastructure, where a new node builds up connections to a randomly selected node from the bootstrapping set. Figure 7 shows MST approximations by using the above mentioned approaches in networks with 100 nodes.



(a) ALMI (b) JXTA (c) HiOPS (d) CARMA (e) RANDOM

Figure 7. MST approximations in networks with 100 nodes

To compare these networks with respect to the communication cost $C(E_T)$, we set up a simple simulation environment. Using this environment, we can create an arbitrary number of network nodes, interconnect them according to a given algorithm and then compute the communication cost metric $C(E_T)$. We have performed several evaluation runs where we randomly created 10, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, and 5000 nodes interconnecting them with ALMI, JXTA, HiOPS, CARMIn and RANDOM infrastructures. After each run, we computed the communication cost $C(E_T)$ in milliseconds needed for the propagation of a multicast message to all existing nodes.

As Figure 8 shows, the JXTA and ALMI approaches relying on global knowledge do provide low $C(E_T)$ values. But as mentioned before, these do not scale in terms of a large number of users. As expected, the RANDOM infrastructure incurs the highest communication cost. The scalable CARMIn approach provides nearly the same $C(E_T)$ values as does the HiOPS overlay, but relies only on local knowledge as does the RANDOM infrastructure, by this means providing a good trade-off between construction and communication costs. The binning approach [24] would show almost the same behavior in terms of the communications cost as the CARMIn approach. However, CARMIn does not require any additional communication for ordering the nodes in the bootstrapping set, whereas nodes following the binning approach have to contact the landmark servers.

We denote the metric describing the knowledge i.e., the number of nodes, which should be known by a new node to join the multicast infrastructure, as $K(V)$. In order to construct an MST, ALMI requires global knowledge of all involved nodes $K(V) = |V|$. A new JXTA node requires the same amount of knowledge $K(V) = |V|$ in order to identify the nearest node. In HiOPS and CARMIn the amount of knowledge is variable and depends on the initial settings. For our comparison in [1] we have used bootstrapping lists for CAMRA and HiOPS with up to $K(V) = \sqrt{|V|}$ nodes. Only the RANDOM infrastructure does not require any global knowledge. Here it is enough to know only one node. Figure 9 represents the respective amounts of knowledge.

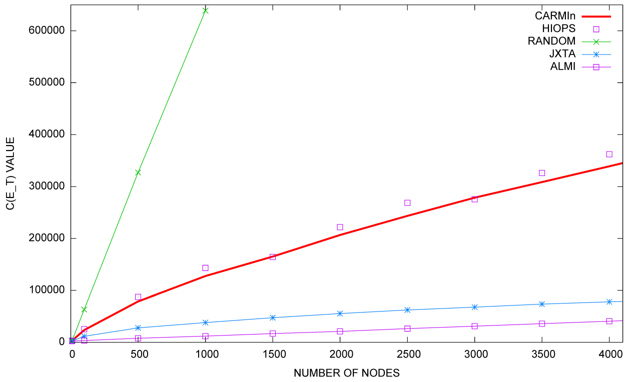


Figure 8. Communication cost $C(E_T)$

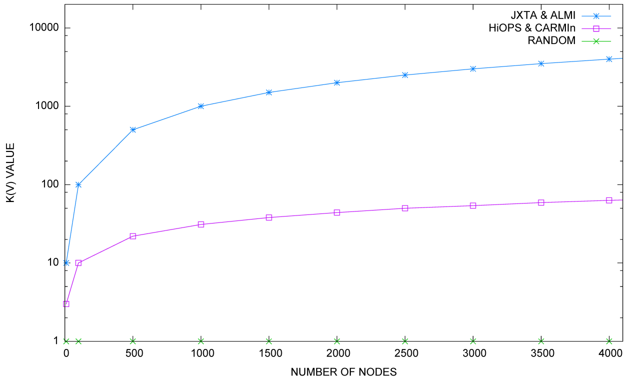


Figure 9. Amount of knowledge K(V)

We denote the metric describing the running time complexity i.e., the construction cost of any multicast tree as $O(T)$. The running time complexity for the MST construction is dominated by sorting of edges i.e., node distances [37]. The sorting complexity is given by

$$\begin{aligned} O(T) &= O(\log|E||E|) = O(\log|V|^2|E|) \\ &= O(2\log|V||E|) = O(\log|V||E|). \end{aligned}$$

Because of this sorting complexity the same construction cost is needed for construction of the JXTA-NNT. As discussed in [1], in HiOPS only rendezvous nodes ($|V_R| = \sqrt{|V|}$) are involved in the MST construction. Assuming that the implication $|V_R| = \sqrt{|V|} \Rightarrow |E_R| = \sqrt{|E|}$

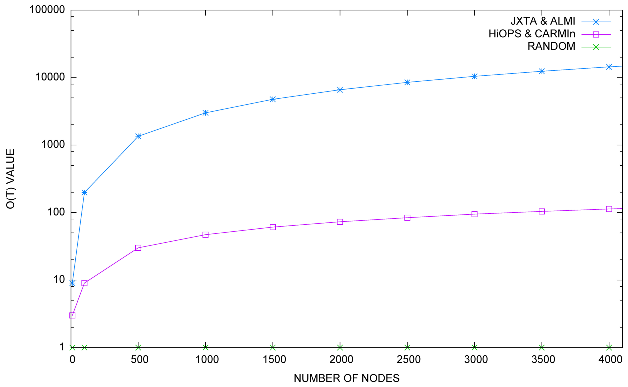


Figure 10. Construction cost O(T)

holds, the construction cost of the HiOPS infrastructure equates to $O(T) = O(\log \sqrt{|V|}\sqrt{|E|})$. According to the CARMIn approach, the nodes from the bootstrapping list ($\sqrt{V}$) have to be sorted depending on their CARMA distance. Thus, the construction cost here corresponds to $O(T) = O(\log \sqrt{|V|}\sqrt{|E|})$ due to sorting complexity too. The construction of the RANDOM infrastructure does not require any nameable construction cost ($O(T) = O(1)$). Figure 10 compares the considered multicast approaches with respect to required construction costs. As shown by this figure, the construction cost of ALMI and JXTA infrastructures is simply too high.

As the evaluation results show, a clear performance improvement of CARMIn over the RANDOM approach is observed. With respect to other algorithms computing an optimal MST (ALMI) or relying on global knowledge (JXTA), the simplicity of the CARMIn approach (lower construction cost, local knowledge) is the advantage, but the multicast cost deteriorates. Moreover CARMIn does provide a more scalable solution than HiOPS, since due to the utilization of CARMA it does not require any additional communication. Therefore, in large-scale application scenarios, which cannot rely on global knowledge and do not have exact information about node distances, we would prefer our CARMIn approach to all other considered approaches.

VI. Conclusion and Future Work

By design, CARMA is meant to be dynamically changing as the communication goes on, reflecting and adapting to the changes in bandwidth conditions. The life-cycle of a CARMA-capable node in a P2P network starts with the downloading of the most recent IP and AS allocation databases from all regional Internet registries and compiling them into an easily indexable internal format. This may take tens of minutes to complete, depending on the CPU speed and bandwidth. Although the RIR databases are updated daily, their growth rate is rather low. Therefore, the startup sequence to refresh the data may be called less frequently than once a day.

In this paper we also have partially addressed the second layer of CARMA, leaving active measurements of bandwidth as well as the integration of CARMA into P2P client software for future publications. Under the assumption that the traceroute hop count represents, to a certain degree, a real topological distance, an experimental validation indicated that the correspondence between predicted flavors and actual topological distances exists and is significant.

The most obvious and straightforward leveraging mechanism for CARMA is peer list reordering. As mentioned in Section IV-A, a P2P client starts to actively request the downloading of a file upon receiving a list of peers who had earlier indicated the possession of the desired content. In all of the P2P clients that we analyzed, either no precedence is given to any peer from the list or it has nothing to do with topological affinity. In fact, in many practical scenarios, even not all peers from the list are queried until the downloading is stopped. However, if the peer list is very large and diverse enough in terms of topological distance and bandwidth conditions, the corresponding precedence mechanism can ensure a significant burst of performance by choosing peers that are likely (according to their CARMA flavor) to provide higher transfer speeds and lower latency.

Therefore, in our future work we would like to address the complete second and third layers of CARMA, calculated by direct measurements involving additional traffic. These layers may be expressed as weighted scores by which all peer priorities are then fine-tuned within the boundaries of their respective first-layer flavors. It should be noted that an implementation of the second CARMA layer will require modifications to the existing software, and that the third CARMA layer will require extensions to existing protocols in order to have any impact on the performance. In this case, the life-cycle of a CARMA-capable node is extended to include the following steps after the initial startup and peer list ordering based on the first-layer flavors:

1) an additional check is performed using the second layer of CARMA in such a way that the original order is not substituted, but rather fine-tuned;
2) at this point, the actual communication to remote parties is initiated; if connections are setup using a CARMA-enabled protocol, the third layer is utilized by the parties providing bandwidth conditions and related information to each other; using this information, peer lists may once again be reordered placing less-loaded nodes at higher positions.

If a peer exchange mechanism is enabled, newly reported nodes must go through all layers of CARMA in order to be placed in the peer list.

We plan to demonstrate the effectiveness of the CARMA approach by performing extensive experiments using the set of BitTorrent client software with plugin support that allow peer ordering to be manipulated. If this proves to be effective, we plan to integrate CARMA into real-life P2P networks. We are currently developing a software library, implementing CARMA under the LGPL license to assist software engineers wishing to optimize the performance of their P2P applications.

REFERENCES

[1] G. Poryev, H. Schloss, and R. Oechsle, "CARMA Based MST Approximation for Multicast Provision in P2P Networks," in *Proceedings of the 2010 Sixth International Conference on Networking and Services (ICNS '10)*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 123–128.

[2] S. Daniel, J. Ellis, and T. Truscott, "USENET - A General Access UNIX at Network," 1980.

[3] Napster-Homepage, Access date: January 2011. [Online]. Available: www.napster.com

[4] M. Jovanovic, F. Annexstein, and K. Berman, "Scalability Issues in Large Peer-to-Peer Networks - a Case Study of Gnutella," 2001. [Online]. Available: citeseer.ist.psu.edu/jovanovic01scalability.html

[5] eDonkey2000 Homepage, Access date: January 2011. [Online]. Available: http://tinyurl.com/ed2klink

[6] B. Cohen, "Incentives Build Robustness in BitTorrent," in *the First Workshop on the Economics of Peer-to-Peer Systems*, 2003.

[7] R. Bustos, A. Aguilar, K. Makki, and R. K. Ege, "Multicast-P2P Content Distribution in Large-Scale Enterprise Networks," in *IEEE Symposium on Computers and Communications (ISCC 2008)*, 2008, pp. 487–494.

[8] A. Feldmann and V. Aggarwal, "ISP-Aided Neighbor Selection in P2P Systems," in *IETF P2P Infrastructure Workshop (P2Pi)*, Berlin, Germany, May 2008.

[9] D. R. Choffnes and F. E. Bustamante, "Taming the Torrent: a Practical Approach to Reducing Cross-isp Traffic in Peer-to-Peer Systems," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 363–374, 2008.

[10] H. Xie, A. Krishnamurthy, A. Silberschatz, and R. Y. Yang, "P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers," DCIA P2P MARKET CONFERENCE, 2008.

[11] J. Kubiatowicz, "Extracting Guarantees from Chaos," *Commun. ACM*, vol. 46, no. 2, pp. 33–38, 2003.

[12] G. Lucas, A. Ghose, and J. Chuang, "On Characterizing Affinity and its Impact on Network Performance," in *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research (MoMeTools '03)*. New York, NY, USA: ACM, 2003, pp. 65–75.

[13] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding Traceroute Anomalies with Paris Traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06)*. New York, NY, USA: ACM, 2006, pp. 153–158.

[14] H. Madhyastha, T. Isdal, M. Piatek, and C. Dixon, "iPlane: An Information Plane for Distributed Services," in *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*. USENIX, 2006, pp. 367–380.

[15] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution?" in *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement (IMC '05)*. New York, NY, USA: ACM, 2005, pp. 63–76.

[16] T. Jiang and A. Zhong, "A Multicast Routing Algorithm for P2P Networks," in *GCC 2003*, 2003, pp. 452–455.

[17] T. Peng, Q. Zheng, and Y. Jin, "Transmission Latency based Network Friendly Tree for Peer-to-Peer Streaming," *j-jucs*, vol. 15, no. 9, pp. 2011–2025, 2009, http://www.jucs.org/jucs_15_9/transmission_latency_based_network.

[18] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: an Application Level Multicast Infrastructure," in *Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems (USITS'01)*. Berkeley, CA, USA: USENIX Association, 2001, pp. 5–5.

[19] K. Radab and A. U. Haque, "A Minimum Spanning Tree Algorithm for Efficient P2P Video Streaming System," in *The 12th International Conference on Advanced Communication Technology (ICACT 2010), 2010*, 2010, pp. 93 – 97.

[20] D. Peleg and V. Rubinovich, "A Near-Tight Lower Bound on the Time Complexity of Distributed Minimum-Weight Spanning Tree Construction," *SIAM J. Comput.*, vol. 30, no. 5, pp. 1427–1442, 2000.

[21] M. Khan and G. Pandurangan, "A Fast Distributed Approximation Algorithm for Minimum Spanning Trees," *Distributed Computing*, vol. 20, no. 6, pp. 391–402, April 2008. [Online]. Available: http://dx.doi.org/10.1007/s00446-007-0047-8

[22] JXTA-Homepage, Access date: January 2011. [Online]. Available: https://jxta.dev.java.net

[23] E. Michael, "Distributed Approximation: a Survey," *SIGACT News*, vol. 35, no. 4, pp. 40–57, 2004.

[24] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware Overlay Construction and Server Selection," in *Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, vol. 3, 2002, pp. 1190– 1199 vol.

[25] J. J. D. Mol, D. H. J. Epema, and H. J. Sips, "The Orchard Algorithm: P2P Multicasting without Free-Riding," in *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 275–282. [Online]. Available: http://portal.acm.org/citation.cfm?id=1157740.1158270

[26] J. Hawkinson and T. Bates, "Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)," RFC 1930 (Best Current Practice), Internet Engineering Task Force, mar 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1930.txt

[27] P. Maymounkov and D. Mazieres, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” 2002, http://www.cs.rice.edu/Conferences/IPTPS02/109.pdf.

[28] eMule Homepage, Access date: January 2011. [Online]. Available: http://www.emule-project.net

[29] G. Poryev, T. Rudyk, and O. Sulima, “Traffic Regulation and Reputation Handling in the BitTorrent Peer-to-Peer Networks,” National Technical University of Ukraine “KPI”, Tech. Rep., 2008.

[30] RuTracker-Homepage, Access date: January 2011. [Online]. Available: http://rutracker.org

[31] TorrentsNetUA-Homepage, Access date: January 2011. [Online]. Available: http://torrents.net.ua

[32] V. Furashev, V. Zubok, and D. Lande, “Parameters of the Ukrainian Internet segment as a complex network,” *in Proceedings of the Open Informatics and Computer Technologies*, vol. 40, pp. 235–242, 2008.

[33] P. Crescenzi and V. Kann, “A compendium of NP Optimization Problems.” [Online]. Available: http://www.nada.kth.se/~viggo/wwwcompendium/node77.html#4555

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

[35] D. Peleg and E. Reshef, “Deterministic Polylog Approximation for Minimum Communication Spanning Trees,” in *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP ’98)*. London, UK: Springer-Verlag, 1998, pp. 670–681.

[36] H. Schloss, R. Oechsle, J. Botev, M. Esch, A. Hoehfeld, and I. Scholtes, “HiOPS Overlay - Efficient Provision of Multicast in Peer-to-Peer Systems,” in *16th IEEE International Conference on Networks (ICON 2008)*, New Delhi, India, 2008, pp. 1–6.

[37] J. B. Kruskal, “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem,” *Proc. Am. Math. Soc.*, vol. 7, pp. 48–50, 1956.