HAL
open science

Towards a Fully Distributed and Decentralized Communication Service for Heterogeneous Networks

Mugurel Ionut Andreica

To cite this version:

Mugurel Ionut Andreica. Towards a Fully Distributed and Decentralized Communication Service for Heterogeneous Networks. E. Andronescu, C. Burileanu. Advances in Engineering: from Theory to Application, Politehnica Press Publishing House, pp.5-12, 2012, 978-606-515-381-3. hal-00805467

HAL Id: hal-00805467

https://hal.science/hal-00805467v1

Submitted on 27 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards a Fully Distributed and Decentralized Communication Service for Heterogeneous Networks

Mugurel I. Andreica, *Member, IEEE*

***Abstract*—This paper presents the postdoctoral research project of the author, which consists of designing and developing a fully distributed and decentralized communication service for heterogeneous networks. The main topics addressed by the research project are presented and for each topic the motivation, challenges and the results obtained so far are discussed.**

***Index Terms*—communication optimization, multicast, peer-to-peer.**

I. Introduction

The topic of communication optimization in distributed systems is very important nowadays and, thus, presents a special interest to the international scientific community. A distributed system is a complex ensemble of geographically distributed components which collaborate in order to solve several problems or in order to provide multiple types of services. Although the design and implementation of a distributed system is an activity belonging to the field of engineering sciences (of Computer and Systems Engineering in particular), the usage of a distributed system exceeds the boundaries of this field. Nowadays, every field of activity uses at least one (type of) distributed system, in order to collaborate, communicate with other partners, store, search and deliver data, perform economic, industrial or research activities. The requirements regarding the functionality, performance and interface with the outside for a distributed system are determined by the system's users, who may belong to various domains.

The postdoctoral research project approaches the topic of communication optimization in distributed systems from the perspective of three complementary levels (see Fig. 1):

- the level of the (multi)point-to-(multi)point message transfer algorithms and protocols
- the level of the communication topology of a distributed system
- the level of the communication services based on a communication topology, like multicast data transfer services or services for storing, searching and delivering data flow objects

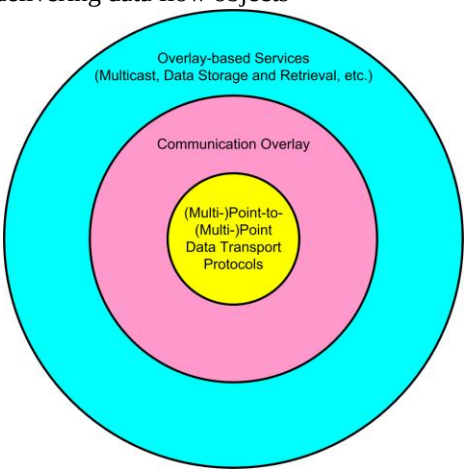

Figure 1. Complementary levels at which the communication service operates.

Each level provides a set of functionalities which are used by the next level. The communication topology of a distributed system is composed of nodes and point-to-point connections between these nodes (these connections are based on point-to-point data transport algorithms and protocols). Advanced communication services, like multicast and the storage, search and delivery of data flow objects, are based on a communication topology within which the messages are transmitted.

Sections II-V present the state-of-the-art in the fields of interest of the project, together with the motivation for performing research activities in the corresponding field and a brief summary of the results obtained so far. Section VI briefly presents other research domains related to data transfer optimization, which are of contextual interest to the project. Section VII concludes and presents future work.

II. (Multi)point-to-(Multi)point Data Transfer Algorithms and Protocols

The data transport protocols UDP and TCP are the most commonly used protocols in distributed systems (of any size) in order to achieve point-to-point communication (between two nodes of the system).

Manuscript received October 1, 2011. This work was supported by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the financial agreement POSDRU/89/1.5/S/62557.

M. I. Andreica is a post-doctoral researcher with the Politehnica University of Bucharest, Bucharest, Romania, Splaiul Independentei 313, postal code: 060042 (e-mail: mugurel.andreica@cs.pub.ro).

The UDP protocol allows the transmission of messages between two nodes, but provides no guarantees regarding the arrival of the message to the destination and does not implement any algorithm for limiting the transfer speed in order to avoid congesting the network links.

The TCP protocol allows the transfer of a stream of bytes, is reliable (it guarantees that messages reach the destination) and implements a congestion control algorithm known as AIMD (Additive Increase Multiplicative Decrease). However, this congestion control algorithm does not allow TCP to use the available network bandwidth appropriately in the case of network links with high latency and high bandwidth [39].

Thus, the UDP protocol has the disadvantage of being too aggressive in certain situations, while the TCP protocol is not aggressive enough. Based on these properties of the UDP and TCP protocols, the research in the field of point-to-point communication algorithms and protocols aimed at the development of new point-to-point data transport algorithms and protocols which have a better transfer speed than TCP in the situations in which TCP behaves unsatisfactorily, but which present TCP's sensibility when network congestion occurs. These protocols can be differentiated according to their behavior towards UDP and TCP (particularly towards TCP):

1. ***fair behavior*** towards TCP flows (they share the available network bandwidth fairly with competing TCP flows, but may be able to use supplementary bandwidth, which the TCP flows are not capable of using)
2. ***prioritary behavior*** towards TCP flows (they have a more aggressive behavior towards competing TCP flows than a standard TCP flow would have)
3. ***subprioritary behavior*** (they use only the available network bandwidth which is not used by the competing TCP and UDP flows)

A. Point-to-point Communication Algorithms and Protocols with a Fair Behavior towards TCP

Category 1 (that of the communication algorithms and protocols having a fair behavior towards TCP) contains the largest number of examples. A great part of these is based on modifying the congestion control algorithm used by the standard TCP protocol: Scalable TCP [23], HighSpeed TCP, FAST TCP [20], TCP Cubic [29]. Application-level data transport protocols based on using the UDP protocol have also been proposed (basically, a congestion control component has been added on top of the UDP protocol): RBUDP, SABUL [15] and UDT [16].

A simple method for increasing the transfer speed of the TCP protocol is to use multiple parallel TCP connections between the same source and destination.

A protocol similar to TCP, which solves many of TCP's problems, is SCTP. SCTP uses internally multiple independent streams on which messages are being transmitted. From a certain point of view, SCTP is similar to using multiple parallel connections, but the cost (of time and memory) of using *N* SCTP flows is lower than that of using *N* parallel TCP connections.

B. Point-to-point Communication Algorithms and Protocols with Prioritary Behavior towards TCP

The solutions from this second category are more rare, because they cannot be deployed on a large scale, in the Internet, without producing a negative impact on existing applications, in which the TCP protocol is predominant. Relentless TCP [27] is such an example, also based on modifying TCP's congestion control algorithm. This time, however, the modification generates a more aggressive behavior than that of a standard TCP flow.

C. Point-to-point Communication Algorithms and Protocols with Subprioritary Behavior towards TCP

The algorithms and protocols from the third category are generically called *less-than-best-effort* (in the conditions in which UDP and TCP flows are considered to be *best-effort*). TCP Vegas [8] reduces its transfer speed in the presence of a standard TCP flow and was designed with the purpose of achieving a larger transfer speed that standard TCP. Other transport protocols designed with the purpose of not obstructing TCP flows are TCP-LP [24], TCP Nice and 4CP [26]. LEDBAT is a new congestion control algorithm proposed by IETF, which uses linear control rules for updating the size of the congestion window. All the point-to-point algorithms and protocols mentioned so far (with the exception of 4CP) use both packet loss and delay (one-way or round-trip time) as metrics for updating the size of the congestion window.

D. Unsolved Problems

With only a few exceptions, the communication protocols having a fair behavior towards TCP have beed designed considering only a single data flow and a sequential processing of their specific events. SCTP is an exception which alows the transfer of data from one node to another on multiple concurrent streams and, with some modifications [19], even on multiple network paths. However, the data flows are independent, maintaining congestion control information for each flow. Under these circumstances, the following problems have not been satisfactorily solved so far:

- parallel transfer of data (on multiple streams), under the circumstances of maintaining a single set of information for all the flows (correlated flows), and
- the development of a communication protocol which processes different events in parallel (e.g. packet loss detection, arrival of notification messages, and so on)

Maintaining a common set of information for all the parallel flows (between the same pair of nodes) could allow, for instance, the reduction of the transfer rate of one flow and the increase of the transfer rate on a different flow. A novel approach which has not been considered so far is to maintain correlated (multi)point-to-(multi)point flows (e.g. between a source node and multiple destinations).

In the case of protocols with a subprioritary behavior, the two objectives, of not influencing negatively the competing flows (TCP or UDP) and of using as much as possible the available network bandwidth are partially contradictory. All the existing solutions fail to achieve at least one of the following goals:

- to be totally non-aggressive towards TCP flows
- to use all the available network bandwidth (which is not used by competing flows)
- to quickly respond to changes in the network bandwidth of the competing flows (e.g. when a competing flow drops or increases its bandwidth, a new flow comes up or an old flow ends)
- to behave fairly towards other flows of the same type (e.g., in [30], a fairness problem regarding the transfer speeds of two concurrent LEDBAT flows has been noticed)

Except for the three main categories of point-to-point communication algorithms and protocols that were mentioned, within each category multiple levels of priority can be defined between the flows belonging to that category. None of the existing solutions considers the possibility of assigning different priorities to flows belonging to the same category.

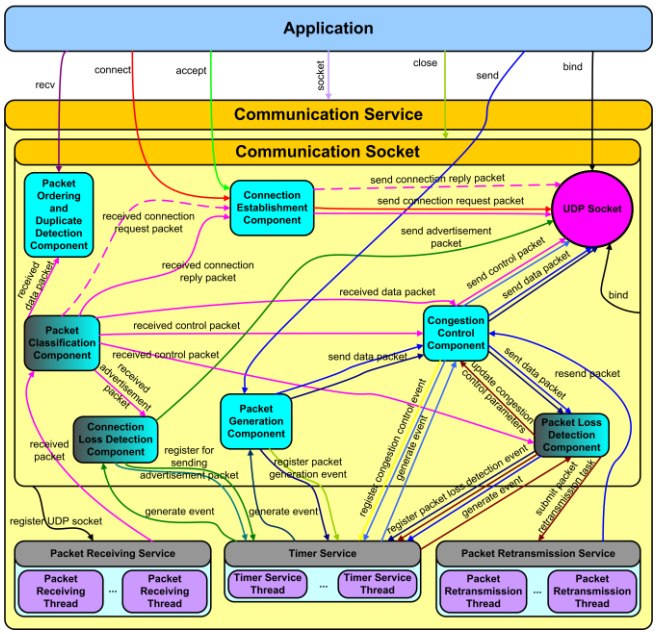

Figure 2. Components of the Python-based communication protocol development framework and interactions between them.

E. Motivation for Performing Research Activities

Point-to-point data transfer is the main service which is at the base of every distributed system and every communication network. The optimization and extension of this type of service may lead to the development of new communication services and applications, with positive impact in many activity domains.

The development of new techniques for parallel data transfer on multiple correlated flows and for parallel processing of the specific events of a communication protocol/service will lead to the improvement of (multi)point-to-(multi)point data transfers and, thus, to a better usage of the available network resources.

The development of efficient algorithms for providing *less-than-best-effort* data transfer services and for including (numerical) priorities within the same priority class will allow a better differentiation of (multi)point-to-(multi)point data transfer services according to their priority and a better usage of the existing network resources.

F. Obtained Research Results

A framework for designing and developing (prototype) communication protocols in the Python programming language has been proposed (see Fig. 2). Based on this framework, a "less-than-best-effort" communication protocol (i.e. with a lower priority than TCP) has been developed. The protocol is a latency-based protocol and employs congestion control by adapting its inter-packet-sending time in a manner based on dynamic binary search.

Novel methods of designing the communication module and loosely coupling it with the request processing module for distributed services have also been devised (see Fig. 3 for several examples regarding the TCP protocol).

III. Communication Topologies

The communication topology of a distributed system is concerned with the way the system's nodes are interconnected. Obviously, in order for the nodes of a distributed system to provide a service or fulfill the system's objectives (e.g. solve certain problems), they must communicate. Communication between two nodes can be performed:

- ***directly*** between them or
- ***through other intermediary nodes*** located within the system.

The systems in which two nodes communicate only directly between them (when they wish to communicate), have several disadvantages, at least regarding the following aspects:

- if one of the nodes is located behined a NAT (Network Address Translation) or firewall device, then the other node may not succeed to initiate the communication towards this node
- opening of a new connection (e.g. TCP connection) for every conversation between two nodes generates a high overhead compared to the situation in which a connection is reused for multiple conversations
- a node cannot have open connections with too many nodes simultaneously, because of the limitations of the protocols and operating systems (e.g. a maximum number of socket descriptors or a maximum number of available ports)
- transferring data over a long-distance TCP connection may be more inefficient than transferring it by using multiple intermediate connections [25]
- multicast communication services cannot be provided efficiently because: the sources should know the identity of all the destinations (which can potentially be in large numbers), and the network bandwidth

consumed by independent data transfers towards every destination would be very high

Because of these reasons, the construction and maintenance of structured communication topologies presents significant advantages in many situations. A communication topology can be modeled as a graph, in which the vertices are represented by the system's nodes, and the edges are represented by the point-to-point connections between the nodes. These connections can be logical or they may be, for instance, TCP connections, SCTP associations, or connections specific to any other protocol or point-to-point data transfer service.

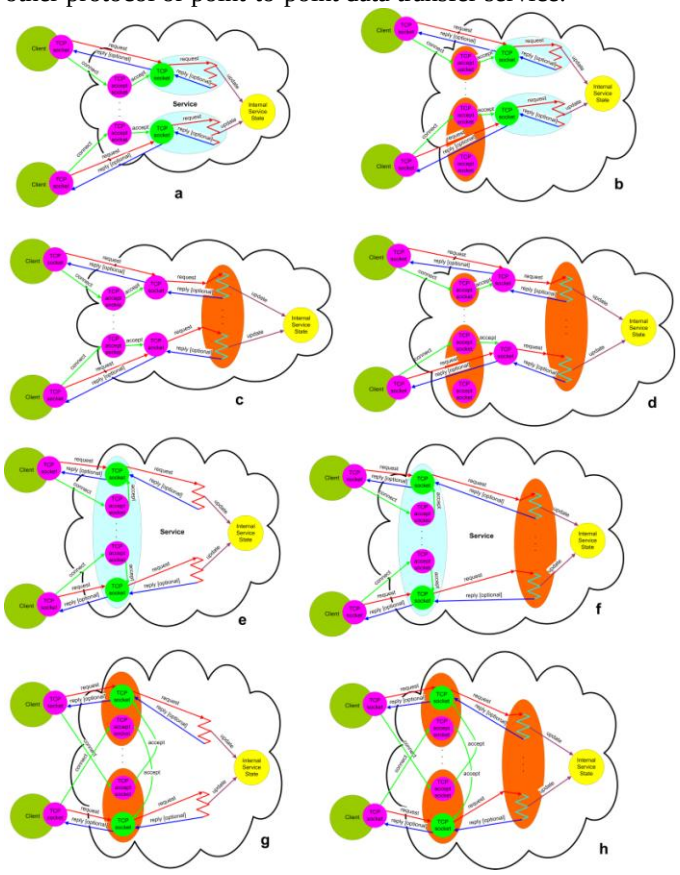

Figure 3. TCP-based service communication. a) 1 accept thread and 1 thread per socket for reading, processing requests and writing; b) thread pool for accepting connections and 1 thread per socket for reading, processing requests and writing; c) 1 accept thread, 1 thread per socket for reading and thread pool(s) for processing and writing; d) thread pool for accepting connections, 1 thread per socket for reading and thread pool(s) for processing and writing; e) 1 thread for accepting and reading requests from all the sockets and 1 thread per socket for processing and writing; f) 1 thread for accepting and reading requests from all the sockets and thread pool(s) for processing and writing; g) a thread pool for accepting and reading requests from all the sockets and 1 thread per socket for processing and writing; h) a thread pool for accepting and reading requests from all the sockets and thread pool(s) for processing and writing.

An important classification of communication topologies [21] considers the following two categories: ***unstructured*** and ***structured***.

A. Unstructured Topologies

The categoy of unstructured topologies contains some of the first file sharing peer-to-peer systems, like Gnutella [21]. Their topologies had a *totally decentralized* structure. Thus, the topology itself was scalable, but the application functions provided on top of the topology (e.g. file search) were not.

Systems like Bittorrent or Kazaa [13] appeared later, which improved both the search and data transfer process. All the Bittorrent nodes downloading the same file belong to the same group (named *swarm*). In order to search for a file, a node must contact a central tracker, from which it knows which nodes store the desired file. Kazaa nodes are differentiated into clients and super-peers. Super-peers are chosen automatically based on their computing power, storage capacity and their network bandwidth. Clients connect to the closest super-peer, through which they search and download files.

B. Structured Topologies

The first generation of distributed systems with a *structured*, *fully decentralized* and *scalable* topology, was represented by Distributed Hash Tables (DHT). Each node of a DHT has a unique identifier in a virtual coordinate space. The nodes connect into a structured topology based on these identifiers. CAN, Chord, Kademlia, Viceroy and Pastry [21] were the initial DHTs which determined the appearance of a new research sub-domain. They were later followed by several other DHTs with similar properties.

The DHTs form structured and scalable communication topologies, in which every message passes through a small number of intermediate nodes. However, they present the following disadvantages:

- they require connectivity capabilities between any pair of nodes, so they cannot include nodes located behind NATs and firewalls
- nodes which are close in the overlay network may be very far geographically; thus, latencies may be quite high and the bandwidth may be low, although the number of hops is small

An attempt to overcome the second disadvantage was performed by using identfier generating functions which partially preserve the geographic locality of the nodes [37].

It is woth mentioning the virtual coordinate space of most DHTs is one-dimensional. Structured, scalable communication topologies based on a multidimensional coordinate space were proposed in [3], [6] and [22].

A problem which is orthogonal to the construction and maintenance of a structured topology based on virtual coordinates is that of generating coordinates for each node, which are strongly connected to the network properties of the nodes. Thus, Vivaldi [12] and Sequoia [28] are systems which can generate coordinates for the nodes within the system in a distributed manner, such that the distance between nodes in the coordinate space is proportional to the latency between the two nodes.

C. Unsolved Problems

The construction of structured topologies based on node coordinates in multidimensional spaces and which uses systems like Vivaldi or Sequoia would lead to the situation in which the neighbors of a node have a small latency or a high

bandwidth towards the node. This way, message routing within the topology could be significantly improved. However, there are only a few methods of constructing structured topologies in multidimensional coordinate spaces. Besides, using systems like Vivaldi or Sequoia is not so easy, because the node coordinates are constantly changing, which could produce an unstable topology. Thus, among the unsolved problems of great practical and theoretical interest there are:

- the development of new methods for constructing and maintaining structured communication topologies based on static node coordinates in multidimensional spaces which present global properties like: low diameter, limited maximum degree of a node, the possibility of routing messages based on the geometric node coordinates and inter-node distances
- the development of new methods of constructing and maintaining structured topologies based on dynamic node coordinates in multidimensional spaces and which presents an improved stability

Another problem of structured communication topologies consists of the fact that they cannot include nodes located behind NAT or firewall devices. An open problem is finding a way to include and efficiently use this kind of nodes within structured topologies.

D. Motivation for Performing Research Activities

The development of new methods for constructing and maintaining structured topologies based on geometric node coordinates in a multidimensional space would create a link between the important mathematical properties of the virtual coordinate space (e.g. geometric routing based on the distance towards the destination) and the physical properties of the system nodes (e.g. latency, bandwidth). This way, we could benefit from the advantages of both perspectives (the virtual and the real one). These topologies could then be used in order to provide high level communication services with improved performances (or even with quality-of-service guarantees).

The efficient inclusion of nodes located between NAT or firewall devices would improve considerably the degree of applicability and usability of the communication topology. A large percent of Internet users use NAT devices, and including them in the topology would significantly increase the amount of resources available within the system.

E. Obtained Research Results

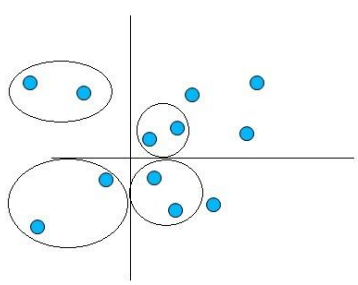

Figure 4. Neighbor selection in a 2D virtual coordinate space: 2 closest neighbors from each quadrant.

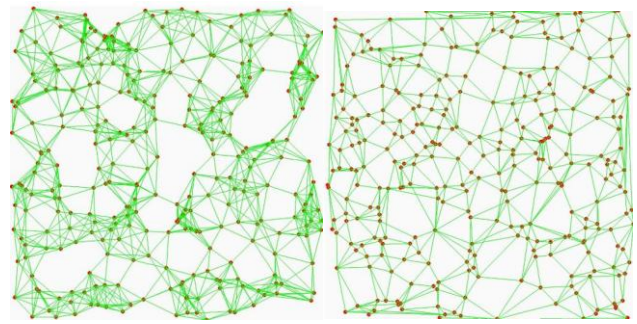

Figure 5. Neighbor selection in a 2D virtual coordinate space: Left-K-nearest neighbors (K=6); Right-K-nearest neighbors from each of R regions (K=1;R=4).

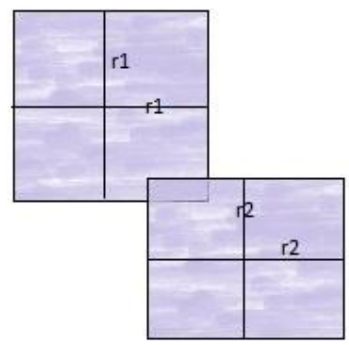

Figure 6. Computing the peer coordinates based on the $L\infty$ distance in a 2D virtual coordinate space. Latency to a set of landmark nodes is computed and the coordinates of the peer are approximated based on intersecting growing squares (of different sizes) around the coordinates of the landmark nodes.

A general framework for developing distributed applications and services based on structured communication topologies was proposed. The framework consists of two levels. The first one is the communication module level, in which point-to-point communication is handled (and the results from the previous section are applicable). The second level is concerned with topology construction and management. Gossiping (periodical transmission of information to the topology neighbors and/or extended neighbors) is at the core of the topology maintenance and update. In order to construct the topology, the framework uses a neighbor selection function. Various neighbor selection functions based on peer coordinates in multidimensional spaces have been considered and evaluated (e.g. based on selecting K-nearest neighbors from multiple regions or on using a hyper-plane division of the multidimensional space – see Fig. 4 and 5 for some examples).

Moreover, several novel techniques for assigning peer coordinates such that the distance in the metric space is proportional to the latency in the Internet between the peers have been designed and tested (see Fig. 6).

IV. Multicast Communication Services

The simplest (but also most inefficient) method of providing multicast communication services is for the source to transmit the content independently to every destination. This approach presents several disadvantages, like: the generation of a large

amount of network traffic, the need for the source to know all the destinations and, last but not least, a low performance (because the source is overloaded).

At the network level solutions like IP Multicast [18] were proposed, but there are very few routers which support this mechanism. Thus, the only possibility to provide multicast communication services is based on constructing a communication topology between the nodes which will benefit from these services. Systems which use this method can be classified as follows:

- they build a multicast sub-topology embedded in an extisting communication topology
- they create a communication topology without using an existing communication topology

Most of the (sub-)topologies used for multicast communication have a tree structure. Such a structure minimizes the amount of network traffic, but presents the disadvantage of being very frail.

A. Multicast Communication Services based on Existing Communication Topologies

Based on the structured topology of the DHTs, systems for providing multicast communication services have been provided: Scribe [31] is based on Pastry and builds a single multicast tree for transferring data. Splitstream [10] is based on using multiple Scribe trees in parallel. Other multicast and broadcast solutions based on virtual spaces of coordinates using the XOR metric or distances based on the longest common prefix of two identifiers were presented in [17] and [36].

B. Multicast Communication Services which are not based on Existing Communication Topologies

Multicast communication services which do not use existing structured communication topologies build their own specific topology. Basically, these systems consider that every node could become a neighbor with any other node within the multicast topology, unlike the previous case, in which the set of neighbors was limited by the extisting communication topology.

ALMA [19] creates a multicast tree in which every node chooses as a parent that tree node towards which the communication cost is minimum (cost is defined as a combination between latency and packet loss rate). HMTP [18] creates a multicast tree which uses the IP Multicast facilities, where such facilities are available. A system called ZIGZAG, which maintains a balanced multicast tree, is presented in [34]. The degree of each node is of the order $O(K^2)$, and the tree diamater is of the order $O(log_K(N))$ (where K is a configurable system parameter). ZIGZAG has a hierarchical structures and every time a new node enters the system, the node contacts the root first. In [4] a method for constructing and maintaining a non-hierarchical multicast tree is presented, in which the degree of every node is $O(K)$, the tree diameter is $O(log_K(N))$ (under low node arrival and departure rates), and every node maintains information about $O(K^2)$ other nodes.

[9] presents a system based on the concurrent usage of several balanced multicast trees. Another approach based on the collaborative construction and maintenance of multiple multicast trees was presented in [35].

Except for the performance metrics, when constructing a multicast tree, other parameters can be considered, like anonymity [38] or data transfer reliability.

It is worth mentioning that, unlike the multicast trees based on structured communication topologies, many of the solutions presented in this section use a centralized component for selecting the neighbors in the tree.

C. Unsolved Problems

Multicast trees are frail and very sensitive to unannounced node departures. In these situations special measures must be taken in order to reestablish the connectivity of the tree. An open problem is to what degree estimations or information regarding the time moment when the nodes will leave the system can be used in order to construct a more stable tree (for instance, if at the moment a node leaves the tree this node is a leaf, then the tree does not become disconnected and no special measures need to be taken). A first attempt towards using such information with this purpose was presented in [33].

Another unsloved problem is represented by the way the construction of multicast trees could use the properties of an existing structured communication topology (e.g. a topology based on node coordinates in a multidimensional space) in order to obtain multicast trees with low diameter or with very good network properties (e.g. low latencies, high bandwidth).

A possible solution for the difficulty of maintaining a multicast tree is to not try to maintain the tree permanently, but rather build it on demand, based on the properties of a structured communication topology. Constructing multicast trees on demand was considered only rarely, because of its high overhead. However, some types of communication topologies (like those based on multidimensional coordinate spaces) have properties which can be exploited in order to construct the tree efficiently. These properties, as well as efficient construction methods of multicast trees in such topologies, have not been analyzed yet.

D. Motivation for Performing Research Activities in this Sub-Domain

Multicast communication has applications in many domains, like the live or on demand transmission of audio/video flows, the quick transmission of signals and/or notifications to all the members of a system, text, audio and/or video conferences, and many others. The stability improvement of a multicast tree would increase the quality of the communication service, because there would be fewer service interruptions when the nodes leave the tree unannounced. The usage of the properties of an existing communication topology on top of which a multicast tree can be constructed could lead to improvements

of the data transfer parameters (e.g. latency, bandwidth, reliability). The construction of on demand multicast trees uses properties which are specific to an existing communication topology and can be a method both for improving the stability, as well as for decreasing the effort of constructing and, especially, maintaining a multicast tree.

E. Obtained Research Results

The multicast tree topologies developed as part of the research project can be classified into the same two categories mentioned at the beginning of this section: (1) embedded within a more general structured communication topology (having a broader scope); (2) constructed only (and optimized) for multicast dissemination.

As part of the first category, techniques for approximate multicast (where not all messages necessarily reach all the peers or some peers may receive duplicate messages) and for constructing multicast trees with improved stability properties (i.e. we make use of departure time information known in advance in order to construct a multicast tree which never gets disconnected) have been developed (see Fig. 7 for an example). As part of the second category, a technique for constructing and maintaining a multicast tree with small diameter and bounded node degrees in a fully decentralized manner has been devised (see Fig. 8 for an example).

V. Communication Optimization Problems in Distributed Systems of Contextual Interest

A. Optimization of Point-to-point Data Transfers within a Communication Topology

In [1] the Bittorrent protocol is used for developing a data transfer optimization service in Grid systems. Optimization of media flows in peer-to-peer overlays using a distributed approach was considered in [7]. A communication architecture based on a one-dimensional structured communication topology which allows the routing of messages over multiple paths and traffic load balancing was proposed in [2].

B. Centralized Scheduling of Data Transfers in Distributed Systems

In order to provide quality-of-service guarantees for the data transfers (e.g. bandwidth, duration) the possibility of using a central scheduler which has full control over the communication topology has been considered. The communication topology must „fit" as well as possible over the underlying physical network topology in order to provide strong quality guarantees. In [14] and [5] architectures and algorithms for this data transfer scheduling model have been proposed. Polynomial data transfer scheduling algorithms with constraints, under the conditions in which the parameters of the communication topology are known and well determined have been proposed in [11] and [32].

VI. Conclusions and Future Work

This paper presented the motivation, main challenges, unsolved problems and the results obtained so far of the post-doctoral research project of the author. Ongoing research activities will tackle the remaining unsolved problems and the obtained solutions will be implemented into a fully distributed and decentralized communication service for heterogeneous networks.

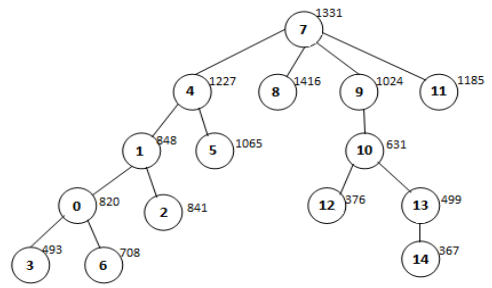

Figure 7. Multicast tree with improved stability. Departure times are written next to each node. Note how earlier departure times are closer to the outer edges of the tree. When a peer departs it will be a leaf in the tree and, thus, the tree will not get disconnected.

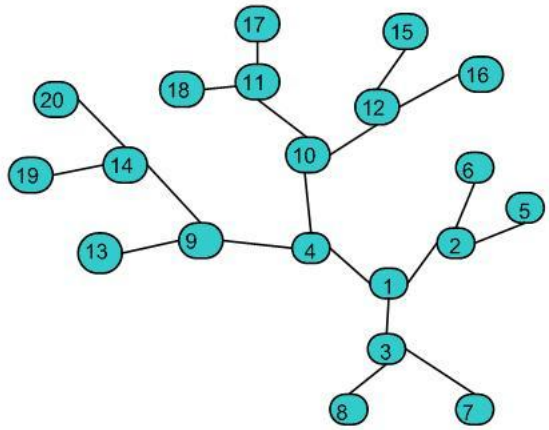

Figure 8. Balanced multicast tree overlay with 20 peers and maximum degree set to 3.

References

[1] A. Zissimos, K. Doka, A. Chazapis, N. Koziris, „GridTorrent: Optimizing Data Transfers in the Grid with Collaborative Sharing", Proceedings of the 11th Panhellenic Conference on Informatics, 2007.

[2] M. I. Andreica, I. C. Legrand, N. Tapus, „Towards a Communication Framework based on Balanced Message Flow Distribution", Proc. of the IEEE International Conference on "Computer as a Tool" (EUROCON), pp. 2354-2359, 2007.

[3] M. I. Andreica, E.-D. Tirsa, N. Tapus, „A Fault-Tolerant Peer-to-Peer Object Storage Architecture with Multidimensional Range Search Capabilities and Adaptive Topology", Proc. of the 5th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 221-228, 2009.

[4] M. I. Andreica, E.-D. Tirsa, N. Tapus, „Data Distribution Optimization using Offline Algorithms and a Peer-to-Peer Small Diameter Tree Architecture with Bounded Node Degrees", Proceedings of the 17th International Conference on Control Systems and Computer Science (CSCS), vol. 2, pp. 445-452, 2009.

[5] M. I. Andreica, E.-D. Tirsa, N. Tapus, F. Pop, C. M. Dobre, „Towards a Centralized Scheduling Framework for Communication Flows in Distributed Systems", Proceedings of the 17th International Conference

on Control Systems and Computer Science (CSCS), vol. 1, pp. 441-448, 2009.

[6] M. I. Andreica, E.-D. Tirsa, N. Tapus, "A Peer-to-Peer Architecture for Multi-Path Data Transfer Optimization using Local Decisions", Proc. of the 3rd Workshop on Dependable Distrib. Data Management, pp. 2-5, 2009.

[7] A. Argyriou, J. Chakareski, „Distributed Optimization of Media Flows in Peer-to-Peer Overlay Networks", Proceedings of the IEEE Global Telecomm. Conf., 2008.

[8] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", Proceedings of the ACM SIGCOMM Conference, pp. 24-35, 1994.

[9] M. den Burger, T. Kielmann, H. E. Bal, „Balanced Multicasting: High-Throughput Communication for Grid Applications", Proceedings of the ACM/IEEE Conf. on Supercomputing, 2005.

[10] M. Castro, et al., "Splitstream: High-Bandwidth Multicast in Cooperative Environments", Proceedings of the 19th ACM Symposium on Operating Systems Principles, 2003.

[11] B. B. Chen, P. V.-B. Primet, „Scheduling Deadline-Constrained Bulk Data Transfers to Minimize Network Congestion", Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid, pp. 410-417, 2007.

[12] F. Dabek, R. Cox, F. Kaashoek, R. Morris, "Vivaldi: a Decentralized Network Coordinate System", Proceedings of the International Conf. on Applications, Technologies, Architectures, and Protocols for Computer Comm., pp. 15-26, 2004.

[13] C.-N. Chuah, R. Keralapura, „Overlay Networks: Applications, Coexistence with IP Layer, and Transient Dynamics", Algorithms for Next Generation Networks, Chapter 8, Springer-Verlag London, 2010.

[14] C. Cirstoiu, R. Voicu, N. Tapus, „Framework for High-Performance Data Transfers Optimization in Large Distributed Systems", Proc. of the IEEE International Symposium on Parallel and Distributed Computing, pp. 385-392, 2008.

[15] Y. Gu, X. Hong, M. Mazzucco, R. L. Grossman, "SABUL: A High Performance Data Transfer Protocol", IEEE Communication Letters, 2003.

[16] Y. Gu, R. L. Grossman, „UDT: UDP-based Data Transfer for High-speed Wide Area Networks", Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 51 (7), pp. 1777-1799, 2007.

[17] Z.-J. Han, R.-C. Wuang, Y. Wang, „PPmcast: a Novel P2P-based Application-level Multicast using Kamulia", Journal of China Universities of Posts and Telecommunications, vol. 16, pp. 114-119, 2009.

[18] B. Zhang, W. Wang, S. Jamin, D. Massey, L. Zhang, "Universal IP Multicast Delivery", Computer Networks, vol. 50 (6), pp. 781-806, 2006.

[19] J. R. Iyengar, P. D. Amer, R. Stewart, "Concurrent Multipath Transfer using SCTP Multihoming over Independent End-to-End Paths", IEEE/ACM Transactions on Networking, vol. 14 (5), pp. 951-964, 2006.

[20] C. Jin, D. X. Wei, S. H. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance", Proceedings of the IEEE INFOCOM, 2004.

[21] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, „A Survey and Comparison of Peer-to-Peer Overlay Network Schemes", IEEE Comm. Surveys and Tut. 7, pp. 72-93, 2005.

[22] E. K. Lua, X. Zhou, J. Crowcroft, P. V. Mieghem, "Scalable Multicasting with Network-Aware Geometric Overlay", Computer Communications, vol. 31 (3), pp. 464-488, 2008.

[23] T. Kelly, „Scalable TCP: Improving Performance in Highspeed Wide Area Networks", ACM SIGCOMM Computer Communication Review (April 2003) 33, pp. 83-91, 2003.

[24] A. Kuzmanovic, and E. Knightly, "TCP-LP: Low-Priority Service via End-point Congestion Control", IEEE/ACM Transactions on Networking, vol. 14 (4), pp. 739-752, 2006.

[25] G.-I. Kwon, and J. W. Byers, „ROMA: Reliable Overlay Multicast with Loosely Coupled TCP Connections", Proc. of the 23rd IEEE INFOCOM, pp. 385-395, 2004.

[26] S. Liu, M. Vojnovic, and D. Gunawardena, "Competitive and Considerate Congestion Control for Bulk Data Transfers", Proc. of IWQoS, 2007.

[27] M. Mathis, "Relentless Congestion Control", Proceedings of the Protocols for Large-Scale and Diverse Network Transports (PFLDNeT), 2009.

[28] V. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, A. Gupta, A. Akella, "On the Treeness of Internet Latency and Bandwidth", Proc. of the 11th Intl. Conf. on Measurement and Modeling of Comp. Systems, pp. 61-72, 2009.

[29] I. Rhee, L. Xu, „CUBIC: A New TCP-Friendly High-Speed TCP Variant", Proceedings of the 3rd International Workshop on Protocols for Fast Long-Distance Networks, 2005.

[30] D. Rossi, C. Testa, S. Valenti, P. Veglia, and L. Muscariello, "News from the Internet Congestion Control World", Research Report 2009D016, Telecom ParisTech, 2009.

[31] A. Rowstron, A. M. Kermarrec, M. Castro, P. Druschel, „Scribe: The Design of a Large Scale Event Notification Infrastructure", Proc. of the 3rd Intl. Workshop on Networked Group Communication, 2001.

[32] P. Soldati, H. Zhang, M. Johansson, „Deadline-Constrained Transmission Scheduling and Data Evacuation in WirelessHART Networks", Proceedings of the European Control Conference, 2009.

[33] Y. Tian, D. Wu, G. Sun, K.-W. Ng, "Improving Stability for Peer-to-Peer Multicast Overlays by Active Measurements", Journal of Systems Architecture, vol. 54, iss. 1-2, pp. 305-323, 2008.

[34] D. A. Tran, K. A. Hua, T. Do, „ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming", Proceedings of the 22nd IEEE Conference INFOCOM, pp. 1283-1292, 2003.

[35] V. Venkataraman, K. Yoshida, P. Francis, „Chunkyspread: Multi-Tree Unstructured Peer-to-Peer Multicast", Proceedings of the IEEE International Conference on Network Protocols, pp. 2–11, 2006.

[36] M. Wahlisch, T. C. Schmidt, G. Wittenburg, „Broadcasting in Prefix Space: P2P Data Dissemination with Predictable Performance", Proc. of the 4th International Conf. on Internet and Web Applications and Services, pp. 74-83, 2009.

[37] W. Wu, Y. Chen, X. Zhang, X. Shi, L. Cong, B. Deng, Xing Li, „LDHT: Locality-Aware Distributed Hash Tables", Proceedings of the International Conference on Information Networking, 2008.

[38] L. Xiao, X. Liu, W. Gu, D. Xuan, Y. Liu, „A Design of Overlay Anonymous Multicast Protocol", Proceedings of the IEEE International Parallel and Distributed Systems Symposium (IPDPS), 2006.

[39] T. Yamamoto, "Estimation of the Advanced TCP/IP Algorithms for Long Distance Collaboration", Fusion Engineering and Design, vol. 83, iss. 2-3, pp. 516-519, 2008.