Published in Lecture notes in computer science 1830, 122-132, 2000
which should be used for any reference to this work

Message Chains and Disjunct Paths for increasing communication performance in large networks

Markus Wulff[1], Peter Kropf[2], and Herwig Unger[1]

[1] Universität Rostock, FB Informatik, D-18055 Rostock, Germany,
{mwulff,hunger}@informatik.uni-rostock.de
[2] Université de Montréal, Montréal, Canada H3C 3J7, kropf@iro.umontreal.ca

Abstract. In this paper we propose a simple powerful method to increase communication performance in large networks such as the Internet. Our approach is based on the usage multiple disjunct paths for the transmitting data from one node to another one. The method exploits the dense connectivity of the Internet by making use of the different routes engaged in communications originating from different nodes to one target nodes. We show that depending on the bandwidth offered and the amount of data to be transfered, a linear speedup can be achieved. An experimental implementation of the method within the WOS framework confirms the theoretical analysis of the method.

1 Introduction

To access the huge amount of mostly inefficiently used hard- and software-resources in the Internet, new approaches to use and maintain this potential are needed. The complex, inhomogeneous and dynamic structure of the world-wide network requires decentral, adaptive mechanisms, that provide access to both local and global capacities in a efficient and transparent way [10]. For any system implementing such functionality, it is crucial to provide to the applications and users eased access to advanced network services. In particular, applications and users request for ever more bandwidth and high speed data transmission. However, it is acknowledged that despite the tremendous advances in network technology, there remains a high demand for even more communication capacity. While the network capacities in the context of the global communications infrastructure or the Internet appear to be insufficient, there exists still a potential of using this infrastructure more efficiently. In fact, when considering the traffic on the Internet, we observe that at the lower communication layers the various techniques of multiplexing (time-, frequency-, code-division, etc) are heavily used to maximize the efficient use of the communications infrastructure. As far as the data transfer is concerned, sophisticated compression techniques are applied to decrease communication times.

There exist however, other ways to increase communication speed as observed by an application or user. If we consider a file transfer, the transport layer will usually set up an end-to-end connection. The network layer usually offers a connectionless packet switching service (e.g. IP). While it might be possible for the packets to travel along different routes, the data transfer still takes place within the scope of the connection set

up at the higher level, thus realizing a sequential communication path. These different mechanisms do not improve communication speed (as seen by the application or user), but rather avoid communication bottlenecks. A simple way to increase communication speed would be to use different connections with different communication paths in parallel for different chunks of data from the same data stream.

This paper presents a method allowing to exploit this potential of parallel usage of the global communication infrastructure for a single data stream. First, we briefly present the Web Operating System (WOS™) [9] environment providing application and user support for the transparent use of the global information infrastructure. Sect. 3 introduces the concept of message chains and disjunct paths for parallel communication and formally analyzes the possible gain in communication speed. Finally, Sect. 4 discusses the implementation of the method within the WOS and presents first experimental results.

2 The Web Operating System – a brief Overview

The Web Operating System (WOS™) [9] is an open middleware solution allowing for software services to be distributed over the Internet. It supports applications and users with mechanisms to offer own resources and to locate and use accessible remote resources while taking advantage of the ever changing software and hardware infrastructure of global systems. The WOS approach to support distributed computing relies on the widespread use of version control techniques. It is a decentralized system, for which there is no complete catalog of available services and resources. Such knowledge is rather dynamically built up and managed. Therefore, the central component of the WOS is a generic communication framework allowing for versioned protocols, used to support communication between components or nodes. Each node operates as a server and a client at the same time and maintains warehouses [14] with information about other nodes. This allows to considerably reduce the response time to requests because expensive search procedures may often be avoided while relying on the information present in the warehouses for fulfilling service requests. The information in these warehouses are continuously updated when new knowledge about other nodes becomes available through service requests issued by the node itself or by other nodes. A collection of such nodes form a WOSNet which is managed in a completely decentralized manner.

The WOS communication layer uses a two-level approach [2]. The first layer offers discovery/location services and the second one is used for service invocation. To build up the WOSNet and to locate requested services and appropriate resources, a search mechanism based on multiple sequential chains has been defined [2, 15]. Based on the knowledge of the WOSNet a node belongs to, a set of search chains consisting in a sequence of nodes to visit along disjunct paths is defined prior to starting the search. The search along these chains is thereafter issued in parallel. Theoretical as well as practical investigations have shown the efficiency of this procedure [15].

A WOSNode is composed of different modules to handle communication, i.e. incoming and outgoing messages using the two WOS communication layers, and to process the associated requests for services, resources or system information. The node's information warehouses, which are some form of cache, store information about its own

and other nodes' services and resources. In particular, this includes addresses of other known WOSNodes which thereby defines a WOSNet as a logical structure on top of the Internet. Fig. 1 shows an example for such a structure. A WOSNode usually knows the machines in its neighborhood. Other machines can be found through the search mechanisms mentioned above.

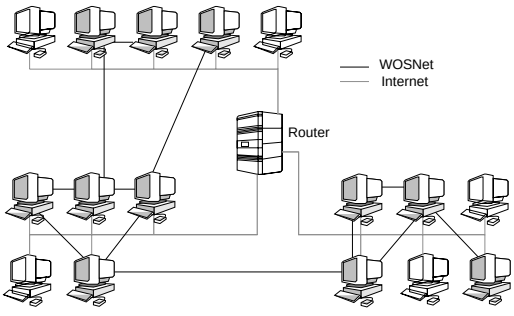

Fig. 1. The structure of the WOSNet.

The possibly large number of nodes, widely spread all over the Internet provide the precondition for new, efficient ways of communication between the machines belonging to the WOSNet by extending the WOS communication layer for exploiting multiple disjunct communication paths between two communicating nodes. The source node of a communication may contact other WOSNodes in his neighborhood which will act as starting or base nodes for using multiple disjunct communication paths to the destination. These multiple paths can be used in parallel and will therefore lower the overall communication time between the source and destination node. It must be noted, that only the source node, the base nodes and the destination node need to be part of a WOSNet. The intermediate nodes may be any node belonging to the Internet.

3 Message Chains and Disjunct Paths

3.1 Internet and Routing

In contrast to parallel and cluster computing systems exhibiting much potential to optimize the communication by different routing algorithms [1, 3, 4, 16], distributed systems mostly just support point to point connections. In some cases, like in subnetworks, broadcasts and multicasts are possible [8]. Because of the large structure of the Internet, routing algorithms cannot be influenced by the user and are rather ruled by standard routing procedures [13]. Other approaches, like [6], try to improve the communication behavior between loosely coupled workstations through low level communication optimization. These approaches are only successful in local environments. Other communication systems like ATM and IPv6 use special mechanisms to provide efficient communication through Quality of Service (QoS) mechanisms and policies [7]. Only

few works, such as [11], motivate that user- or application specific dispositions may require the use of disjunct paths in the Internet.

3.2 Multiple sequential search chains

As mentioned in Sect. 2, a search and localization method based on multiple sequential search chains [2, 15] has been applied to the WOS framework. This method basically consists in a combination of the broadcast and the serial search and request strategy. The broadcast strategy is used to perform searches in the local area (e.g. on a diffusion based subnetwork such as an Ethernet) and to concurrently initiate multiple serial searches at different nodes in the network. The serial searches are defined through lists of nodes to visit. These lists are built up based on the information available in the node's warehouse. The search for a service or resource consists thus basically in sending search lists to a set of nodes which then concurrently perform the search defined in the associated list for that node. This first step of distributing the lists follows thus conceptually a broadcast or multicast strategy. However, as opposed to pure broadcasting, the data sent to the nodes are individual data. Using a stochastic time estimation model, this strategy was shown to yield very favorable response times [15]. The performance of searching for services and resources can be further improved by using *Message Chains* [17].

3.3 Message chains and disjunct paths

The concept of message chains is based on active messages [5]. Tokens are the basic components of a message chain. Beside the information to be transmitted, each token contains a set of addresses of machines, which must be visited one after the other, thus representing the lists introduced above. Each of those lists defines a *disjunct path* in the network from the source node to the destination node of a communication. The different communication daemons of the machines (way points) support an automatic active forwarding of any token to the next machine in the given chain. In addition, every machine may have to execute a service request as specified for the token processed before forwarding it to the next node. The output of such services is sent directly back to the requesting node or can be attached to the token. An example of such a service might be to provide information about the current load of the communication channels the visited node is connected to.

In complex networks like the Internet, almost every machine can be reached using more than one path. Therefore, two different nodes could use more than one route thereby improving performance. However, it is usually not possible for a user or an application to decide which route a packet (i. g. an IP datagram) should use. This is the responsibility of the lower protocol layers [8]. Furthermore, the default routing between two machines may change from time to time. But these changes of the standard routing paths usually do not occur very frequently, and it is possible to trace a route and to determine the average communication parameters for the transmission.

The message chains and disjunct paths approach takes advantage of the potential of multiple routes. Some selected nodes in the neighborhood of the source node act as base nodes from which the data is transmitted along their respective standard routes to the destination node. The standard routes for these base nodes to the destination node

should of course be disjoint or at least differ considerably. This could for example be detected prior to any transmission using ICMP messages in a similar way the `traceroute` utility explores a route.

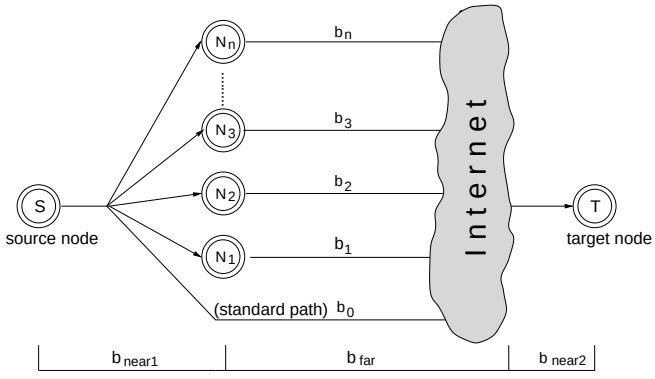

Fig. 2. Communication along disjunct paths.

Fig. 2 shows the principles of a communication using disjunct paths. The source node S can split the message to transmit into n pieces of different sizes and send them to T via the base nodes N_i using n disjunct paths offering the bandwidth $b_0 \ldots b_n$. The standard path with the bandwidth b_0 is the route from S to T without involving any base node. The transmission path from S to T can be divided into two local (near) areas in the vicinity of S and T, and a long distance area (far) between the base nodes N_i and the neighborhood of T.

This communication scheme requires the node S to know suitable base nodes N_i to use disjunct paths. The WOSNet introduced in Sect. 2 defines a logical structure over the Internet and provides the necessary mechanisms to find the nodes which may act as base nodes. Of course, the use of disjunct paths for data transmission makes sense only, if the amount of data to be sent is big enough. The speedup gained with disjunct paths must be higher than the time necessary to find the bases and their alternate routes.

3.4 Analysis of the communication speedup

To analyze the performance of this method, we consider the case of a file transfer from a node S to node T such as illustrated in Fig. 2. There are $n + 1$ paths available with the bandwidth $b_0 \ldots b_n$ to transmit a file of size D. The bandwidth of the standard routing path between S and T is assumed to be b_0. The transmission time t is the total amount of time needed to transmit the data D. In order to achieve a transmission speedup, the time $t_{\mathrm{near}} = t_{\mathrm{near}_1} + t_{\mathrm{near}_2}$ for all local connections from the source S to the bases N_i ($i = 1 \ldots n$) must be much smaller than the time t_{far} for the long distance connections within the network. In this case, the share of t_{far} in the total communication time t determines the potential for communication speedup using disjunct paths. The

transmission time for sending the data using the standard route only (serial case) is

$$t_{\text{serial}} = t_{\text{near}} + t_{\text{far}} = \frac{D}{b_{\text{near}}} + \frac{D}{b_0}, \tag{1}$$

where b_{near} denotes the mean bandwidth available for the local communications near S and T ($b_{\text{near}} = (b_{\text{near}_1} + b_{\text{near}_2})/2$). Equation (1) makes clear, that a speedup can only be achieved, if $b_{\text{near}} \gg b_{\text{far}}$, e. g. the transmission of data to the bases $N_1 \ldots N_n$ is pretty fast. Depending on the bandwidth b_i, the whole amount of data D shall be divided into pieces $d_0, d_1, \ldots, d_n$ such that the transmission through all $n+1$ different paths can be accomplished in almost the same time $t_{disjunct} = \frac{d_i}{b_i}, i = 0 \ldots n$. When transferring the data D in parallel along the $n+1$ paths, the transmission time t_{far} becomes $t_{far} = \frac{d_i}{b_i}$ with $i = 0 \ldots n$:

$$t_{far} = \frac{D}{b_0 + b_1 + \cdots + b_n} \tag{2}$$

The size each piece of data ($d_0, d_1, \ldots, d_n$) for path i can be determined by

$$d_i = \frac{D}{b_0 + b_1 + \cdots + b_n} b_i \quad \text{with} \quad i = 0 \ldots n \tag{3}$$

The overall transmission time $t_{parallel}$ using disjunct paths becomes now

$$t_{\text{parallel}} = \frac{D}{b_{\text{near}}} + \frac{D}{\sum_{i=0}^{n} b_i}. \tag{4}$$

Let us assume $b_{\text{near}} = k_{\text{near}} b_0$, i.e. the bandwidth b_{near} is defined as the bandwidth of the standard path multiplied with some constant value. The bandwidth for the other paths is defined in a similar manner relatively to b_0 by $b_i = k_i b_0, i = 1 \ldots n$. The communication speedup using disjunct paths as compared to the use of single route only becomes

$$S_{\text{comm}} = \frac{t_{\text{serial}}}{t_{\text{parallel}}} = \frac{b_0 + k_{\text{near}} b_0}{b_0 + \frac{k_{\text{near}} b_0}{1 + k_1 + \cdots + k_n}} \tag{5}$$

By replacing $k_1 + \cdots + k_n$ in (5) by K with $k_i = \frac{b_i}{b_0}$, the speedup becomes

$$S_{\text{comm}} = \frac{1 + k_{\text{near}}}{1 + \frac{k_{\text{near}}}{1+K}} \quad \text{with} \quad K = \frac{\sum_{i=1}^{n} b_i}{b_0} \tag{6}$$

If the local communications times (t_{near}) are very fast, we have $k_{\text{near}} \to \infty$ which results in a speedup of K. On the other hand, if the accumulated bandwidth available along the disjunct paths $i = 1 \ldots n$ is small as compared to the bandwidth b_0 of the standard path, we have $K \to 0$, which results in a speedup of 1, i.e. there is no speedup at all. Fig. 3 visualizes the expected speedup S_{comm} as defined in (6), depending on k_{near} and K.

A larger k_{near} represents a faster connection in the local area as compared to the long distance connection, resulting in a growing speedup. This growth is not linear,

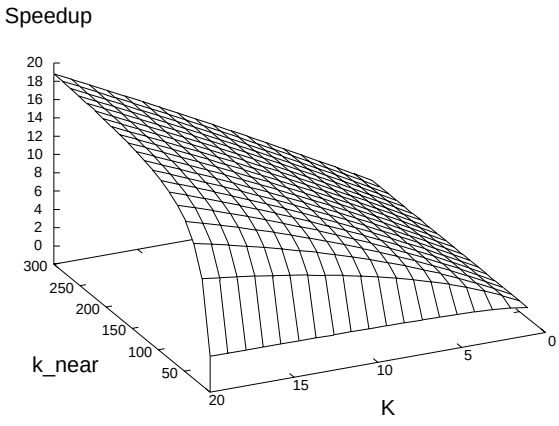

Fig. 3. Visualization of the speedup through disjunct paths.

because the influence of b_{near} on the overall performance is small. After some point a faster local connection will therefore not increase the speedup any more. If further powerful disjunct paths are added, the growth of the speedup raises in a linear manner. It must be noted that throughout this performance analysis no overhead introduced by the method has been considered. The saturation of the speedup growth in a real system depends of course as well on such overhead times, which depend, as well as the latency introduced, on the size of the data subsets d_i. The following two assumptions should therefore be made in order to benefit from this method:

1. not every machine in the Internet tries to use disjunct paths at the same time (WOSNet $\subset$ Internet);
2. a sufficient amount of data D should be transmitted, and the data subsets d_i should not become too small.

4 The Realization in the WOS

4.1 The WOSForward service

Message Chains are introduced in the WOS as a special service: WOSForward service (WFS). If a user or application wants to transfer data from one machine to another one using disjunct paths, he issues an appropriate request to his WOSNode. As described in Sec. 3.2, other WOSNodes are needed as base for this kind of service. The requesting user or application first searches the WOSNet for base nodes in its neighborhood. This search is made by calling the standard search and location services of the WOS using the WOS communication layer. If suitable nodes with a fast connection to the requesting node, i.e. with a high bandwidth b_{near1}, are found and the WOSForward service is available at all those nodes, the data to be sent is split into pieces depending on the number of paths and their bandwidth (see equation (3)). These pieces are then sent out in a quasi parallel manner on the disjunct paths.

At each base node the incoming data packets will be forwarded to the target machine using the standard routing path of that node. The flow of a service request at a base node is shown in Fig. 4.

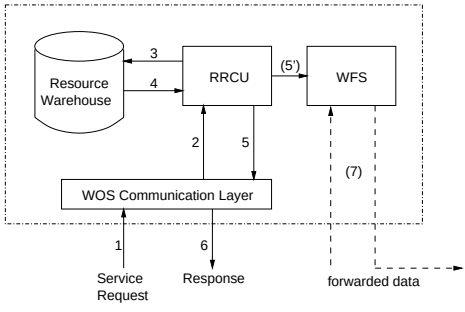

Fig. 4. The flow of a WOSForward request in a base WOSNode.

The incoming request (1) is passed directly to the module *Remote Resource Control Unit* (RRCU) (2). The RRCU looks up the requested WOSForward service in its local Service/Resource Warehouse (3) and (4). If this service is available on the node and is opened by the owner for public use, the WOSForward service (WFS) process is started by the RRCU (5'). In either case, whether the service is available or not, an answer is returned to the requesting node (5, 6). Then, assuming the service is available, the data can be transmitted. It will be directly forwarded by the WOSForward service, thus bypassing the WOS communication layer, which improves the communication speed by eliminating its protocol overhead.

4.2 Experimental Results

First experiments have shown, that the expected speedup can be achieved. Fig. 5 shows the environment for the test. Files of $1, 2, 4 \ldots 64$ MB were first sent from the source machine S to the target machine T along the standard path without using any WOS services. Then, each file was split into two equally sized pieces and sent via two disjunct paths, the standard path and the path originating at machine R. Each machine is a WOSNode and at the base node R the WOSForward service was installed to pass the data from S to T.

Fig. 6 shows the result obtained with this experiment. The measurements confirm the speedup predicted in Sect. 3.4, equation (6). For larger file sizes, a linear speedup is observed, while the overhead introduced by the base node search and the splitting up of the files eliminates the speedup gained in the case of smaller files.

Further experimentation is currently under way. This involves in particular larger WOSNets as well as simultaneous use of the WOSForward service by different source nodes within the same WOSNet. Moreover, we will also investigate the influence of latency on the speedup achieved.

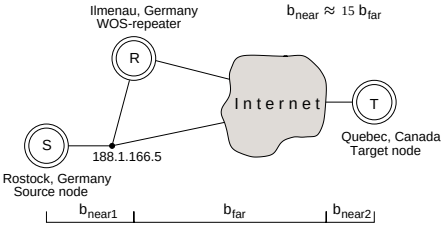

Fig. 5. Experimental setup for the communication with two disjunct paths.

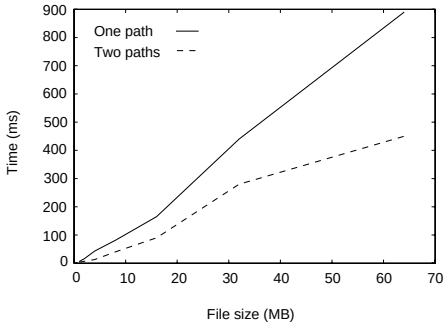

Fig. 6. Transmission times for one path and two disjunct paths.

4.3 Future developments

The search mechanisms for finding base nodes is one source of overhead introduced. While this method was already shown to be efficient [15], further work is necessary to optimize the management and maintenance of the warehouses, which store information about the WOSNet. Because there is no central catalog for a WOSNet, the concept of dynamically defined or versioned *communities* of components (such as WOSNodes) will be applied [12]. Components can adapt their behavior as they enter or leave a community, and may participate in several communities simultaneously. They can negotiate their relationship with other communities and may modify the shared context with other members of a community. This concept of communities may be well applied to the WOSForward service, where a suitable set of base nodes for possibly multiple source nodes may form a community who offers faster communications through its disjunct paths.

Splitting up data into different pieces to be transmitted along multiple paths exhibits as well a side effect concerning fault tolerance [11]. If one path brakes down, the data sent along the other ones still arrives and the receiving node may immediately request retransmission of the missing data along another path. This aspect of fault tolerance offered by the disjunct path communication will be further investigated.

5 Conclusion

This article shows that the concept of message chains and multiple disjunct paths can improve the performance of data transmission in the Internet. The Web Operating System (WOS™) provides all functionality which is necessary to realize this ideas. Together with an appropriate WOS service, this approach makes the working efficient and transparent for the user in many cases where a powerful search mechanism and a fast data transmission is needed.

References

[1] G. R. Ash. *Dynamic Routing in Telecommunications Networks*. McGraw Hill, New York, 1998.

[2] G. Babin, P. Kropf, and H. Unger. A Two-Level Communication Protocol for a Web Operating System (WOS). In *IEEE 24th Euromicro Workshop on Network Computing*, pages 934–944, Sweden, 1998.

[3] D. Bertsekas and R. Gallagher. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1991.

[4] J. T. Brassil, A. K. Choudhury, and N. F. Maxemchuk. The Manhattan Street Network: A High Performance, Highly Reliable Metropolitan Area Network. *Computer Networks and ISDN Systems*, 1994.

[5] R. Buyya. *High Performance Cluster Computing Vol 1*. Prentice Hall, Upper Saddle River, NJ, 1999.

[6] G. Hipper and D. Tavangarian. Evaluating the Communication Performance of Beowulf- and FastCNA- workstation clusters through Low-Level Communication Benchmarks. In SCS A. Tentner, editor, *High Performance Computing*, pages 271–276, Boston, 1998.

[7] G. Hommel. Quality of Communication-Based Systems. In *International Workshop on Quality of Communication-Based Systems*, TU Berlin, Germany, 1994. Kluwer Academic Publishers.

[8] G. Hunt. *TCP/IP Network Administration, Second Edition*. O'Reilly, Cambridge, 1998.

[9] P. Kropf. Overview of the WOS Project. In SCS A. Tentner, editor, *ASTC High Performance Computing*, pages 350–356, San Diego, CA, 1999.

[10] S. B. Lamine, J. Plaice, and P. Kropf. Problems of computing on the Web. In SCS A. Tentner, editor, *High Performance Computing*, pages 296–301, Atlanta, 1997.

[11] M. Lyubic and C. Cap. Enhancing Internet Security Through Multiple Paths. In *Distributed Computing on the Web (DCW'98)*, pages 55–60, Rostock, Germany, 1998.

[12] J. Plaice and P. Kropf. Intensional Communities. In World Scientific Press, editor, *Intensional Programming II*, Singapure, 2000.

[13] Y. Rekhter and T. Li. *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771, Mar. 1995. http://info.internet.isi.edu/in-notes/rfc/files/rfc1771.txt.

[14] H. Unger. The Adaptive Warehouse Concept for the Resource Management in the WOS . In *Distributed Computing on the Web (DCW)*, pages 167–174, Rostock, Germany, 1999.

[15] H. Unger, P. Kropf, G. Babin, and T. Böhme. Simulation of search and distribution methods for jobs in a Operating System (WOS). In SCS A. Tentner, editor, *ASTC High Performance Computing*, pages 253–259, Boston, 1998.

[16] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[17] M. Wulff and H. Unger. Message Chains as a new Form of Active Communication in the WOSNet. In SCS A. Tentner, editor, *ASTC High Performance Computing*, 2000.