

On the Stability of Peer-to-Peer Networks in Real-World Environments

Zoltán Czirkos

Budapest University of Technology and Economics, Hungary

Gábor Hosszú

Budapest University of Technology and Economics, Hungary

INTRODUCTION

Communication in computer networks can be organized in two different ways, according to the client/server model and the peer-to-peer model (Spinellis & Androutsellis-Theotokis, 2004). In the client/server model, the network is centralized. There is one host on the network, the server, which provides services to its clients. Its network address is usually well-known. On the other hand, in the peer-to-peer model, there is no central point in the network. Hosts participating are sometimes called “servents” (Gnutella, 2006), as they act both as servers and as clients at the same time: they provide services to other servents, while they also use the services of others.

Nodes in unstructured peer-to-peer networks usually communicate via message flooding. For example, a search request for a given file in the Gnutella network is sent to all neighboring servents. However, this solution is not scalable, and it generates a lot of unnecessary network traffic.

Structured peer-to-peer networks assign unique identifiers, called node IDs, to participating nodes. These overlay networks store key-value pairs. Every piece of information has a key, for example, the name of the file. This key is scrambled with a hash function (Lua, Crowcroft, Pias, Sharma, & Lim 2004), which generates a seemingly random number derived from the key. Then these numbers are used to assign files to specific nodes: each node stores the files, which have their names’ hash value numerically close to the unique identifier of the node. That is why structured networks are also called distributed hash tables (DHTs), as for every key, it is easy to find the node which stores the corresponding value. This process is called consistent hashing (Stoica, Morris, Karger, Kaashoek, & Balakrishnan 2001), as every node uses the same

hash function. Examples for structured networks are CAN (Ratnasamy, Francis, Handley, Karp, & Shenker 2001), Chord (Stoica et al., 2001), Pastry, and Kademlia (Maymounkov & Mazieres, 2002). These are all DHTs, but they use different topologies and routing mechanisms.

DISTRIBUTED HASH TABLES

In a distributed hash table, a hash function is used to derive a small number from a key representing some data (the value). Information is then stored on one of the participating peers. The application level networks mentioned above all use hash tables to store information, but the exact management of storage is different in the following three aspects (see Table 1):

- The selection of a hash function. Most networks use MD5 or SHA-1. The choice of a function is not really important for this use, as they are only used to make data evenly distributed among the nodes. For example, Kademlia uses SHA-1 (Eastlake & Jones, 2001).
- The selection of a metric function. A metric defines a distance between two IDs, so hashed data can be assigned to specific locations, nodes in the network. Every node stores key-value pairs, which have a hash value closest to its ID, according to the metric function. Kademlia uses the XOR operation.
- The selection of a topology. This is closely related to the selection of the metric function. CAN is usually symbolized as an n-dimensional torus, while it calculates the distance between identifiers by using the Pythagorean Theorem. Kademlia is usually represented with a binary tree. In Chord,

Table 1. Topologies DHT networks

Overlay network	Metric	Topology
CAN	Euclidean	n-dimensional torus
Chord	Subtraction	ring
Kademlia	XOR operation	binary tree

the peers are organized in a ring, and messages are always sent clockwise around the ring.

RELIABILITY IN PEER-TO-PEER NETWORKS

Reliability of a peer-to-peer network is directly influenced by the dependability of the underlying network packet transfer service. The different topologies, however, are affected differently by packet losses and other errors. Most structured peer-to-peer networks, for example, CAN (Ratnasamy et al., 2001), have an exact topology. In the n-dimensional circular torus of CAN, every node has to maintain only a small number of connections; in a two-dimensional example, this number is four (up, down, left, and right). Data to be sent are forwarded (routed) on the overlay network from node to node, finally arriving at their destination. CAN is therefore able to use session-oriented TCP as its transport protocol, as a node always communicates only with its neighbors.

There are other structured networks, which have no specific topology, for example, Kademlia (Maysounkov & Mazieres, 2002). In Kademlia, messages are not forwarded inside the overlay (there is no routing defined), rather they are sent directly between the source and the destination as datagrams. The purpose of the overlay is only to find the physical network address (IP address, port number) of the destination node in question, and it uses UDP for its messages. Therefore, network errors directly influence the communication between peers, and this is especially true for specific source-destination pairs. Permanent network errors, nodes that cannot be reached (because they are behind a firewall, for example), all degrade the quality and performance of the overlay.

The availability of a specific connection can naturally be tested by a simple ping message. Due to network errors, information available at nodes can sometimes be unreachable for others. The exact distribution of errors is usually highly uneven; with some nodes having good

connectivity and others not. This issue can be solved by the data replication. As node IDs are usually chosen randomly, nodes which are close to each other in the application network address space can be quite far from each other in the physical address space, and even geographically. Therefore, sending messages to more than one node, which are close to a specific destination, can result in replicating data at very different locations, almost as if destinations were randomly chosen.

Figure 1 shows our simulation of a random Kademlia overlay topology. The simulated scenario was that we tested all the participating nodes of the overlay (in this experiment, we used $n=200$ nodes) if they are able to send an information message to a certain destination node and some replication of the information message to the closest nodes of the destination. We did not use a real topology, since in Kademlia there is not real topology; in fact, every node can send message to every other. But the success of the sent message was measured by a random variable, namely the ratio of the bad links of each participating node. In this experiment, a replication factor of $k=8$ was selected, which means that every participating node stores its key-value pairs at eight different locations.

The ideal case is when all network connections are functioning, and there are no errors. If there are failing connections, senders of messages choose nodes as destinations, which are not the eight closest ones in the entire network, but a bit further in node identifier address space. This can happen since peers can detect the failing links. (For example, a node is intending to send its message to eight peers, but it detects that the node with the third closest address is unreachable. Then it sends its piece of information to the 9th closest node, too.)

In Figure 1, the three different plots visualize network messages arriving at destination nodes, in case of various ratios of nonfunctioning network links. The X axis shows the nodes of the overlay; they are sequenced in the order of closeness to the destination address. X=0 is the closest one; the higher the sequence number on the X axis, the further the node denoted by the actual sequence number from the destination node. In other words, X=0 is the primary destination (closest), X=1 is the secondary (second closest), X=2 is the tertiary, and so on. The Y axis shows the number of nodes, which could reach the destination. This is the value which is our point of interest. Let us suppose that we have a key-value pair stored in the overlay with 200 nodes, and



7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/stability-peer-peer-networks-real/13414

Related Content

Challenges to Records and Information Management in the Fourth Industrial Revolution

Christine Rigda (2021). *Handbook of Research on Information and Records Management in the Fourth Industrial Revolution* (pp. 223-240).

www.irma-international.org/chapter/challenges-to-records-and-information-management-in-the-fourth-industrial-revolution/284728

An Overview of Executive Information Systems (EIS) Research in South Africa

Udo Richard Averweg (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 2964-2970).

www.irma-international.org/chapter/overview-executive-information-systems-eis/14012

Determinants of ERP Implementation and System Success in India: A Case Study

R. Rajendran, V. Kalaiarasiand M. Amaravathi (2015). *Journal of Cases on Information Technology* (pp. 35-52).

www.irma-international.org/article/determinants-of-erp-implementation-and-system-success-in-india/139266

An Exploratory, Longitudinal Study of Factors Influencing Development of a Networked Company

Deborah Hardy Bednarand Lynn Godkin (2009). *Information Resources Management Journal* (pp. 59-72).

www.irma-international.org/article/exploratory-longitudinal-study-factors-influencing/1365

Information Fusion of Multi-Sensor Images

Yu-Jin Zhang (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 1950-1956).

www.irma-international.org/chapter/information-fusion-multi-sensor-images/13845