Abstract — The definition of multicast routing was already described more than 20 years ago. Many routing algorithms and the evaluation of these algorithms based on software techniques have been proposed since the year 2000. Multicast is required in various environments with consideration of transportation quality and different requirements. This means the requirements of multicast service increase steadily and it’s the time to meet economic and time-to-market with quality of service. In this paper, we describe the process of analysis, design and implementation of applying Overlay Multicast Protocol (OMP) to application software according to the arrangement with a client that wanted to apply specific requirements. In perspectives of software engineering, we share our experiences and development process for applying OMP.

Keywords — Overlay Multicast Protocol, Software Architecture, Software Engineering

1. Introduction

Software solutions are more effective to various services on the Internet when it is hard to utilize hardware-based multicast technique for supporting one-to-many or many-to-many communications. Therefore, many multicast solutions and routing algorithms have been proposed. In this paper, Tree Building Control Protocol (TBCP) is selected as the most appropriate multicast algorithm after surveying a number of existing algorithms and client’s requirements. Client’s requirements are analyzed and the scope is defined upon functional, non-functional, and technical constraints of the project, applying TBCP by a software engineering method for the same.

Recently, the requirement engineering is more important for the successful project. This is able to realize the cooperation with customer. It needs time to make a consensus on the goal, scope, terminology, and the artifacts from software life cycle. It is not independent on any phase such as design, implementation, and test. It is more typical for out-sourcing. Therefore, architecture design is important in point of customer co-work environment as well as requirements specification.

As the artifact from the project in cooperation with our customer, the design, implementation and testing of OMP architecture is described. In addition, we also mention the various engineering tasks that are carried out in the development process. Thus, it is expected that this paper summarizes our works and helps team projects which require application of overlay multicast techniques in accordance with software engineering.

2. The elicitation and analysis of requirements on Overlay Multicast Protocol (OMP)

To begin with, it is important to define the purpose of the project for applying OMP to one-to-many communication and in this an algorithm may be changed according to its practical usage. In general, defining the purpose of the service is the first step of development in the case of developing a routing algorithm. [1]

In the process of defining the purpose, the first tasks were to outline the business benefit, technical constraints and quality requirements of applying OMP. Performance, usability and scalability for security in the development were given major considerations.

In case of functional requirement, group configuration and member configuration each member of a group, multicast routing, and data replication are major consideration point. And performance and availability are most important and difficult attributes to indicate in the application such as traffic surveillance. In summary, the following concerns are considered with high priority:

- Response Time in communication between OMP server and client
- Ability of OMP server to cater to many clients

Therefore, it is designed upon quality attribute scenarios. First, it is discussed the situation that a video stream internal event is sent from the OMP server to the client during normal operating conditions. The client receives the video stream within 5 seconds of its dispatch from the OMP server using the maximum available bandwidth with 11 Mbps. Second scenario is that a video stream internal event is sent from the OMP server to a number of clients during normal operating conditions. The number of clients receiving the video stream is predefined number 100 from the OMP server using the maximum available bandwidth. For this requirement, OMP architecture is designed and elaborated. This architecture is also considered in point of availability upon the concern as ability of the system to address failure of data transmission between network nodes and between the OMP server and client nodes.

TBCP is decided as the major multicast algorithm after analyzing and comparing various academic research products for multicasting such as Nemo based on Java [10], Meridian from Connell University [11], End System Multicast from Carnegie Mellon University [12], Nice from University of Maryland [13].

ISBN 978-89-5519-131-8 93560 - 559 -
Feb. 12-14, 2007 ICACT2007
There are various views and perspectives that describe software architecture. In this paper, the Component and Connector view type (that depicts a runtime view of the system) is used. The OMP architecture was designed as Figure 1. Consideration was given to scalability for user’s security.

![Figure 1. OMP Architecture](image)

Based on above C&C style description, the C&C view of OMP describes dynamic behavior of OMP system, join and rejoin as well as data transmission that frequently happened in OMP environment. Among several C&C view type, the Figure 1 combines peer-to-peer, client-server and publish-subscribe style. In the peer-to-peer style of the C&C view type, components directly interact as peers by exchanging services. In OMP system, each node should keep the information where which node is its parent and which nodes are its child while communicating with surrounded nodes. Because of the dynamic change of node configuration, a node can directly communicate any node while constructing stream path [9].

In the client-server style of the C&C view type, servers provide a set of services though one or more interfaces and clients use zero or more services provided by other servers in the system. The OMP system C&C view architecture follows the client-server style in that the OMP server provides node information and video stream to clients. The publish-subscribe style is used in the each nodes for dynamic node configuration. The event bus and surrounded components in the figure 1 represents the publish-subscribe style. The style minimizes the dependency among procedures so that the system improves modification.

3. Analysis and application of OMP Algorithm

3.1 TBCP algorithm

TBCP has been designed according to the Application Level Multicast (ALM) model, also known as application-based distribution [7]. ALM is a new technique used to provide multicast distribution services in situations where no support is provided by the network for multicast. TBCP can build a loop-free spanning tree connecting TBCP entities that may run either in end systems or in programmable edge devices. Such a spanning tree is built by means of a distributed algorithm that does not require any interaction with multicast routers and any knowledge of network topology.

Because OMP algorithm does not centralize the node configuration, meaning, that each node controls node configuration of its children nodes, the algorithm should be mounted not only on the server but also on the client modules. For that reason, the algorithm should be implemented in different versions if Operating System of the server and client are different. For example, we developed two different OMP algorithms for Linux and Windows versions.

3.2 Implementation and Environment

As shown Figure 2, the OMP has been designed based on object oriented approach in which the Template design pattern has been applied.

![Figure 2. OMP Architecture](image)

In this OMP architecture, it is tried to make reliable and efficient software with engineering utility including design pattern, design tool and testing tool. Recently, the distributed and recursive approach is popular as Figure 3. For this realization of TBCP, some classes in Figure 4 are introduced for unique object characteristics such as LinuxAlgorithmImp for platform dependent communication, NodeStruct for global or local node configuration information, TBCPPacketInfo for processing packet format, DecisionMaker for realization of TBCP algorithm, SocketTCP for the execution of transport layer.

We implement OMP under Linux (CentOS-4 kernel 2.6.9) for server module using Eclipse3.1 CDT and Windows XP for client module using Microsoft Visual studio 6.0.

3.3 Unit Test

We did unit testing using CPPUnit [8], an open source unit testing tool, for confirming whether the functionalities declared in classes works based on test case or not. The CPPUNIT_ASSERT(), built-in method in CPP/unit, allows tester to build independent testing classes in which target methods can be verified. This improves maintainability and reusability of test cases.
3.4 Integration Test

TBCP algorithm generates various node configurations so that we deliberately produce all cases that the algorithm can generate and test, based on the cases. For example, if a parent node allows for the joining of a new node, which causes exceeding the maximum number of child nodes, tree reconstruction should be executed. There exist two cases of node configurations. First, the new node connects directly and the parent node forces one of the original child nodes to move under remaining children nodes. The other case is that the parent node sends a “go” message to the new node and the new node connects to the designated node. As the node configuration depends on the Round Trip Time (RTT) between nodes, we put appropriate RTT value deliberately for testing all cases that the TBCP generates. The following Figure 6 shows an example of node construction process and packet transaction in case of a joining new node.

```xml
<Message Type="WELCOME_ACK" Value="1005">
  <Node Type="ORIGIN">
    <IP_Address>125.131.73.137</IP_Address>
    <RTT>4</RTT>
  </Node>
</MESSAGE>
```

Figure 5. Unit test's result

Figure 6. Testing TBCP Join Procedure

4. Conclusion

This paper describes the experience through the overall software development process starting from requirement elicitation to implementation and testing while we execute the Overlay Multicast Protocol project.

The algorithm of OMP differs according to the usage of OMP. Therefore, the architecture of OMP project has been designed based on the client’s decision of the quality attributes that the client needs the most. In the implementation phase, business constraints such as patent and maintainability issues were also considered. In addition, technical constraints, for example, the different platforms for server and client modules were resolved by using a template design pattern. CPPUnit, an open source JUnit testing tool, increased the testing performance by supporting an independent testing platform. The contribution of this paper is that we present a solid software development approach where we try to solve the problems through the application of software engineering technologies. Also, our approach suggests a software development methodology, client-oriented documentation and code that give an idea of a sound software engineering approach to software engineers.

REFERENCES

[8] [http://sourceforge.net/projects/cppunit](http://sourceforge.net/projects/cppunit)
[11] [http://www.es.cornell.edu/People/eps/meridian/overview.php](http://www.es.cornell.edu/People/eps/meridian/overview.php)
[12] [http://esm.es.cmu.edu/](http://esm.es.cmu.edu/)
Figure 3. TBCP Sequence

Figure 4. OMP Sequence Diagram