Chapter VII
Peer-to-Peer (P2P) Network Security:
Firewall Issues

Lu Yan
University College London, UK

INTRODUCTION

A lot of networks today are behind firewalls. In peer-to-peer (P2P) networking, firewall-protected peers may have to communicate with peers outside the firewall. This chapter shows how to design P2P systems to work with different kinds of firewalls within the object-oriented action systems framework by combining formal and informal methods. We present our approach via a case study of extending a Gnutella-like P2P system (Yan & Sere, 2003) to provide connectivity through firewalls.

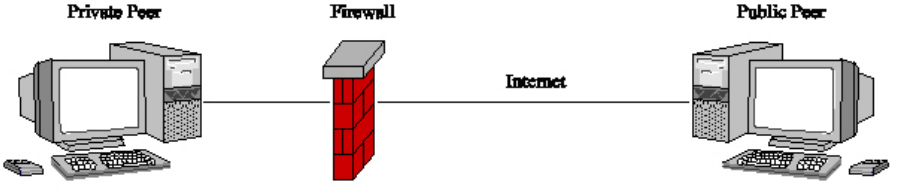
PROBLEM DEFINITION

As firewalls have various topologies (single, double, nested, etc.) and various security policies (packet filtering, one-way-only, port limiting, etc.), our problem has multiple faces and applications have multitude requirements. A general solution that fits all situations seems to be infeasible in this case. Thus we define the problem as shown in Figure 1: How to provide connectivity between private peers and public peers through a single firewall?

We select the object-oriented action systems framework with Unified Modeling Language (UML) diagrams as the foundation to work on. In this way, we can address our problem in a unified framework with benefits from both formal and informal methods.

Action systems is a state based formalism. It is derived from the guarded command language of Dijkstra (1976) and defined using *weakest precondition* predicate transformers. An action, or guarded command, is the basic building block

Figure 1. Problem definition



in the formalism. An action system is an iterative composition of actions. The action systems framework is used as a specification language and for the correct development of distributed systems.

Object-oriented (OO)-action system is an extension to the action system framework with OO support. An OO-action system consists of a finite set of classes, each class specifying the behavior of objects that are dynamically created and executed in parallel. The formal nature of OO-action systems makes it a good tool to build reliable and robust systems. Meanwhile, the OO aspect of OO-action systems helps to build systems in an extendable way, which will generally ease and accelerate the design and implementation of new services or functionalities. Furthermore, the final set of classes in the OO-action system specification is easy to be implemented in popular OO languages like Java, C++ or C#.

In this chapter, however, we skip the details of semantics of action systems (Back & Sere, 1996) and its OO extension (Bonsangue, Kok, & Sere, 1998).

GNUTELLA NETWORK

Gnutella (Ivkovic, 2001) is a decentralized P2P file-sharing model that enables file sharing without using servers. To share files using the Gnutella model, a user starts with a networked computer A with a Gnutella *servent*, which works both as a server and a client. Computer A will connect to another Gnutella-networked computer B and then announce that it is *alive* to computer B. B will in turn announce to all its neighbors C, D, E, and F that A is alive. Those computers will recursively continue this pattern and announce to their neighbors that computer A is alive. Once computer A has announced that it is alive to the rest of the members of the P2P network, it can then search the contents of the shared directories of the P2P network.

Search requests are transmitted over the Gnutella network in a decentralized manner. One computer sends a search request to its neighbors, which in turn pass that request along to their neighbors, and so on. Figure 2 illustrates this model. The search request from computer A will be transmitted to all members of the P2P network, starting with computer B, then to C, D, E, F, which will in turn send the request to their neighbors, and so forth. If one of the computers in the P2P network, for example, computer F, has a match, it transmits the file information (name, location, etc.) back through all the computers in the pathway towards A (via computer B in this case). Computer A will then be able to open a direct connection with computer F and will be able to download that file directly from computer F.

UNIDIRECTIONAL FIREWALLS

Most corporate networks today are configured to allow outbound connections (from the firewall protected network to Internet), but deny inbound connections (from Internet to the firewall protected network) as illustrated in Figure 3.