Model Applications

Table of contents

1. Introduction

The applications discussed here are not necessarily XLattice project objectives. Rather they are "gedanken" applications, applications which XLattice should be able to support.

The objective is to sketch out the applications in sufficient detail to understand how they might best be implemented, and then give a rough proof that they could be built on top of XLattice with reasonable likely performance. Consideration of the need to serve these model applications should also force an appropriately simple design.

It should be possible for a node to participate in more than one of these activities at one time, without any unintended leakage of information from one to another.

1.1. The Model Applications

- Realtime Strategy Game
- Business Messaging System
- Anonymous P2P File Sharing System
- Universal Chat Facility

1.2. Common Requirements

A first analysis of the model applications listed above shows all of them using the common elements set out below.

command line interface	This connects a user's keyboard and screen to the console. In a Java implementation this would be part of the runner which launches the other elements (console, node, etc) in one or more virtual machines (JVMs).
(virtual) console	Interfaces a single user to one or more nodes, providing an input stream to and an output stream from each.
clock	All of the applications require a clock. In one it is critical that all participating nodes share the same absolute time; in other words the clocks must all be synchronized with the real time (UT) to some degree of accuracy. In another, it is critical that all nodes share the same time to some degree of accuracy, but it is not critical that this shared time be synchronized with the real external time. In the other two applications, it is not critical that clocks be synchronized either between nodes or with the real clock time.
node	The key XLattice component. A node is

	controlled by a console, has an identity and an address, has a local file system, is capable of authenticating messages and other cryptographic operations, is capable of being a member of one or more lattices, and can send and receive messages over a number of connections using a variety of transport protocols.
identity	The identity of a node: essentially a public key pair unique to the node.
address	The lattice ID of a node plus an associated address. This is usually a physical address (for example, IP address plus port number) but can be anothe lattice ID.
lattice	An ordered set of the nodes participating in a lattice. A node may belong to more than one lattice.
local file system	That portion of the platform's memory and hard drives which are reserved to XLattice. Persistence of in-memory files is not guaranteed; the machine can be turned off or crash. Files on hard drives are assumed to persist: if you turn the machine off and back on, they will still be there.
authentication	Responsible for verifying the identity of another node using public key cryptography.
crypto	Encrypts or decrypts messages or files. Usually supports more than one protocol.
message	The fundamental unit of communication. Messages may be sent to one or more nodes using one or more transport protocols. They may be encrypted. And they may be routed, that is, forwarded to one or more other nodes, possibly with protocol translation (for example, a message might arrive over tcpip and then be forwarded using smtp).
connection	The connection between two nodes over which messages pass. A connection always has two endpoints and an associated transport protocol.
transport	The protocol used for transporting messages over a connection. At least the following four transport protocols are assumed to be available to all applications.
tcpip	The standard Internet protocol for reliable

	communications, TCP/IP. Reliable means that either eventual delivery of the message or notice of failure is guaranteed.
udp	The Internet's datagram protocol, used for unreliable communications. Unreliable means that delivery is made on a best-effort basis, without notification in case of failure.
smtp	SMTP, the email protocol.
http	HTTP, the protocol used by the World Wide Web. This is a client/server protocol. One of the endpoints will be acting as a Web server, possibly running on the standard port (80), and the other will be acting like a browser.

1.3. Interoperability

As mentioned above, it should be possible for a user to run more than one XLattice application simultaneously, without unintended leakage of information from one application to another.

It is highly desirable that it also be possible to pass information from one application to another in a controlled way.