

CYBERNET

OpenSkies

Networking Engine

Introduction



www.openskies.net

MMPG
MASSIVE
MULTIPLAYER
ONLINE GAMING

Cybernet Real Time Intelligent Routing Technology

Introduction

The level of traffic across the Internet now exceeds the capability of the networks to serve web content quickly. As a result, demand for faster content delivery methods such as caching have been growing dramatically. Although the available caching technologies solve much of the needs for faster static content delivery, they are not sufficient for the delivery of non-cacheable dynamic content.

Cybernet has developed a patent pending, distributed network server technology for the real-time data transfer of dynamic network content. Our Real Time Intelligent Routing technology provides a low cost, high performance solution that is easy to implement. Real-time on-line applications for this technology include:

- **Multi-player games**
- **Stock market interactions**
- **Gambling**
- **Internet-based classes**
- **Chat rooms**
- **Video Teleconferencing**

The majority of web traffic is composed of one-to-one or one-to-many interactions. However, the growth of the online gaming business and chat rooms has shown that there is a growing demand for a more interactive experience on the Internet. This demand for interactivity has already grown to the point where the national networks are no longer be able to handle the traffic and users are staying away due to poor performance. Handling the demand for an interactive Internet requires novel approaches to increase performance and provide a better quality of service to the end users.

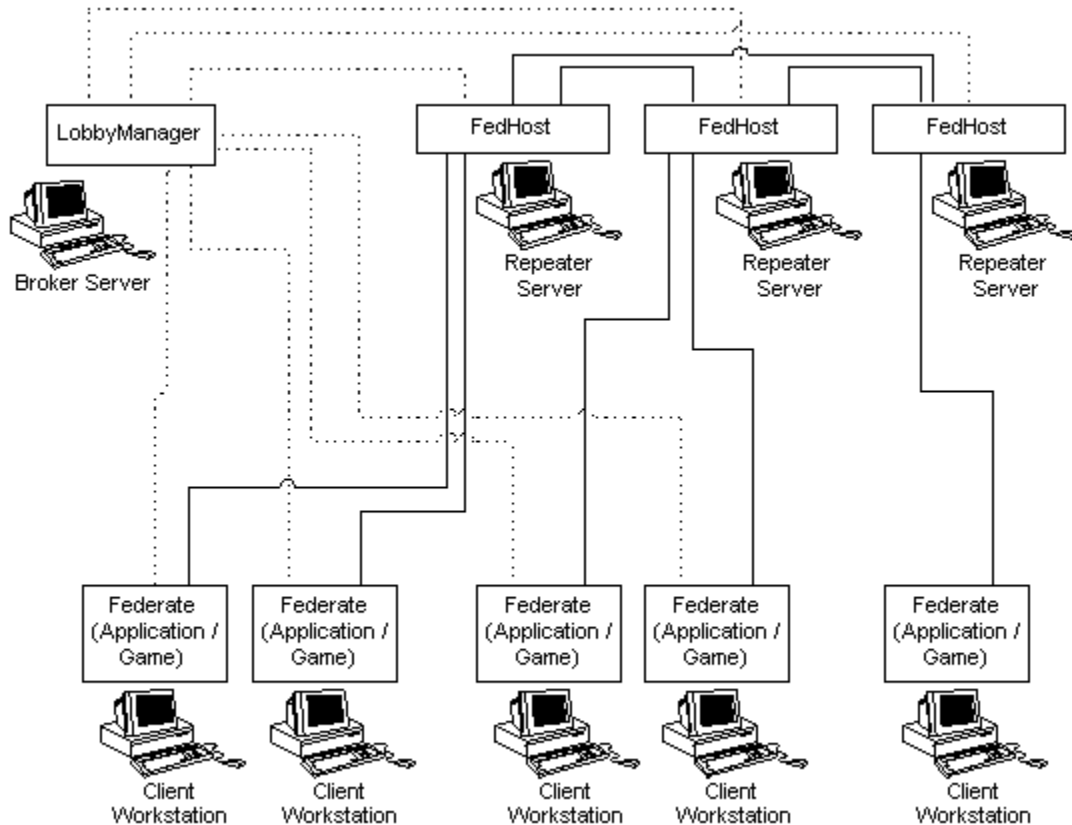
Cybernet's Real Time Intelligent Routing Technology meets this need. Our technology provides a solution for increased performance where there are many-to-many interactions in real-time. Cybernet is focused on bringing this technology to the online gaming business. By providing intelligent routing capabilities that service non-cacheable requests across the Internet, we can provide a faster and better quality of service to the end user in many-to-many interactions.

Cybernet's technology takes the concept of caching to the next level. While caching distributes content across the Internet to reduce load on a web server, our intelligent routing technology reduces the proliferation of real-time traffic across the backbone. In the case of interactive gaming, user requests for data from all of the other player data creates an N^2 problem of data transfer across the Internet. By creating a distributed network running our intelligent routing software, we can remove much of the redundancy and reduce the bandwidth load by 25% to 90%.

In addition, this software architecture provides the ability to reduce the overall bandwidth to the client by culling out data that the client does not necessarily need. An example of this comes in the form of interactive TV. Interactive TV systems provide the ability to watch a sporting event from multiple cameras. Typically, each of these video streams are sent to the user at once and the user chooses which stream to watch at the client. Our culling technology determines which streams the user is actually watching and requests from the server only the data for those particular streams. In an example where there are 10 different camera angles and the user is only watching one, we can reduce the bandwidth load by 90%. Openskies also reduces the bandwidth from the client by performing the required routing on external servers. A client who is currently visible from 20 other clients need only send one update rather than being burdened by having to send 20 of them. This is especially important for those players with slower Internet connections.

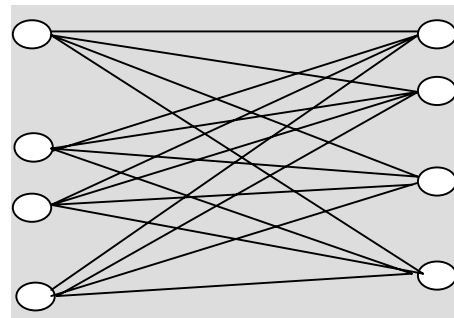
Technology

Cybernet's Real Time Intelligent Routing technology uses a software-based system running across a distributed server network. The clients (called *federates*) connect to one of the distributed servers (called *FedHosts*). These FedHosts then act as traffic cops, acting in concert to route the data where it needs to go.



The intelligent routing software actually recognizes what traffic is going where and who is subscribing to it. So instead of a 100 player game having 10,000 different streams between the players, our network consolidates these streams across the network and can reduce the overall bandwidth by an order of magnitude.

The technology behind this service is based on decades of research by the U.S. military, which have been the predominant researcher in multiplayer simulations for the past 20 years. The U.S. military has developed a distributed network software architecture called *High Level Architecture (HLA)* that provides for a much more scalable network system than any previous or currently developed system. By adapting this architecture, we have created a patented distributed server system that supports massive multiplayer games, which would otherwise be impossible. In addition, this technology will allow us to meet the



Number of Connections for a typical online, 8-player game

bandwidth needs of many other types of multi-participant interactions such as video teleconferencing.

Introductory Distribution

We have assembled this distribution to describe and demonstrate the Openskies technology. All of the Openskies SDK documentation may be found at <http://www.openskies.net/download/download.shtml>. The documents included in this package are the following:

1. **Introduction** - This Document. (<http://www.openskies.net/files/Introduction.pdf>)
2. **Openskies MMPOG** – Describes the various components of the Openskies Massive-Multiplayer Online Game SDK at a high level. This includes a description of the server components as well as the APIs used to network-enable your application using the Openskies SDK. (http://www.openskies.net/files/Openskies_MMPOG.pdf)
3. **Openskies Network Architecture** – Describes the communication between the various Openskies modules in more detail. (http://www.openskies.net/files/Openskies_Network_Architecture.pdf)
4. **Quickstart** – a simple step-by-step guide designed to help you get the Openskies examples up and running. (<http://www.openskies.net/files/Quickstart.pdf>)
5. **Quickstart Server** – a simple step-by-step guide designed to help you get the Openskies servers running (useful for licensees). (http://www.openskies.net/files/Quickstart_Server.pdf)
6. **Openskies IET Guide** – This document is written for developers. It guides the reader through the creation of a simple “HelloWorld” application using the Openskies IET (*Import/Export Table*) API. It also serves as programmer guide and reference for the IET interface. This distribution also includes the complete VC++ 6.0 project file for the HelloWorld sample application. (http://www.openskies.net/files/Openskies_IET_Guide.pdf)
7. **Openskies P2PS Guide** – Written for developers, this document is a programming guide and reference for the P2PS (*Point-to-Point Switch*) API. This distribution also includes a complete VC++ 6.0 project file for a sample application, SwitchTester, which uses this API. (http://www.openskies.net/files/Openskies_P2PS_Guide.pdf)
8. **NetMAXOS Guide** - This document guides you through the setup of the various server components of the Openskies system. These components reside on a Linux-based NetMAX system and are responsible for routing data, managing users, authentication, and database. (http://www.openskies.net/files/NetmaxOS_Guide.pdf)
9. **Openskies Culling Guide** – This document is the programmer guide for writing culling rules that reside on the FedHost servers. This distribution also includes the source, headers, and makefile for a Linux OS sample culling module. (http://www.openskies.net/files/Openskies_Culling_Guide.pdf)
10. **Openskies Security Guide** – This document is the programmer guide for writing authentication modules that reside on the FedHost and LobbyManager servers. This distribution also includes the source, headers, and makefile for a Linux OS sample authentication module. (http://www.openskies.net/files/Openskies_Security_Guide.pdf)
11. **DB_Connection Guide** - This document provides a brief description of the functionality provide by the CDB_Connection class which is provided in the Openskies SDK as source code. This class provides a communication library to the Openskies Database Server which allows the client application to query and update user and authentication information. (http://www.openskies.net/files/DB_Connection_Guide.pdf)

12. **Openskies Database Guide** – This document describes the various components included in the Openskies distribution for maintaining a database. The example database and communication includes support for authentication, user accounts, and network traffic monitoring/archiving. (http://www.openskies.net/files/Openskies_Database_Guide.pdf)
13. **Netmax™ Installation – Openskies Addendum** – This document describes additional steps you will need to take to install your Netmax with Openskies SDK. (Included in licensee distribution only)
14. **Netmax™ Professional Suite - User Manual** - This is the user manual for the Netmax Professional Suite upon which Netmax/Openskies is based. (Included in licensee distribution only)
15. **Openskies HLA Guide** - If you want to learn how to use the OpenSkies HLA API for your application, start with this guide. This document will show you how to integrate the Openskies SDK into your application.
(http://www.openskies.net/files/Openskies_HLA_Guide.pdf)
16. **OpenSkies HLA Extensions** – If you are C++ fluent and ready to get into the guts of the HLA programming, first read the DMSO distributed HLA references and then read this document which details Openskies' HLA supported function, unsupported functions, and extensions from standard HLA specification.
(http://www.openskies.net/files/Openskies_HLA_Extensions.pdf)
17. **Other HLA References** - If you are curious where this model started, or have a question on the HLA architecture, take a look through any or all of these Acrobat format HLA documents:
 - **M&S HLA - Federate I/F Spec.** – Defines the HLA Interface Specification.
 - **HLA OMDT Guide** - Development Tool User's Guide.
 - **HLA OMT Specs** - Object Model Template Specifications.
 - **HLA RTI 1.3v4 A** - HLA Runtime Infrastructure.
 - **HLA RTI 1.3v4 B** - HLA Runtime Infrastructure.
 - **HLA RTI 1.3v4 C** - HLA Runtime Infrastructure.
 - **HLA FED Specs** - Federation Executions Details Specs.
 - **HLA Rules** - High Level Architecture Rules.