

XPN Protocol Overview

XPN is a Peer-To-Peer (P2P) network protocol which operates at the OSI¹ Session Layer. It provides Instant-Messaging, File Transfer, Shared Directory Browsing and is fully routable. The origin of its name lies in "e**X**tensible **P**acket **N**etwork", an acronym reflecting the dynamic nature of each information 'packet' on the network.

Structure

XPN operates on a Peer-To-Peer model, but functions just as well in a Client-Server situation. Any computer which processes XPN packets is defined as a "node". On a fundamental level, the network topology is 'flat', i.e. all nodes are equal in terms of their overall role of communicating with and maintaining the integrity of information and other nodes on the network. The protocol does allow for distinct roles such as "Client" and "Server" to be loosely defined, in which a particular node might give preferential bandwidth access to local nodes, or function as an authentication resource (a server role).

In most instances communication is two-way, i.e. some form of confirmation signal is sent to acknowledge the receipt of data or provide a requested resource to the source node. We shall examine the example of a *WHOIS* packet to illustrate this relationship:

Node 1	Transmission	Node 2
Sends <i>WHOIS</i> request	→	Processes request for information
Stores information about Node 2	←	Sends <i>INFO</i> reply

In a real world scenario Node 1's *WHOIS* packet would be transmitted to all connected nodes, prompting multiple individual *INFO* responses.

Syntax

The syntax of the protocol was designed to be human-readable and language-independent (i.e. no specific programming language constructs except for the concept of a string variable and an array are used). This choice was justified on the basis of the success of the commonly-used HTTP Protocol. Concepts such as the *Method*, *Field*, *Data* packet structure are retained. The layout of each packet is as follows:

```
METHOD  
<Field 1>  
<Field 2>  
...  
<Data>
```

The HTTP Request/Reply paradigm was used as the basis for data exchanges in which a single Request/Reply was required, such as a *PING* request. As such each

¹ http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci212725,00.html

packet includes in its header² both a *Source* and *Destination* field, so the receiving node can correctly address its reply packet.

Example

The first packet sent by a node joining the network is usually a *WHOIS* packet, sent with the destination *BROADCAST*. When a node receives a packet marked for *BROADCAST*, it forwards it to all connected nodes, excluding the node from which it was received. This ensures the packet permeates every node on the network, which in the case of a *WHOIS* packet is essential in determining the active users.

An example packet is may look like:

```
WHOIS XPN/2.0
Destination: BROADCAST
Source: D538A3EA7D794A84BC2774E2DA065F16
ID: A4E116E4DD29440997FD89C97D095BF1
TTL: 10
Hops: 0
Content-Length: 0
```

To which the reply is:

```
INFO XPN/2.0
Destination: D538A3EA7D794A84BC2774E2DA065F16
Source: 09F698DEBDC645E795469DA77F311FB2
ID: D2109A3DD42245A1AC210012F7713B7F
TTL: 10
Hops: 0
Name: Test Server
Type: Server
DKey: 8139
Address: 80.194.88.110:301
Content-Length: 0
```

In this instance, the method is either *WHOIS* or *INFO*, which identifies the packet type. *Source* and *Destination* fields are populated with the unique GUID³ of the source and destination nodes, and *TTL* (Time To Live) and *Hops* provide information relating to the individual packet's path through the network.

² A portion of metadata which precedes the packet's actual data or 'payload'

³ **Global Unique IDentifier**. A technology usually embedded in the operating system which generates a unique text string to identify objects.