Protocol description for Fsync (Files Synchronization) project.

This document contains the protocol description for FSync project. At first we consider the transport layer. Then we describe the message formats and basic data types.

The Fsync is used for files synchronization in P2P network. The protocol is fully encrypted and utilises the SSL protocol as the transport. The two side certificate verification is used for nodes authentication purposes (see [1] for details).

The Fsync protocol is implemented over the SSL layer. The following basic types are used in Fsync protocol (Table 1).

Table 1 FSync basic types

Type	Size (bytes)	Format
UINT8	1	Unsigned byte
UINT16	2	Unsigned short
UINT32	4	Unsigned integer
UINT64	8	Unsigned 8 bytes integer
BOOL	1	true (0) or false (1)
UUID	16	16 unsigned bytes
STRING	4 bytes + string length	4 bytes length + N bytes string

The FSync messages description are shown in table below (Table 2).

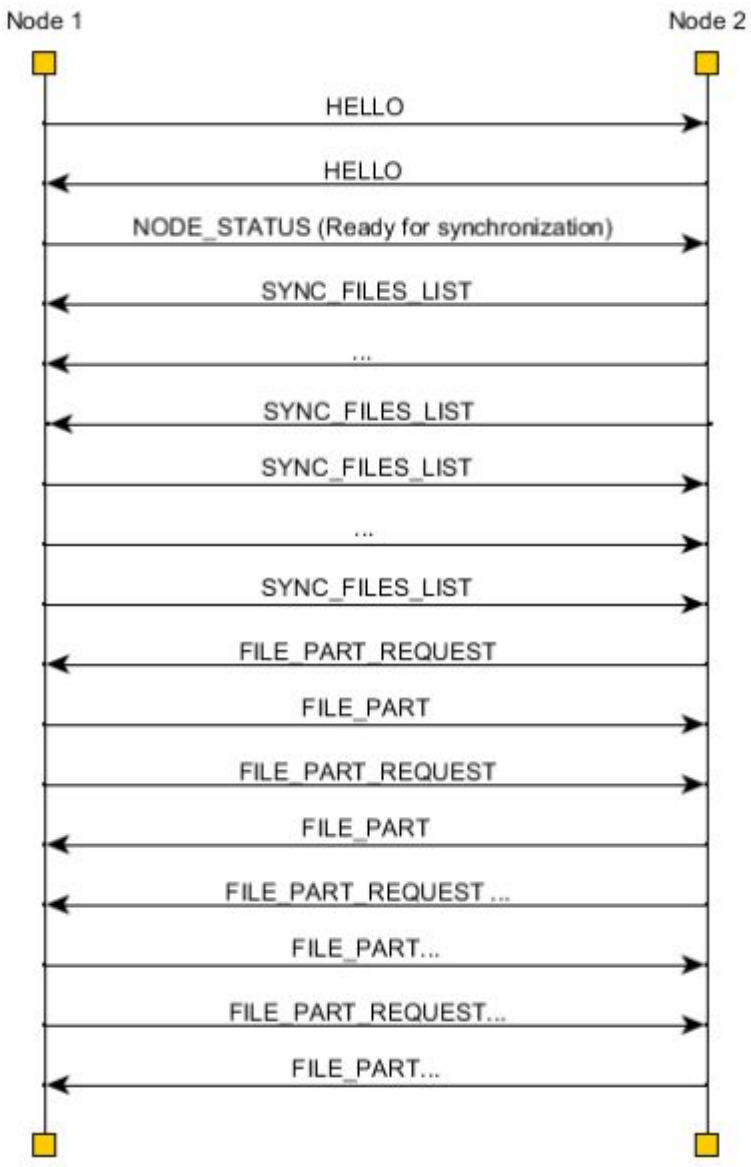
Table 2 FSync messages description

Message	Description	Message format
HELLO	Handshake. This message is used for node UUID and protocol version notification.	HELLO = UUID UINT32 • UUID - node ID • UINT32 - protocol version
NODE_STATUS	Node status notification. Message is used for node	NODE_STATUS = UUID

	status notification.	UINT32 ● UUID - node ID ● uint32_t - node status
SYNC_FILES_LIST	Files list notification. This message is used for files list exchange between different nodes.	SYNC_FILES_LIST= UUID BOOL UINT8 { FILE_INFO } ● UUID - node ID ● BOOL - indicates the last part of the files list. The list of files is transmitted as the several SYNC_FILES_LIST messages. For last message this flag must be False. ● UINT8 - files number in list. ● Files list. FILE_INFO = UINT32 STRING { UINT8 }16 UINT64 BOOL ● UINT32 - unique file identifier for node. ● STRING - File path. ● { UINT8 }16 - MD5 digest. ● UINT64 - file size ● BOOL - flag indicates that the file exists. (True - file is exist, False - file isn't exist)
FILE_PART_REQUEST	File part request.	FILE_PART_REQUEST = UUID UINT32 UINT32 ● UUID - node ID ● UINT32 - file id ● UINT32 - file block number. Files are transferred in 4096-byte blocks.
FILE_PART	File part.	FILE_PART = UUID

		UINT32 UINT32 UINT16 { UINT8 } • UUID - node ID • UINT32 - file id • UINT32 - file block number. • UINT16 - file block size (Maximum allowed value is 4096 bytes) • { UINT8 } - file content

Messages sequence diagram.



References

1. The Secure Sockets Layer (SSL) Protocol Version 3.0