Core Flight System
Software Bus Networking Application
Design As Built

Christopher.D.Knight@nasa.gov

(650) 604-3471

NASA Ames Research Center

Intelligent Systems Division (Code TI)

Design (1)

SBN is a cFS application that:

- **Connects point-to-point with other SBN applications on other multiple cFS software busses together such that messages sent by an application on one bus can be received by an application on another bus.**
- **Has a modular network architecture (TCP, UDP, Serial, SpaceWire, etc.) to connect peers and supports mixed-mode peer networks.**
- **Utilizes an “announce” and “heartbeat” protocol to provide network state awareness to the SBN application.**
- **Remaps and filters outgoing messages (configured by table.)**

Design (2)

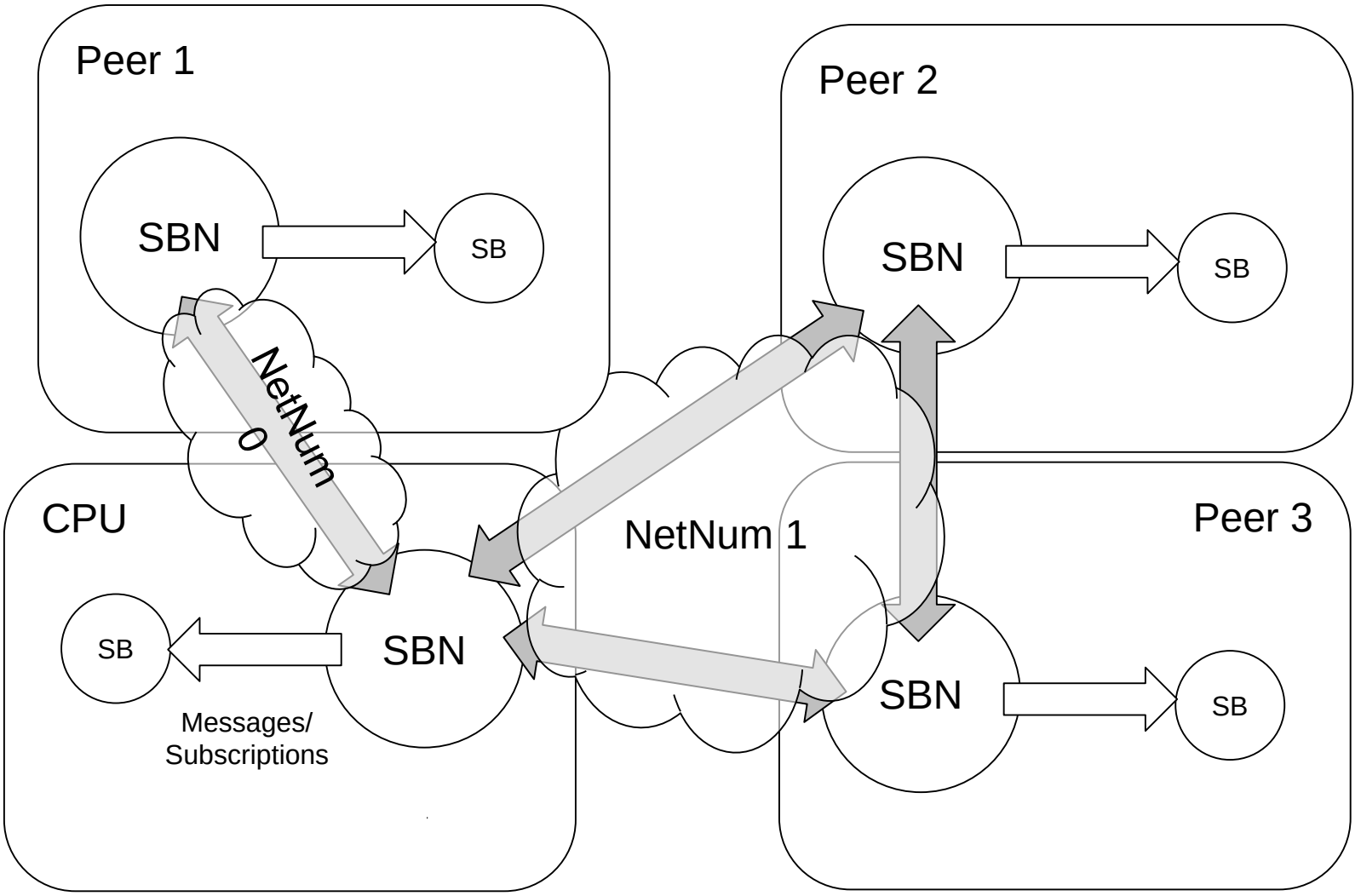
SBN is a cFS application that:

- **Subscribes to the CFE_SB_ALLSUBS_TLM_MID and sends a CFE_SB_SEND_PREV_SUBS_CC to receive all existing subscriptions at startup.**
- **Subscribes to the CFE_SB_ONESUB_TLM_MID message that informs SBN when a local application has (un)subscribed.**
- **Polls pipes and network connections periodically (either via SCH command or a timeout.)**
- **Ensures all SBN and CCSDS headers are big-endian over the wire.**

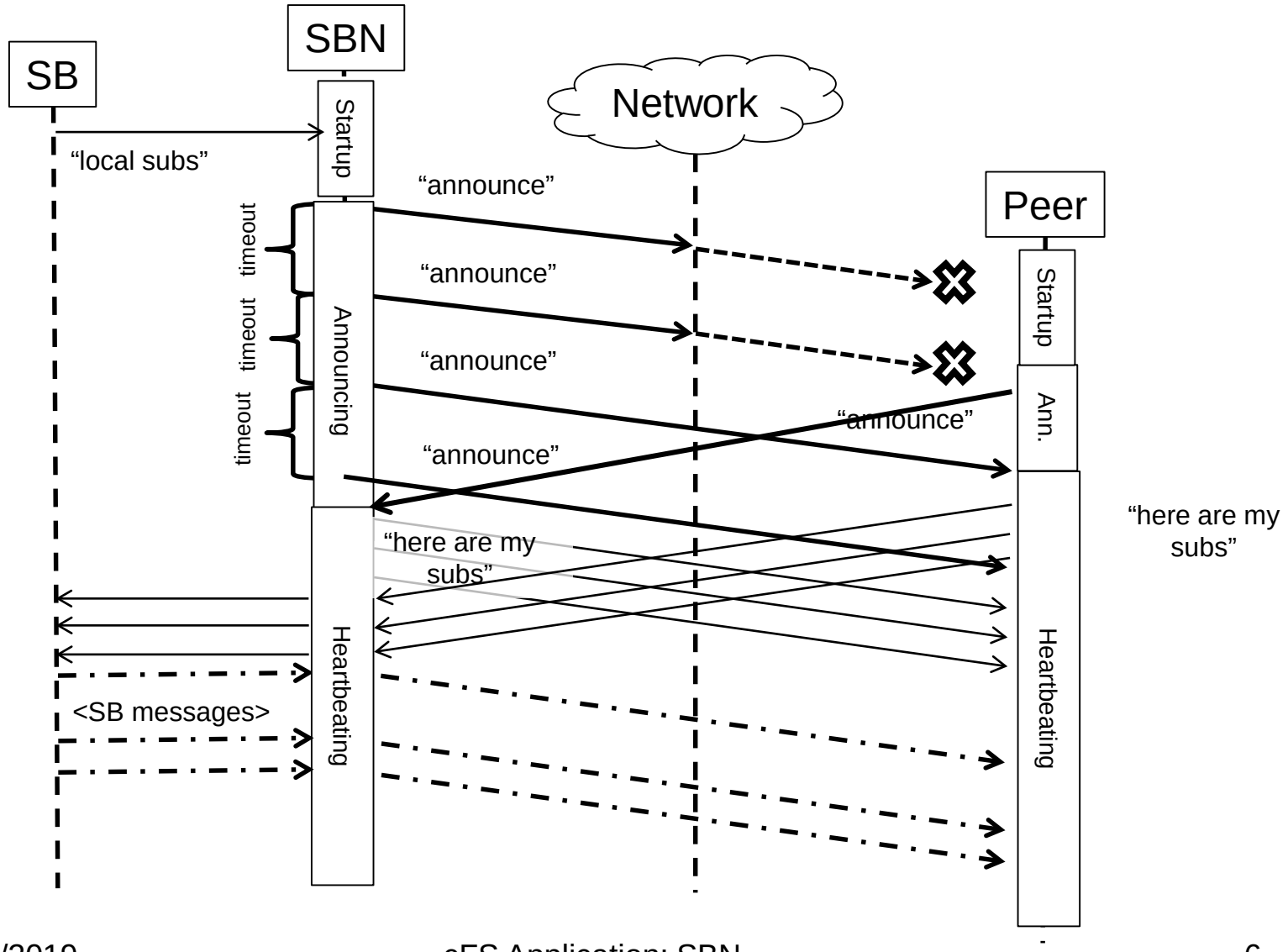
Concerns/Future Developments

Issue	Fix
Chatty protocol of announcing and heartbeating.	Push protocol chatter down to modules.
UDP module polls socket.	Switch to select or multi-threading.
MID collisions likely in large networks.	Utilize remapping/filtering.
SBN “star network” only, lacks any forwarding/routing capability.	Develop forwarding/routing architecture, or develop/integrate separate app (CI/TO?)
cFS SB limits the total number of MIDs to 256. Large SBN networks will need significantly more.	Increase SB limits. Investigate impacts.
SBN subscribes to all MIDs of all other subs, plus subs for all peers.	Increase limits, develop filtering to limit subs for peers, “subscribe all” to receive all SB messages (need deep pipe.)
SBN configuration is error-prone.	Redesign, use OSAL configloader.

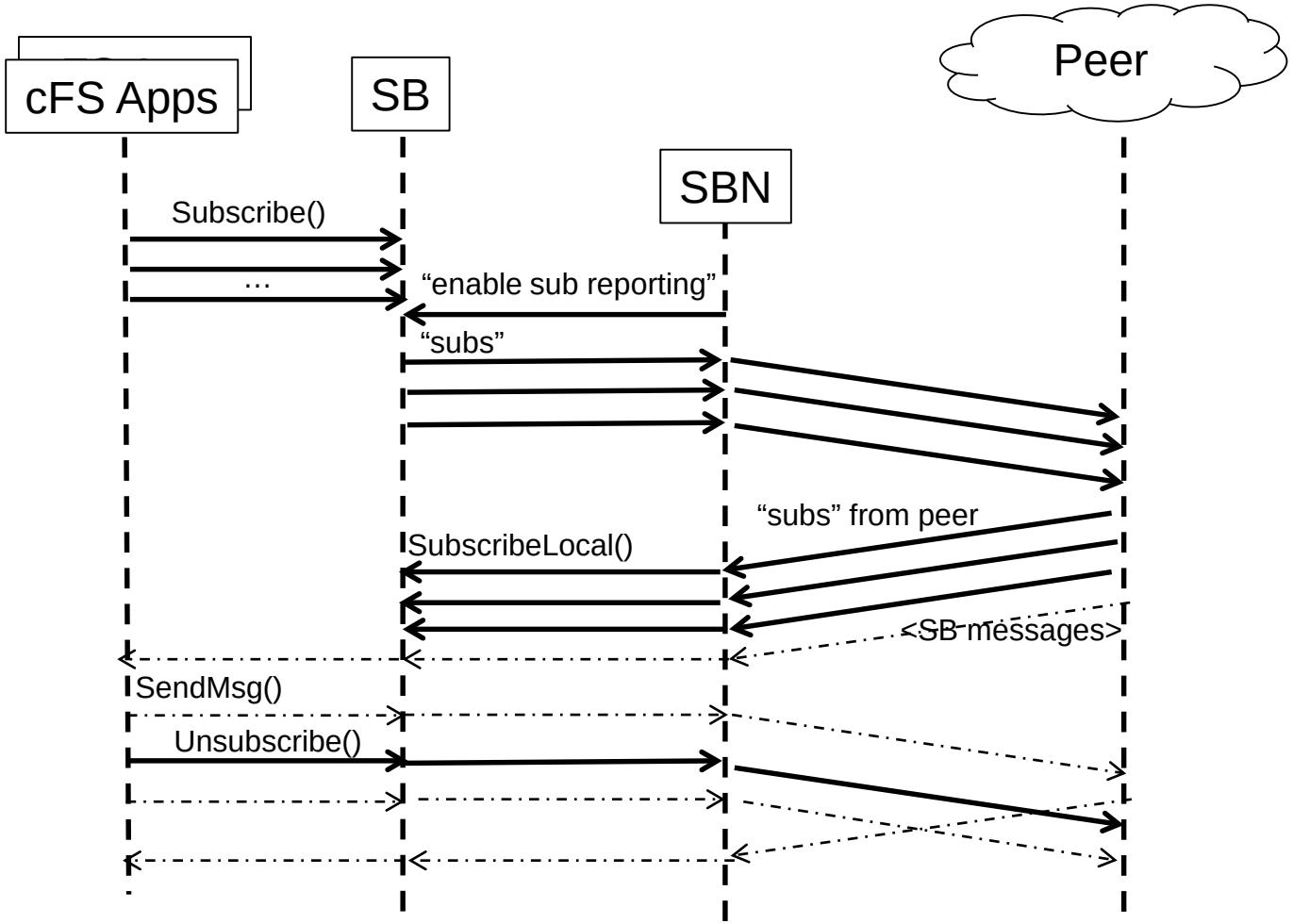
Context Diagram



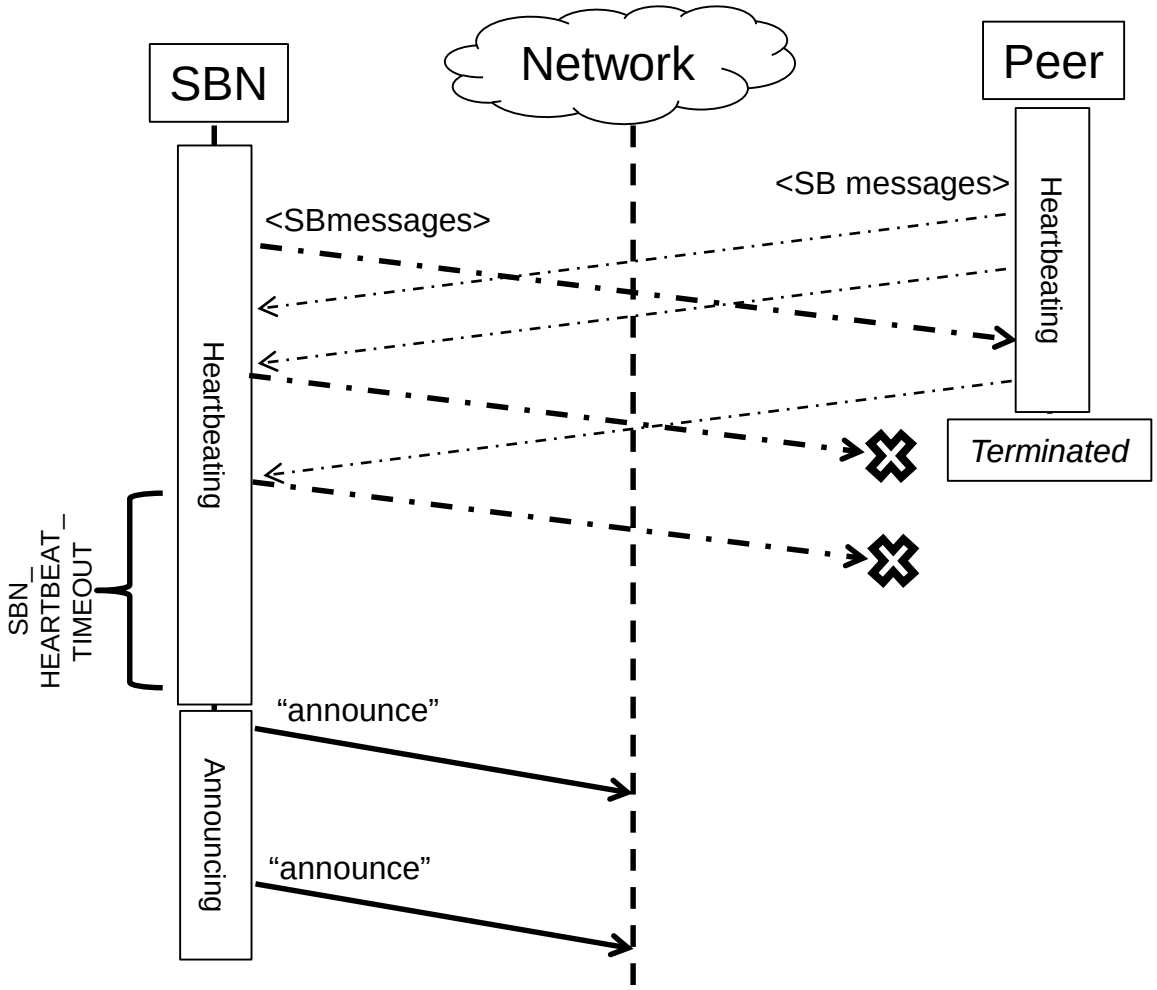
Sequence: Setup



Sequence: Packet Exchange



Sequence: Teardown



Configuration Files: Module Data

Parameter	Type	Description
Protocol ID	int	The identifying number for this protocol.
Protocol Name	char[50]	The name of this protocol.
Module Path	char[50]	The path to the module file.
Module Operations Symbol	char[50]	The symbol in the module's symbol table containing the methods the module provides.

The configuration file is modeled on the ES startup script, one record per line, fields separated by commas and lines terminated with semicolons. For example:

```
1, UDP, /cf/sbn_udp_module.so, SBN_UDP_Ops;
2, TCP, /cf/sbn_tcp_module.so, SBN_TCP_Ops;
6, Serial, /cf/sbn_serial_module.so, SBN_Serial_Ops;
```

Configuration Files: Peer Data

Parameter	Type	Description
CPU Name	char[SBN_MAX_PEERNAME_LENGTH]	The CPU name of the node (needs to match CFE_CPU_NAME.)
CPU ID	uint32	The ID of the node (needs to match CFE_CPU_ID.)
Protocol ID	int	The protocol ID for the module to connect to this node.
Spacecraft ID	uint32	The Spacecraft ID
QoS	uint8	The QoS of the connection.
NetNumber	int	The Network Number (if you have multiple distinct networks sharing peers.)
+++ module-specific parameters (e.g. hostname/port, serial device filename, etc.)		

Configuration Parameters (1)

Parameter #define SBN_*	Default	Description
int SUB_PIPE_DEPTH	256	Depth of the subscription pipe.
int MAX_ONESUB_PKTS_ON_PIPE	256	Maximum number of individual subscription messages on the subscription pipe.
int MAX_ALLSUBS_PKTS_ON_PIPE	64	Maximum number of “all subscriptions” messages on the subscription pipe.
int MAX_INTERFACE_TYPES	6	Max number of interface modules that can be loaded.
int SCH_PIPE_DEPTH	10	Depth of the scheduler pipe.
int STATUS_MSG_SIZE	128	In Hk messages, the module can provide its own data. This is the max size of that block of RAM.
boolean DEBUG_MSGS	undef	If defined, SBN will produce copious debug events.

Configuration Parameters (2)

Parameter #define SBN_*	Default	Description
char * VOL_MODULE_FILENAME	"/ram/SbnModuleData.dat"	Path of the module configuration in the volatile memory.
char * NONVOL_MODULE_FILENAME	"/cf/SbnModuleData.dat"	Path of the module configuration in the non-volatile memory.
int MODULE_FILE_LINE_SIZE	128	Max length of a module file line.
char * VOL_PEER_FILENAME	"/ram/SbnPeerData.dat"	Path of the peer configuration in the volatile memory.
char * NONVOL_PEER_FILENAME	"/cf/SbnPeerData.dat"	Path of the peer configuration in the non-volatile memory.
int PEER_FILE_LINE_SIZE	128	Max length of a peer file line.

Commands

Command (#defined as SBN_<cmd>_CC)	Parameters	Description
NOOP	(none)	General app aliveness test – verifies command handler and event generation.
RESET	(none)	Reset application housekeeping telemetry counters.
RESET_PEER	uint8 PeerNum	Resets the task, stopping and clearing config.
SEND_HK	(none)	SBN sends telemetry containing the current status of the SBN network.
MYSUBS	(none)	SBN sends telemetry containing the local subscriptions that SBN is subscribed to.
PEERSUBS	uint8 PeeerNum	SBN sends telemetry containing the subscriptions the local SBN is aware of for that peer.
SCH_WAKEUP	(none)	SBN waits on wakeup messages from the scheduler and also has a built-in timeout in case SCH is not running.

Housekeeping requests are sent as commands with housekeeping-specific command codes. Responses all are sent as telemetry with the same message ID but the first byte of the response is the command code that made the request.

Housekeeping (1)

SBN_HkPacket_t

Telemetry Point	Data Type	Description
TlmHeader	uint8[CFE_SB_TLM_HDR_SIZE]	The CCSDS header.
CC	uint8	The command code that requested this housekeeping.
<Padding>	uint8 * 3	32-bit align the remainder
CmdCount	uint8	Number of successful ground commands (includes commands from on board sources).
ErrCount	uint8	Number of commands with process errors.
SubCount	uint16	Number of subscriptions for local apps.
EntryCount	uint16	Total number of entries (hosts and peers.)
HostCount	uint16	Number of host entries.
PeerCount	uint16	Number of peer entries.
PeerStatus	SBN_PeerStatus_t * SBN_MAX_NETWORK_PEERS	Details for each peer.

Housekeeping (2)

SBN_PeerStatus_t

Telemetry Point	Data Type	Description
InUse	uint8	Set to !0 when in use.
QoS	uint8	The CCSDS quality of service.
ProtocolId	uint8	The ID of the protocol to use to connect to this peer.
State	uint8	Whether this node is connected (heartbeating) or disconnected (announcing.)
Name	char * SBN_MAX_PEERNAME_LENGTH	The name of the peer.
ProcessorId	uint32	The cFS processor ID of the peer.
SpaceCraftId	uint32	The ID of the spacecraft for this peer.
LastSent	OS_time_t	Last time I sent this peer a message.
LastReceived	OS_time_t	Last time I received a message from this peer.
SentCount	uint16	Number of messages sent to this peer since last reset.
RecvCount	uint16	Number of messages received from this peer since last reset.
SentErrCount	uint16	Number of errors raised when sending to this peer.
RecvErrCount	uint16	Number of errors raised when trying to receive from this peer.
SubCount	uint16	Number of subscriptions sent to me by this peer.
<Padding>	uint8 * 2	32-bit align the next block.
IFData	uint8 * 32	IF-specific private data block.

Housekeeping (3)

SBN_HkSubsPacket_t

Telemetry Point	Data Type	Description
TlmHeader	uint8[CFE_SB_TLM_HDR_SIZE]	The CCSDS header.
CC	uint8	The command code that requested this housekeeping.
PeerIdx	uint8	The index of the peer this is a subscription list for.
SubCount	uint16	The number of subscriptions for this peer.
Subs	CFE_SB_MsgId_t * SBN_MAX_SUBS_PER_PEER	Subscriptions for this peer.

Event IDs

Event ID #define SBN_..._EID	Description
SB	Local software bus
INIT	Application initialization
MSG	SB message
FILE	Configuration (module and peer) file
PEER	Local peer resources (pipes, memory)
PROTO	Network protocol
CMD	Commanding
SUB	Subscriptions