AppNote 4 - Spinnaker Datagram Protocol (SDP) Specification

SpiNNaker Group, School of Computer Science, University of Manchester

Steve Temple - 25 Nov 2011 - Version 1.00

Introduction

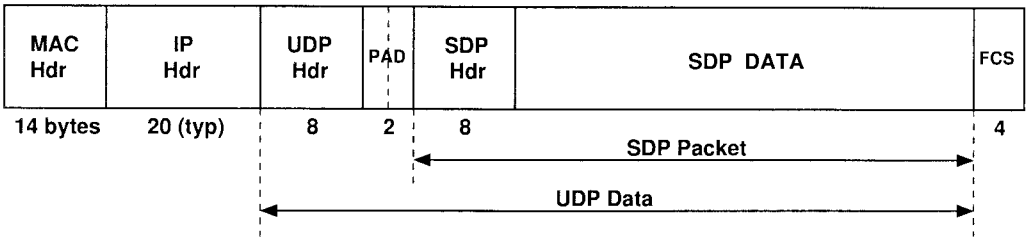
SDP is a protocol for moving blocks of data in a Spinnaker system. It provides a basic, non-guaranteed point-to-point communication method which may be used as-is, or with additional control mechanisms to provide more reliable data transport.

Data is sent in packets or datagrams which can contain up to ~64 kbytes of data. The current implementation limits the amount of data to ~256 bytes in order to mimimise the size of buffers in the Spinnaker chips.

Each SDP packet contains an SDP header which controls the addressing of the packet and a data field which holds the payload. It is necessary to be aware of the length of the packet but this is considered the responsibility of the agent conveying the packet and does not occupy a specific field in the packet.

SDP packets may be conveyed by a variety of mechanisms. For example, between Spinnaker chips they are conveyed as sequences of point-to-point Spinnaker packets. When sent over the Internet, they are usually embedded in UDP packets. Each of the various transit mechanisms will have its own way of conveying the packet length and may also carry out integrity checking on the data.

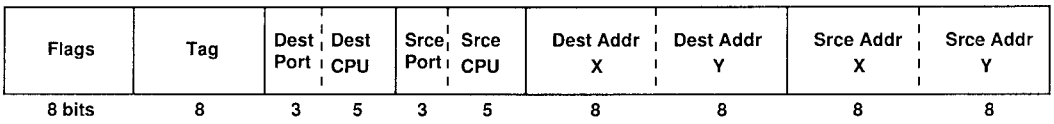
An SDP packet embedded in a Ethernet-based UDP packet is shown in the diagram below.



SDP header

The SDP header is an 8 byte field which is used to route SDP packets. There are 8 fields as shown in the struct definition and diagram below.

```
typedef struct // SDP header
{
  uchar flags;
  uchar tag;
  uint dest_cpu:5;
  uint dest_port:3;
  uint srce_cpu:5;
  uint srce_port:3;
  ushort dest_addr;
  ushort srce_addr;
} sdp_hdr_t;
```



The Flags field is used internally by the SDP protocol code. It should be initialised to either 0x87 or 0x07 depending on whether or not a reply packet is expected.

The Tag field is an IPTag number which is used when the packet is being sent over a network which uses standard IPv4 routing. If the packet is just being sent internally inside a SpiNNaker machine this field is not used and should be set to zero. See below for notes on IPTags.

The Dest Addr field is the point-to-point address of the destination SpiNNaker node. It follows the convention that the X coordinate of the node is in the high byte and the Y coordinate is in the low byte.

The Srce Addr field is the point-to-point address of the sending node.

The Dest CPU field contains the virtual CPU number of the destination CPU. The Dest Port field contains a 3 bit port number. Port 0 is reserved for debugging purposes and ports 1-7 are passed directly to applications. These two field are packed into a byte. The Srce CPU and Port contains similar information relating to the sending CPU.

The value (Port=7, CPU=31) is special in that this is used to indicate to an Ethernet connected node that the packet should be routed out onto the Internet. In this case, the Tag field specifies the IPTag to be used.

Note that all fields must be filled by application software - the source address and port are not supplied automatically.

When a packet arrives at a destination and the application wishes to reply to the sender using the same packet, the Srce and Dest fields are swapped to achieve this.

The bits in the flag byte are documented below though application programs will rarely have to use these other than as described above.

```
// Bits in SDP Flags byte

#define SDPF_REPLY 0x80 // Reply expected
#define SDPF_xxx_40 0x40 // Spare
#define SDPF_SUM 0x20 // Checksum before routing
#define SDPF_DP2P 0x10 // Disable P2P check in routing
#define SDPF_DLINK 0x08 // Disable Link check in routing
#define SDPF_LMASK 0x07 // Link bits mask
```

SDP Length

Where the length of an SDP packet needs to be specified, the value that should be used is the total length of SDP header and SDP data fields. As the header occupies 8 bytes the length should be specified as 8 plus the number of SDP data bytes. The minimum length of an SDP packet is therefore 8.

IPtags

An IPTag is a small integer (0..254) which is used to allow SDP packets to be routed to an IP address and port number. Spinnaker nodes which have an Ethernet interface maintain a table indexed by IPTags

which provides the mapping between a tag and the IPaddr/port pair.

IPTags may be either permanent or transient. Permanent IPTags are explicitly created and removed by a command given to SC&MP either by a host or a SpiNNaker application.

Transient IPTags are created when an SDP packet arrives at a SpiNNaker node via UDP and a reply packet is expected. In this case a transient IPTag is created based on the IP address and reply port of the incoming UDP packet. This is passed to the destination in the SDP packet and quoted in the reply SDP packet. When the reply reaches the Ethernet-attached node the IPTag is used to route the reply UDP packet and the IPTag table entry is deleted at that point.

IPTags can have a timeout associated with them so that if the reply associated with a transient tag fails to arrive the table entry can be reclaimed. The IPTag table in the first SC&MP implementation contains 16 entries of which the first 4 are reserved for permanent IPTags.

Embedding SDP in UDP

When an SDP packet is being conveyed in a UDP packet it is placed in the UDP data field. The length of the SDP packet can be inferred from the length specified in the UDP header and the checksum in the UDP header is used to check the integrity of the SDP packet.

Because most UDP packets are carried over Ethernet, the start of the UDP data field is not 4-byte aligned. In order to make the start of the SDP packet aligned to a 4 byte boundary (which makes processing in Spinnaker rather easier), a 2 byte pad field is placed at the start of the UDP data field.

The first of these 2 bytes is currently used to allow the timeout field of an IPTag created by the reception of the packet to be specified. The second byte is reserved and should be set to zero. Valid values for the timeout byte are 0 to 16 as follows

0	No (infinite) timout
1	10 ms
2	20 ms
...	
N	$10 * 2^{N-1}$ ms
...	
16	327680 ms

Spinnaker Command Protocol

SDP packets are used extensively for conveying commands and responses to those commands around a Spinnaker system. In order to do this, further structure is imposed on the SDP packet by subdividing the data field. The data field is split into 6 fields according to the following struct

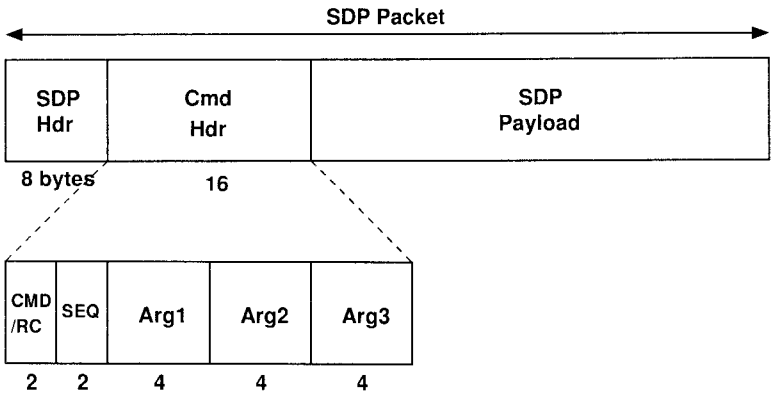
```
typedef struct
{
  ushort cmd_rc; // Command/Return Code
  ushort seq; // Sequence number
  uint arg1; // Arg 1
  uint arg2; // Arg 2
  uint arg3; // Arg 3
  uchar data[SDP_BUF_SIZE]; // User data (256 bytes)
} sdp_cmd_t;
```

The *cmd_rc* field is a code indicating the command that is being specified in the case of a packet conveying

a command. Where the packet is a response to a command, the *cmd_rc* field indicates a return code following execution of the command.

The *seq* field may be used for error checking to detect lost packets and to allow for a retry mechanism.

The fields *arg1*, *arg2* and *arg3* are provided to allow 32-bit arguments or return values to be transported while the *data* field allows arbitrary data structures to be conveyed. At present, this field is limited to 256 bytes.



All Spinnaker cores are expected to receive and respond to SDP command packets sent to them on port 0. A number of commands are already defined to assist with program loading and debugging.

Change log:

- *1.00 - 25nov11- ST* - initial release - comments to *temples@cs.man.ac.uk*