BBN Technical Memorandum No. 1223

Smart Packets (SPKT) Header

Job No. 11822000

September 27, 1999

Prepared for:
Defense Advanced Research Projects Agency
Computer Systems and Technology Office
3701 North Fairfax Drive
Arlington, VA 22203-1714

Prepared by:
GTE
10 Moulton Street
Cambridge, MA 02138
Smart Packets (SPKT) Header
Alden W. Jackson, Wenyi Zhou, and Beverly L. Schwartz

Abstract

This document describes how to encapsulate a BBN Technologies Smart Packets packet. The BBN Technologies Smart Packets project is one project in DARPA's Active Networks program. Smart Packets packets will be sent within an ANEP (Active Network Encapsulation Protocol) packet. ANEP was developed for the Active Networks program; it is described in an IETF experimental draft. There are four types of Smart Packet headers depending on whether the packet carries program code, data, messages from the user program, or errors from the run-time execution environment or security aspects. They are described separately in the following. When SPKT gets a packet whose type is unknown, it forwards the packet on without further evaluation; at the same time, it sends an SPKT unknown type error message back to the source node.

1. Common header

The first 4-byte of all four types of SPKT header is common. The map of this common part is shown below:

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------+
| ver | code | context number | serial number |
+---------------------------------------------+
```

Description:

`ver`:

4-bit version field specifies the version of the Smart Packet header. Whenever the Smart Packets header or the way Spanner (a CISC-style stack based assembly language with high level types we created for the Smart Packets project) is encoded changes radically, the version number will be increased. value = 0 for the first version of SPKT, range: 0 to 15.

`code`:

4-bit of code field is used to differentiate between program code, data, messages, and errors.
0 SPKT Program in Spanner encodings
1 SPKT Data generated from a Smart Packets program
2 SPKT Message generated from the user program
3 SPKT Error and Diagnostic Message. Error messages include errors for run-time interpreter, authentication, and ACL (access control list). Diagnostic message contains information for user debugging purpose.

context number:
8-bit unique number on the source to identify this Smart Packets program, range: 0 to 255. There can be maximally 256 Smart Packets processes running simultaneously on a single machine. The first Smart Packets process running has the context number of 0.

serial number:
16-bit sequence number uniquely identifies a Smart Packets program within a context, range: 0 to 65,535. The serial number increases sequentially starting with number 0 for Smart Packets programs within a given context.

For SPKT Data, Message, and Error packets, the context number and serial number are obtained from the original packet which contains the Smart Packets program that produced the data, message, or error to be sent to the source of the program.

A Smart Packets program can be uniquely identified by the trio: source address (from the ANEP header), context number, and serial number.

2. Common portion of Smart Packets data, message and error packets

A remote site which receives and runs a Smart Packets program can send data, message or error packets back to the source of the program. The restriction of sending these only to the source may be relaxed when there is a better understanding of the security and network traffic implications.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ver</td>
<td>code</td>
<td>context number</td>
<td>serial number</td>
</tr>
<tr>
<td>IP address of source of Smart Packets program</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 4 bytes for IPv4, 8 bytes for IPv6 )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See section 1 for description of these fields.

3. Smart Packet Data packet header

As a Smart Packets program runs, the program can construct a data-filled packet which can be sent back to the source of the program. This describes how to encapsulate a Smart Packets data packet.
Smart Packets (SPKT) Header

Description:

ver: see section 1

code: value = 1 for SPKT Data packet; see section 1

source IP address, context, serial:
all of these come from the packet the Smart Packets program was delivered in; see sections 1 and 2

data:
any bucket of bytes of any length; the program puts data into this area as it executes

4. Smart Packet Message packet header

A Smart Packets program can send a text based message back to the source of the program. Eight bits are dedicated to a system error number. If these bits are set to 0, then there is no system error being reported on.

Description:

ver: see section 1

code: value = 2 for SPKT message packet; see section 1

source IP address, context, serial:
all of these come from the packet the Smart Packets program was delivered in; see sections 1 and 2
syserror number:

8-bit code field is used to specify different SPKT error messages generated from the user program. These messages are from SPKT primitives, function calls, etc.

- No error 0
- Can't access MIB 1
- Type request and return for MIB don't match 2
- MIB doesn't recognize interface number as valid 3
- Can't allocate memory for MIB 4
- MIB table error 5
- Address doesn't correspond to any interfaces 6
- Packet too large to be sent on interface 7
- Interface not valid to send packet on 8
- Unknown error occurred when trying to send packet 9
- No default route to send packet 10

message:
clear-text specified/generated by Smart Packets program

5. Smart Packet Error and Diagnostic packet header

When a program can't be executed, or when the Smart Packets virtual machine can't continue execution, an error message is sent back to the source. In addition, this packet format is used by Smart Packets programs to send diagnostic messages to the Smart Packs program source.

Description:

ver:
see section 1

code:
value = 3 for SPKT error and diagnostic packet; see section 1

source IP address, context, serial:
all of these come from the packet the Smart Packets program was delivered in; see sections 1 and 2

type:
4-bit field specifying type of error or diagnostic message

- No error 0
- Security error 1
- Virtual Machine run time error 2
- Error in received packet 3
- Diagnostic message 4

specific code:
12-bit field specifies more detailed error or diagnostics code within each type.
6. Smart Packet Program code packet header

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>code</td>
<td>context number</td>
<td>serial number</td>
</tr>
<tr>
<td>program length</td>
<td>argument len</td>
<td>RESERVED</td>
<td></td>
</tr>
<tr>
<td>arguments</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>program code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>baggage area</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description:

ver, context, serial:
see section 1

code:
value = 0 for SPKT program code packet; see section 1

program length:
16-bit field specifies the length of the Sprocket program code.

argument length:
8-bit field for the amount of arguments in bytes.

arguments:
the area for Smart Packets program arguments; each argument is separated by a NULL character.

program code:
the area for Smart Packets program code.

baggage area:
the area for storing the data after executing Smart Packets program. The length of the baggage area grows as intermediate nodes add data to the packet, so long as the packet length is smaller than the path MTU.
CONTENTS

1. Common header ............................................................................................................. 1
2. Common portion of Smart Packets data, message and error packets ............................ 2
3. Smart Packet Data packet header ....................................................................................... 2
4. Smart Packet Message packet header ............................................................................... 3
5. Smart Packet Error and Diagnostic packet header ............................................................ 4
6. Smart Packet Program code packet header ..................................................................... 5