Network Woking Group
Internet Draft
Expire in six months

Danny Cohen
Myricom
Craig Lund
Mercury Computers
Tony Skjellum
Mississippi State University
Robert George
Mississippi State University
Thom McMahon
Mississippi State University

May 1998

The End-to-End (EEP) PacketWay Protocol for High-Performance Interconnection of Computer Clusters

<draft-ietf-pktway-protocol-eep-spec-03.txt>

Status of this Memo

Table of Content:

1. Introduction

PktWay is an open family of specifications for inter-networking high performance SANs (System Area Networks) and high performance LANs (Local Area Networks) into computing clusters.

Most modern SANs have much in common, such as high data rates, low message latency and low bit error rates. Such SANs are often packet networks made of point-to-point links with flow control and source routing. Yet these SANs do not provide heterogeneous networking support, and are subsequently incapable of direct inter-communications with other SANs. PktWay's goal is to provide high performance "internetting" of such SANs and of high performance LANs.

The core PktWay protocol comprises the End-to-End Protocol (EEP) and the Router-to-Router-Protocol (RRP). This document specifies the EEP (End-to-End) protocol of PktWay. A companion document ("Specification for the Router-to-Router (RRP) PktWay Protocol") specifies the Router-to-Router protocol of PktWay.

Computing clusters and modern MPPs (Massively Parallel Processing systems) are sets of processors interconnected by high performance SANs. Examples are Intel's Paragon and ASCI-red, CRAY's T3D and T3E,

and IBM's SP2 and SP3. Most modern SANs have much in common, such as high data rates, low message latency and low bit error rates. Such SANs are often packet networks made of point-to-point links with flow control and source routing.

Unfortunately, there is no efficient way to "internet" these SANs - to allow each computing node to have high performance communication directly with any other computing node, in any other interconnected SAN. Hence, there is no way to interconnect such high performance SANs to form as efficient computing cluster as possible.

The objective of PktWay is to provide high performance communication among all the processors in a cluster of tightly coupled heterogeneous SANs. PktWay borrows heavily from the experience and wisdom of IP, with a few modifications needed for high performance. PktWay sacrifices generality and scalability to improve performance.

1a. PktWay and IP

IP is the general solution for "internetting" heterogeneous diverse networks, proven for over 25 years. However, IP was designed for the generality required for Wide Area Networks, without regard to the high performance requirements of tightly coupled systems. In addition, IP was designed to addresses "systems" rather than individual processors in MPPs (as PktWay does). For example, a 9,000 processor system is not expected to be assigned 9,000 IP addresses.

PktWay is slightly below IP in the OSI Reference Model. It has many Level-3 features, like IP, but also can support IP as if PktWay was a Level-2 protocol. Hence, it is below IP. In addition, PktWay supports Level-2 optimizations (such as source routing).

Like IP, as a heterogeneous network layer, PktWay packets are transported by the native data-link layer of each SAN. As a result, PktWay packets are encapsulated with any native routing headers and trailers as required by the local network fabric.

Like IP, PktWay uses routers between its SANs. When an HR (half-router) receives a packet for a destination on its own SAN it forwards that packet directly to its destination. If the packet is for a destination out of this SAN, the HR forwards it to another HR which is en route to that destination.

Unlike IP, RRP defines the communication among the HRs both

intra-router and inter-router. In the IP environment only the intra-router communication is not defined, only the inter-router communication.

Unlike IP, the PktWay routers do not have to pop each packet back to Level-3, and are capable of operating entirely at Level-2, if this operation is requested by the communicating hosts. This Level-2 operation is discussed later in this introduction.

Like IP, the PktWay protocol utilizes the native capabilities of its constituent SANs and routers. PktWay defines neither how each HR maps the network in the SAN to which it is attached, nor how each half-router constructs SAN-headers for each of its hosts. The PktWay protocol also does not define how error-checking is conducted by each SAN (e.g., CRC8, CRC32, CRC64, or anything else). Instead, PktWay assumes that these capabilities are native to each SAN, and defines only how these maps are exchanged, and how these error indications are carried from where they were detected, to the destination node.

Like IP, the PktWay protocol defines neither how routes are selected, nor what corrective actions should be taken in case of faults. Instead, PktWay provides the information needed by the host nodes for devising routes and detecting and circumventing faults.

Like in IP environments, when hosts are powered up they may contact their default half-routers to register themselves and to inquire about other hosts (by name or node capabilities). This registration could be used in support of dynamic discovery procedures. The half-routers may help nodes discover each other (like IP's DNS) and may provide routing alternatives, possibly with different characteristics (e.g., MTU, length, and cost). The PktWay protocol does not specify how to choose among them.

To sum it all up, PacketWay has learned many lessons from IP, but has been heavily optimized for high performance SANs, while IP is the protocol of choice for WANs.

1b. General

PktWay supports resource discovery, by name or capabilities.

PktWay's unit of data is 64-bit long (8 bytes). Hence, a PktWay packet is always a multiple of 8B quantities. PktWay provides hosts with padding as required.

PktWay iself is big-Endian 8B-word based. Hence, the terms "first bit" and "first byte" are equivalent to MSbit and MSByte.

PktWay handles the Little vs. Big-Endian issue for its payload by providing a field in the EEP header which defines the endianness and "the chunk-size" of the data in the payload (Data Block). The intent is that byte-swapping hardware, if any, could be used to invert the endianness of payloads with uniform data elements (e.g., all the data being 32-bit floating point). Although this approach does not address the problems of transporting general structures (e.g., a "struct" of C), it does allows the participation of smart memory cards as PktWay nodes, as well as supporting direct memory access (DMA) operations.

The PktWay protocol is designed to allow wormhole (or "cut-through") forwarding, in which a router can start forwarding packets after receiving the first four bytes only (that include the PktWay-protocol version, priority, and the destination-type) without waiting for information that may not be needed for the packet forwarding task. This is unlike IP routers that receive the sender address before receiving the destination address, even though the former is not always needed whereas the latter is.

PktWay's addresses are short (23 bits) because, unlike IP, PktWay is not designed for global operation. The amount of state that is stored in the half-routers per node (type, name, paths, capabilities, etc.) makes it impractical for scalability beyond a few tens (hundreds?) of thousands of nodes, over a (relatively) small number of SANs.

PktWay does not support SAR (Segmentation And Reassembly). Instead, it provides means for hosts to discover the minimum transmission unit (MTU) over several alternative paths to any other node. A PktWay packet must never exceed the minimum MTU along all the network hops from the source node to the destination node.

Several protocol extensions, which are layered on the core PktWay protocol, have been defined. These include dynamic resource and routing discovery, secure PktWay, and multicast PktWay. These protocol extensions will be described in documents to be provided later.

1c. The Level-2 Operation of PktWay

PktWay's goal is to move data from a source node, (on some arbitrary SAN) to a destination node, (either on the same SAN, or on another SAN). Sources and destinations can be physical entities, such as a processor or a smart memory board, or logical entities, such as a group of cooperating processes or a collection of threads. Sources, destinations, and routers are such nodes.

Within each PktWay configuration all nodes have unique 23-bit physical PktWay addresses. A system designer can assign these PktWay addresses manually. Alternatively, the optional PktWay Server Layer may provide a way to assign and discover addresses dynamically. Throughout this document "address" always means the 23-bit physical PktWay address.

To optimize for performance, PktWay has a data transfer mode that directly leverages the native message routing schemes used within each SAN. This mode uses a "Planned Transfer" paradigm. During the planning phase, a source node collects information on optimal routes to a destination, expressed in the various native formats of all the intervening SANs. A source node later uses this information for low latency transfers to that destination. In PktWay, the transfer phase of a Planned Transfer is called "L2-forwarding". The RRP document demonstrates the use of L2-forwarding.

PktWay also supports a more traditional data transfer mode that requires no planning. Such transfers specify the destinations by their addresses only. In PktWay, this more traditional approach is called "L3-forwarding".

PktWay packets may be routed by Level-2 (L2) forwarding, Level-3 (L3) forwarding, or a combination thereof.

In L3-forwarding (similar to IP forwarding), the L2-routing through each SAN is determined by an inter-SAN router upon entering that SAN. The router prefixes the packet with an L2 routing header (such as a source route) corresponding to the destination address specified in the packet directing the packet either to its destination or to an intermediate router. It is a task for that router to determine the L2-routing-header corresponding to the given PktWay-address.

In L2-forwarding the source prefixes the packet with all the L2-routing headers needed along the entire path to the

destination. Each router has only to get the L2-routing-header from the leading L2RH (L2-Routing-Header record) that was provided by the source.

PktWay allows hosts to construct a source-route built entirely of Level-2 headers, allowing each SAN to exploit the full performance of its native interconnection fabric. These SAN-headers (equivalent to MAC-headers) are provided by the SANs that will use them, in their native format. PktWay does not define the format of the local routing envelope. Instead, it defines how the encapsulated PktWay packets should be passed between half-routers, leaving it up to the local network of each SAN to properly deliver the packet.

If hosts so prefer, they can address their destinations either by any arbitrary name, a PktWay physical address (which is handled like the Level-3 IP-address), or by concatenating a sequence of Level-2 SAN-headers. Although the generation of a sequence of L2 Routing Headers requires more effort to construct initially, PktWay source routing results in considerably lower network latencies, as the packets are allowed to cut-through route through the intervening SAN networks .

2. A note about the PacketWay Documents

The PacketWay protocol is defined by a series of documents:

* EEP (End-to-End Protocol)
* RRP-1 (basic Router-to-Router Protocol)
* RRP-2 (dynamic inter-SAN routing)
* PktWay enumerations

Each of these documents should include the same "PacketWay at a Glance (Cheat-Sheet)", this note, and the Notations page. They should include also (as appendices) a copy of the PacketWay glossary of terms and its acronyms and abbreviations list.

The EEP and the RRP documents will be published first as Internet-Drafts and later as Proposed-Standards, Draft-Standards, and Standards.

The Enumeration Document will be first published as an "Informational-RFC" and later will be maintained by IANA.

The enumeration document may be attached to the EEP/RRP documents, as a matter of convenience. The enumeration is NOT a part of the PktWay standard, just as RFC0739 (the original "Assigned Numbers" RFC) is not a part of RFC0791, that defines IP.

Similarly, the EEP-document has "Appendix-A: A Recommendation for PktWay Address Assignment" which is a recommendation only and NOT a part of the PktWay standard, just as IP-address-assignment is not a part of RFC0791, that defines IP.

The appendices are brought for clearance and convenience. They are not a part of the PktWay specification.

Information about the PktWay activity may be found in the URL: http://www.erc.msstate.edu/PktWay/

3. Notations

The shorter "PktWay" is used for "PacketWay".

8B means "8-byte" (64 bits).

0x indicates hexadecimal values, e.g., 0x0100 is 2^8=256(decimal).

0b indicates binary values, e.g., 0b0100 is 4(decimal).

xxxx indicate a field that is discarded without any checking (e.g., padding).

[fff] indicates that fff is an optional field, when appropriate.

[exp] in equations, is the integral part, rounded down, of `exp`. e.g., [23/8]=2.

All length fields do not include themselves, and therefore may be zero.

Lengths are specified either (a) by byte count, implying that some padding bytes may follow to fill 8B-words, or (b) by 8B-word count and PL, the number of trailing padding bytes (with PL between 0 and 7).

4. PktWay EEP Messages

4a. The Pktway Message Structure

PktWay messages have 6 components, including 4 optional ones:

[1]: [Optional Sequence of L2-Routing-Headers and Symbols]
[2]: EEP Header (16 bytes) (PH)
[3]: [Optional Header fields] (OH)
[4]: [Optional, Most likely: Data Block] (DB)

[5]: [Optional Trailer fields] (OT)
[6]: EEP Trailer (8 bytes) (TAIL)

4b. The Optional Fields

[1]: as explained later, if the 9th+10th bits of a messages are 0b10 then the message starts with an L2RH, but if the 9th through the 12th bits of a message are 0b1111 then this message starts with a "symbol". The other values of these 4 bits indicate the lack of L2RH and symbols and that the message begins with the EEP-header.

[3]: if the h-bit in the EEP header [2] is 1 then there are optional header (OH) fields. The sequence of these OH fields is terminated with an OH field marked as being the last one (with C=1).

[4]: if DL>0, in the EEP header, zero then a Data Block (DB) is included in this message.

[5]: the optional header fields, [3], may indicate that some optional trailer fields are present after the DB, [4]. The order and the formats of the trailer fields are defined by the optional header fields.

It is expected that most messages will have Data Blocks (DB), and that most messages will not have Optional Header fields (OH), nor Optional Trailer fields (OT).

Leading L2RHs and symbols [1] are consumed by the HRs before reaching the destination which receives only the other components, [2] through [6]. These parts, [2] to [6], constitute the End-to-End Protocol of PktWay.

TAIL, the EEP trailer, [6] may be modified along the way to the destination, unlike [2], [3], [4] and [5], which arrive exactly as sent by the source.

Each PktWay packet may be first L2-forwarded (zero or more times) before being L3-forwarded (zero or more times).

Although PktWay headers and trailers are always in Big Endian order, the byte order of the Data Block is not defined by PktWay.

Since all the elements of PktWay (L2RHs, EEP-headers, optional fields, data, and EEP-trailers) are always multiples of 8B-words,

it is recommended that PktWay headers (and data) be aligned on 8B-boundaries in the nodes' memory.

5. Optional Sequence of L2RHs and Symbols [1]

PktWay messages may start with a mix of L2RHs and symbols.

A PktWay source may specify native routes, by placing the native routes before the PktWay Header. The native routes (for all SANs and LANs beyond the initial one) must appear within a sequence of PktWay L2-Routing-Header records (L2RH).

In certain situations symbols may be included among the L2RHs. These symbols are used for conveying information to the routers that handle the messages, such as about encryption. A symbol does not specify its destination and is processed (and consumed) by the entity that encounters it.

In L2-forwarding each intermediate HR consumes an L2RH and the preceeding symbols (if any). When a packet reaches its destination all of [1] (the Optional Sequence of L2RHs and Symbols) should be consumed.

5a. L2 Routing Headers (L2RHs) Records

The contents of the L2RH are totally SAN dependent, with the exception of the first 2 bytes that distinguish this record from an EEP-header and also provide the Length (0<L<64) indicating the number of routing bytes of that L2RH (not including these 2 bytes).

This distinction (between L2RHs and EEP-headers) is necessary for routers that L2-forward packets starting with L2RHs, but L3-forward packets starting with EEP-headers. Similarly, hosts expect packets to start with EEP-headers (with optionally preceeding symbols), and may discard packets that start with L2RHs.

It's up to each SAN to provide padding, as needed, to fill the L2RH words.

Each L2RH is defined by the entity that will process it. In addition to routing information per se, it may also include demuxing information such as a local message-type. For example, over Myrinet the L2RH should end with 0x0300 which is the Myrinet-type assigned to PktWay (and possibly some padding, too).

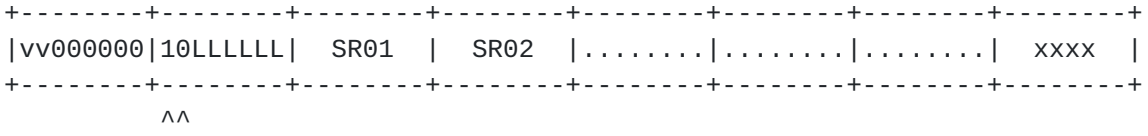
The L2RH must contain enough information to allow a router to create any necessary local routing headers and trailers. Although the low-level network implementation is beyond the scope of this document, the native source routing format must be documented in sufficient detail to allow for heterogeneous network interoperability.

When a PktWay message is encapsulated inside any native SAN message (Paragon or Myrinet, for example), it's up to that SAN to distinguish between it and its own native packets. This is not a PktWay issue. For example, Myrinet uses its Message-Type to recognize PktWay messages.

PktWay-Routers on boundaries between SANs L2-forward packets starting with L2RH or L3-forward packets starting with EEP-headers. L2RH are distinguished from EEP-headers by the value of the first two bits of the Destination-Type field.

5a1. L2RH FORMAT:

Each L2RH is in the format:



The first 2 bits are vv=0b00 for the working version of the protocol. They may have other values for experimental versions.

The next 6 bits should be all zeroes.

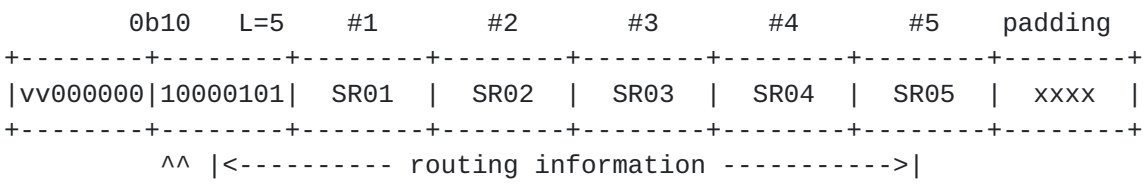
The next two bits must be 0b10 to indicate that this is an L2RH record. This 0b10 was chosen to be consistent with the 0b10 of PktWay-addresses, as described in [2] below.

The next 6 bits are the byte count (L) of the routing information that starts in the next byte and is followed by as many padding bytes as needed to fill to the next 8B-boundary.

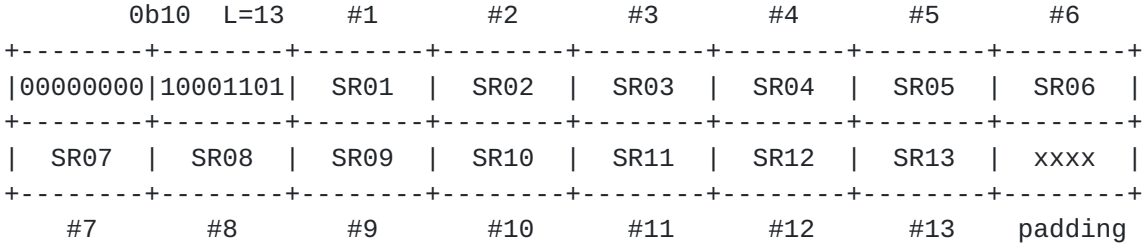
L does not include itself, hence it could be between 0 and 63. However, since this record contains some routing bytes, L is greater than 0. The total number of 8B-words in the L2RH is [(L+9)/8] where the square brackets indicate the integer part, rounded down, of the quantity within. Therefore, the number of padding bytes is PL=8*[(L+9)/8]-2-L.

5a2. L2RH EXAMPLES:

An L2RH with an SR with 5 routing bytes:



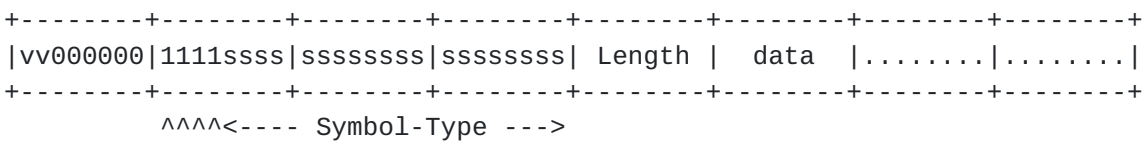
An L2RH with an SR with 13 routing bytes:



5b. Symbol Records

5b1. Symbol Format:

Each symbol is in the format:



The 5th byte is the byte-count (L) of the data for this field that starts in the next byte, and is padded with as many padding bytes as needed to fill 8B-words.

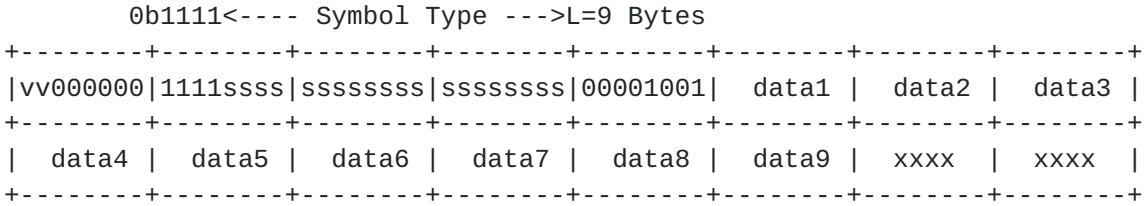
The length (L) does not include itself, hence it is between 0 and 255. The total number of 8B-words in the symbol L2RH is [(L+12)/8] where the square brackets indicate the integer part, rounded down, of the quantity within. Therefore, the number of padding bytes is PL=8*[(L+12)/8]-2-L.

Symbols may be mixed among the L2RHs, before the EEP-header.

The values of the Symbol-Type field are defined in the PktWay Enumeration document.

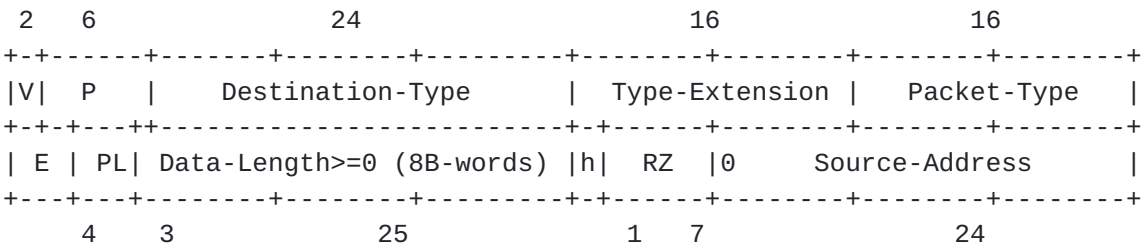
5b2. Symbol Example:

A symbol with 9 data bytes.



6. EEP Header [2]

The EEP (aka PH) has 16 bytes.



These fields are described below:

		Bytes.bits
a. Version	(V)	0.2
b. Priority	(P)	0.6
c. Destination-Type	(DT)	3.0
d. Packet Type Extension	(TE)	2.0
e. Packet Type	(PT)	2.0
f. Endianness	(E)	0.4
g. Padding Length	(PL)	0.3
h. Data Length	(DL)	3.1
i. Options flag	(h)	0.1
j. Reserved	(RZ)	0.7
k. Source Address	(SA)	3.0

6a. Version (V) 2 bits

This field is static. Its 2 bits are 0b00 for the working version of the protocol. These bits should have other values for co-existing experimental versions.

6b. Priority (P) unsigned integer, 6 bits

It is anticipated that some SANs, especially those working in real time, will want to implement priorities. This field supports such usage.

All ones is the highest priority, and all zeroes the lowest. Ideally, packets with higher priority should gain access to contested resources before packets with lower priority. Implementations may ignore the Priority field.

6c. Destination-Type (DT) 24 bits

The purpose of this field is to specify the header type, as well as the destination of the packet, when applicable.

This field may specify:

* A physical PktWay address (of 23 bits);
* An L2-Routing-Header (L2RH) of a variable length;
* A logical address (of 20 bits); or
* A symbol (of 20 bits).

In addition, it is anticipated that additional types will be needed in the future.

A variant of Huffman coding is used to accommodate all these methods for the Destination-Type field. This is done by assigning the MSbit of 0 to physical addresses, 2 MSbits of 0b10 to L2RH, 3 MSbits of 0b110 to future needs, 4 MSbits of 0b1110 to logical addresses, and 4 MSbits of 0b1111 to symbols.

This assignment is summarized in the following table:

MSbits	Method
0xxx	Physical
10xx	L2RH
110x	Reserved
1110	Logical
1111	Symbol

A single C-style 16-way switch can dispatch quickly the protocol processor to the right handler required for any of the methods used to specify the destination.

The Physical addresses are unique within each instance of PktWay. Nodes should have addresses assigned to them. The method of assigning unique addresses within each PktWay is not specified here.

Examples of potentially addressable PktWay nodes include: groups of cooperating processes, an entire MPP, or each of an MPP's many processors or processes.

The 0b10xx was chosen for L2RH to be consistent with the 0b10 indication of L2RHs, as described earlier in this document.

"Logical Addresses" (e.g., for broadcast and for multicast groups)

are also in this address space. The destination-Type is a "Logical Address" if its 4 MSbits are set to 0b1110.

A few Physical-addresses are reserved:

0x000000 Undefined address (illegal where an address is expected, but is allowed in the SA field)

0x7FFFFE ("Hey-You!") This address could be used at power up to address nodes or routers, over point-to-point links. ("If you receive it, it's for you.")

0x7FFFFF (Broadcast) This address is reserved for broadcast operations which may be added in later versions. ("If you receive it, it's for you.")

6d. Type Extension (TE) 2 bytes

An extension of the following PT field.

Logically, the TE should be after the PT. However, the PT is 8B-word aligned, easier to process than the TE which is 2B-aligned, but not 8B-aligned. Since the PT is more frequently used than the TE, it was assigned to the better aligned field.

6e. Packet Type (PT) 2 bytes

The PT field provides the information needed for efficient de-multiplexing of multiple protocol layers. Whereas traditional protocol layering requires several stages of sequential de-multiplexing, PktWay provides enough information to support a single combined de-multiplexing operation (such as in support of zero copy TCP). Thus, the PT field may indicate, for example, that the data blocks contain IP, SNMP, ATM, Ethernet, or other layered protocols.

PT values to support popular parallel programming APIs such as MPI have been defined. The PktWay Enumeration document defines several values for this PT field.

The PT field value of "RRP" indicates that message contains commands used in the PktWay Router-to-Router Protocol (RRP).

Some PTs will also use the 2 byte Type Extension (TE) field which precedes the PT for passing PT-specific parameters, such as implementation specific de-multiplexing information.

RRP messages (as described in the PktWay RRP document) use the TE field to distinguish among the various RRP-messages.

Special Packet Types

RRP - PktWay's Router/Router protocol (see the RRP document).

ERR - Error reporting packet, usually sent to the Source Address (SA, see below) in response to a PktWay message that could not be properly handled, such as "Destination Unknown." The TE indicates the nature of the error (e.g., UNK) as defined in the PktWay Enumeration document.

6f. Endianness (E), 4 bits

If the SAN interface of the receiving-node detects Endianness that is different than its own and if the entire Data Block (DB) consists of N-byte fields, then it may activate byte-swapping hardware for N-byte fields, saving much work for the receiving node.

The first bit (MSbit) of E, 'e' indicates whether the DB is in Big-Endian order (e=0) or in Little-Endian order (e=1). The next 3 bits could control hardware byte swapping, if any, which assumes that all the data consists of words of the same length.

The meaning associated with the values of the 3 LSbits of this field are defined in the PktWay enumeration document.

6g. Pad Length (PL) unsigned integer, 3 bits

The number of padding bytes that were added at the end of the DB (i.e., from the end of the data to the end of the DB). PL can be between 0 and 7.

6h. Data Length (DL) unsigned integer, 25 bits

Length, in 8B-words, of the data block, not including the L2RHs, EEP-header, OH, OT, and TAIL, including any optional padding. Hence, the net length of the Data Block is 8*DL-PL bytes. The minimum is zero, and the maximum length is (2^25-1)*8 bytes = ~2^28 = 256 MBytes.

6i. Optional Header-Field Flag (h) 1 bit

This bit is set to 1 if there are one (or more) optional header (OH) fields following the standard 16-byte EEP-header.

6j. Reserved (RZ) 7 bits

This field is reserved for future use. Applications should neither use it, nor count on others not to use it. It should be always set

to zero (0b0000000).

6k. Source Address (SA) 24 bit

This field contains the physical address of the packet's original source in the same format as the DT. However, unlike the DT, the SA must be a physical address.

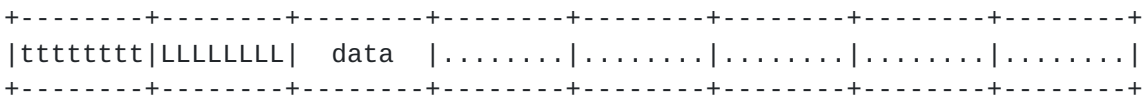
Filling in this field is optional. A value of zero means that the SA is not specified.

Routers may use this field to identify the sender to which error messages may be returned.

7. Optional Header Fields (OH) [3]

A PktWay-message has Optional Header fields (OH) following the EEP-header, if the Option-Flag (h) is set to 1 in the EEP-header.

Each OH is in the format:



The first byte indicates the optional header field type (OH-TYPE).

The first bit, T, of the first byte indicates the processing of this OH-TYPE:

T=0: Optional (may drop this field if this OH-TYPE is unknown)
T=1: Mandatory (should not process this message if this OH-TYPE is unknown)

The second bit, C, of the first byte indicates whether there are more header fields (i.e., whether this is the last field of this message).

C=0: More Optional Header fields follow
C=1: End of Optional Header fields group (i.e., this is the last OH)

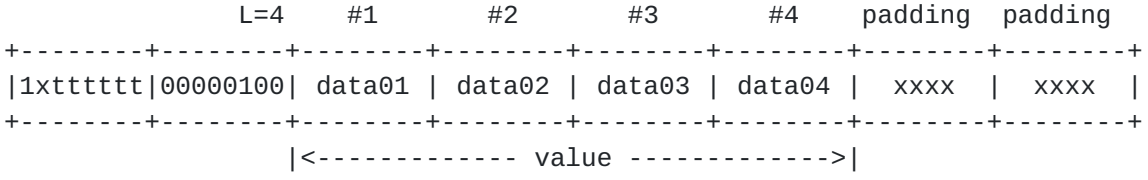
The other 6 bits of this byte, tttttt, define application-specific OH-TYPEs.

The second byte is the byte-count (L) of the data for this field that starts in the next byte, and is padded with as many padding bytes as needed to fill 8B-words.

The length (L) does not include itself, hence it is between 0 and 255. The total number of 8B-words in the symbol L2RH is [(L+9)/8]

where the square brackets indicate the integer part, rounded down, of the quantity within. Therefore, the number of padding bytes is PL=8*[(L+9)/8]-2-L.

Example: An Optional Header Field (OH) with a mandatory OH-TYPE and 4 data bytes:



8. Optional Data Block (DB) [4]

The DB is free for applications to use in any way. Routers must not modify this field.

The DB has DL 8B-words, including optional padding (at the end) of PL bytes. Hence, the number of data bytes is 8*DL-PL. Both DL and PL are specified in the EEP-header.

The maximum length of the DB is 8*(2^25-1)B = ~256 MByte.

9. Optional Trailer Fields (OT) [5]

A PktWay-message has Optional Trailer fields (OT) if so indicated in an Optional Header field, e.g., an OH field may indicate that a CRC64 is in the OT.

An OT may have just the data for an OH defined above (following the EEP header), or be a stand alone, self-defined field in the same format as OH.

The OT-fields are in the order defined by the OHs. For example, if an OH-field indicating that a CRC32 is in the OT, is followed by another OH-fields indicating that a CRC64 is in the OT, then the OT with the CRC32 should be followed by the OT with the CRC64. Self defined OT fields must follow OTs defined by the OHs.

10. EEP Trailer (TAIL) [6]

The TAIL consists of only the Error Indication (EI) field which is a single 8B-word.

Routers may start forwarding packets toward their destinations before detecting transmission errors (such as in wormhole routing). The EI field provides such routers with a means to append an error indication to the end of a packet.

An all zero EI value means that no error was indicated. Any non-zero EI value indicates one or more errors.

The packet source will usually initialize the EI field to all zeros. However, as an alternative example, a memory board may create a packet with a non zero EI field (EI=1) that indicates that a parity error was detected by the memory board.

Each router does an arithmetic left shift, on the EI field by one bit unless its MSbit is 1. Routers that detect transmission errors also set the LSbit (after the shift) to 1.

This provides the ability to identify which routers have indicated errors (if the route is known).

11. Appendix-A: A Recommendation for PktWay Address Assignment

This section of the EEP document is a recommendation only, and not a part of the PktWay standard.

Unlike IP addresses, physical PktWay addresses are not globally unique, but must be locally unique within each PktWay configuration. Hence, when SANs that were developed independently are interconnected to form a PktWay, conflicting physical addresses may occur.

It is recommended not to attempt to assure local uniqueness of physical addresses by subdividing the global address space (hence, attempting to achieve global uniqueness).

Instead, it is recommended that every SAN would have local PktWay addresses, between 1 and the number of its local nodes, and also have a global "bias" to be added to all the addresses in that SAN. Hence, by proper setting of the biases of interconnected SANs, the local uniqueness of PktWay addresses is achieved.

The coordination of these biases is left (at least now) for manual (static) out-of-band coordination.

The use of such biases simplifies the mapping of physical addresses to their SANs.

12. Appendix-B: Glossary

Address: A unique designation of a node (actually an interface to that node) or a SAN.
Buddy-HR: HRs are "buddies" if they are on the same SAN.
Cut-Thru: See wormhole.
Destination: The node to which a packet is intended
Dynamic-Routing: Routing according to dynamic information (i.e., acquired at run time, rather than pre-set).
Endianness: The property of being Big-Endian or Little-Endian (transmission order, etc.)
Ethertype: A 16-bit value designating the type of Level-3 packets carried by a Level-2 communication system.
HR: Half-Router, the part of a router that handles one network only.
L2-Forwarding: Forwarding based on Level-2 (i.e., data-link layer of the ISORM) information, e.g., the native technique of each SAN or LAN. Also called "source routing."
L3-Forwarding: Forwarding based on end-to-end (Level-3 i.e., network layer of the ISORM) addresses. Also called "destination routing."
Map: The topology of a network.
Mapper: A node on a SAN/LAN that has the map and an RT for that network. It is expected that the mapper dynamically updates the map and the RT.
Multi-homed Node: A node with more than one network interface, where each interface has another address.
Node: Whatever can send and receive packets (e.g., a computer, an MPP, a software process, etc.)
Node structure: A C-struct (or equivalent) containing values for some attributes of a node.
Planned Transfer: Transfer of information, occurs after an initial phase in which the sender decides which Level-2 route to use for that transfer.
RCVF: The "Received From" set includes all the physical addresses through which an RT was disseminated, starting with the address of the mapper that created that RT.
Re-direct-message: A message that tells nodes which HR should be used in order to get to a certain remote address.
Router: The inter-SAN communication device
Security Context: A relationship between 2 (or more) nodes that defines how the nodes utilize security services to communicate securely.
Source: The node that created a packet.
Source-Route: A Level-2 route that is chosen for a packet by its source.
Symbol: Data preceeding the EEP header of a PktWay message,

interleaving with the L2RHs.

Twin-HR: Two HRs are twins if they both are parts of the same inter-SAN router.
Wormhole-routing: (aka cut-thru routing) forwarding packets out of switches as soon as possible, without storing that entire packet in the switch (unlike Stop-and-forward)
Zero-copy TCP: A TCP system that copies data directly between the user area and the network device, bypassing OS copies

13. Appendix-C: Acronyms and Abbreviations

0bNNNN	The binary number NNNN (e.g., 0b0100 is 4-decimal)
0xNNNN	The hexadecimal number NNNN (e.g., 0x0100 is 256-decimal)
8B	8 byte (64 bits) entity
ADDR	The Address-record of RRP
APIn	Application/Program Interface
AT	Address Type
ATM	Asynchronous Transmission Mode
B	Byte (e.g., 4B)
b	bit (e.g., 32b)
BC	Byte Count (of parameters)
BER	Bit Error Rate
CAPA	The CAPAbility-record of RRP
CC	Capability Code
CSR	Common Source-Route
DA	Destination Address
DB	Data Block
DL	Data Length (in 8B words)
DSP	Digital Signal Processor
DT	Destination-Type
e	The MSbit of E
E	The Endianness field (in the EEP header)
EEP	End/End Protocol
EI	Error Indication
GP	General Purpose
GVL2	An RRP message, requesting L2 route to a given destination
GVRT	An RRP message asking an HR to give its routing tables
h	Optional header fields flag
HR	Half Router
HRTO	An RRP message asking which HR to use for a given destination
ID	Identification
IGMP	Internet Group Management Protocol
INFO	An RRP message providing information about nodes
IP	The Internet protocol
ISORM	The ISO Reference Model
L	Length field (exclusive of itself)
L2	Level-2 of the ISORM (Link)
L2RH	Level-2 Routing Header
L2SR	Source Route

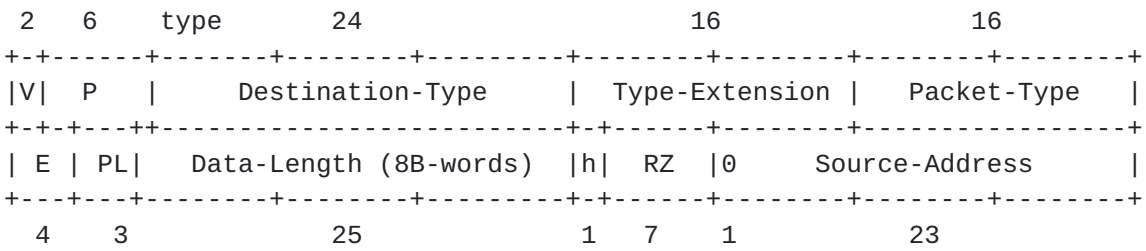
L3	Level-3 of the ISORM (Network)
LA	Logical Address
LADR	The Logical-addresses-record of RRP

LAN	Local Area Network
LRT	Local Routing Table
LSbit	Least Significant bit
LSbyte	Least Significant byte
MAC	Message Authentication Code / Media Access Control
MPI	Message Passing Interface
MPP	Massively Parallel Processing system
MSbit	Most Significant bit
MSbyte	Most Significant byte
MSU	Mississippi State University
MTU	Maximum Transmission Unit
MTUR	The MTU-record of RRP
M/C	Multicast
NAME	The name-record of RRP
NFS	Network File Server
OH	Optional Header field
OH-TYPE	The Type of an Optional Header field
OT	Optional Trailer field
P	The Priority field
PAD	Padding After Data
PBD	Padding Before Data
PCI	The Peripheral Component Interconnect "standard"
PH	PacketWay Header
PL	Padding Length (always in bytes)
PPP	The Point-to-Point Protocol
PROM	Programmable ROM (Read-Only-Memory)
PT	Packet Type (2B)
PVM	Parallel Virtual Machine
PW	The Myrinet Packet Type assigned to PktWay (PW=0x0300)
Q	Quality (of a path)
RCVF	Received-From list, or the Received-From record of RRP
RDRC	A re-direct message of RRP
RH	Routing Header
RID	Record ID
RL	Record Length (in 8B-words)
RRP	Router/Router Protocol
RT-hd	RT (Routing Table) header
RT	Routing Table
RTBL	An RRP message proving a Routing Table
RTHD	The Routing-Table-Header record of RRP
RTyp	RRP's Record Type
RZ	The Reserved field (in the EEP header)
SA	Source Address
SAN	System Area Network
SAN-ID	The 24-bit PktWay-address of a SAN
SAR	Segmentation and Reassembly
SN	Serial Number
SNID	SAN-ID
SNMP	Simple Network Management Protocol

SR	Source Route (always at Level-2)
SRQR	The Source-Route-and-Q-record of RRP
ST	Symbol Type

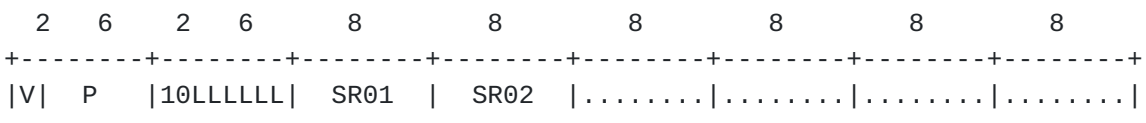
TAIL	PacketWay EEP Trailer
TE	Type Extension (2B)
TELL	An RRP message requesting information about nodes partially specified
UNK	Unknown
V	Version
WRU?	An RRP message asking its recipient to identify itself
XRT	External Routing Table
xxxx	A padding byte

14. Appendix-4: PktWay at a Glance (aka "The Cheat-Sheet")



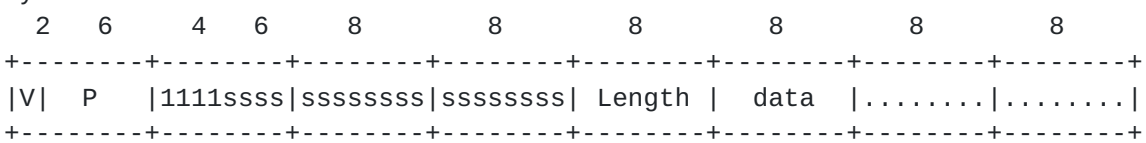
type = 0xxx Physical Address
10xx L2RH
110x Reserved
1110 Logical Address
1111 Symbols

L2RH:



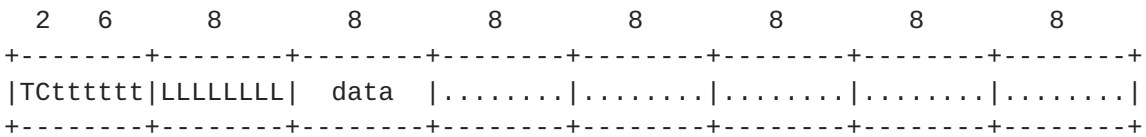
Length

Symbol:



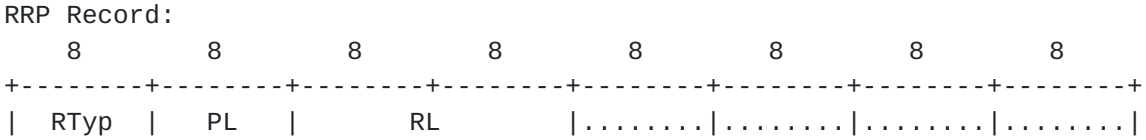
<---- Symbol Type --->

Optional Header:



T: 0=optional, 1=mandatory; C: 0=more OH-fields follow, 1=last OH-field

RRP Record:



```
+--------+--------+--------+--------+--------+--------+--------+--------+
RRP-messages: GVL2, L2SR, RDRC, TELL, INFO, HRTO, WRU,  GVRT, RTBL;
        RTyp: ADDR, NAME, CAPA, LADR, SRQR, MTUR, RCVF, RTHD;
```

15. Security Considerations

This RFC raises no security issues.

16. Editor' Address

Danny Cohen
Myricom, Inc.
325 N. Santa Anita Ave
Arcadia, CA 91006

Phone: 626-821-5555
Fax: 626-821-5316
Email: Cohen@myri.com