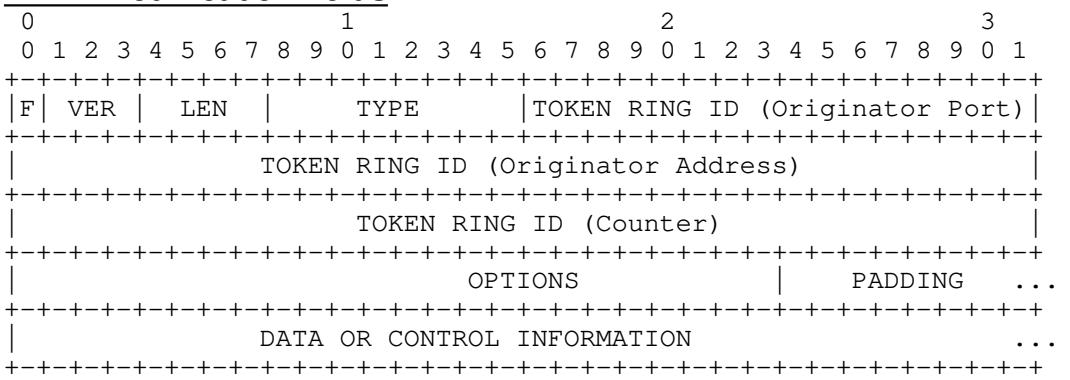
Document Version 1.1, 11/11/94

RMP Fixed Header Fields



F (FWD): Should this packet be forwarded to the IP Multicast address for this token ring? This allows processes that don't have IP Multicast support to use a member of the ring that does support IP Multicast to send multicasts on their behalf. When a packet with this flag is received, a copy of it is sent to the IP Multicast address for the token ring. The process that receives it may or may not choose to handle this copy immediately.

VER: This specifies the version of the protocol. The version specified by this RFC is 1. Packets received with a differing version number should be rejected.

LEN: The length of the options field, in four byte words. This allows variable length options to easily be included without penalizing the common case. A quick check to this FWD/VER/LEN byte can detect the common case of a non-forwarding packet of the correct version with no options. With this version of the protocol, the value of this byte for this common case is equal to 0x10.

TYPE: This specifies the type of the packet. The valid types are listed below, along with whether they are sent to all members (multicast) or to a single member (unicast).

Number	Packet Type	Multicast?
0	Reserved	
1	Data Packet	Yes *
	Control Packet	
2	ACK	Yes
3	Confirm Token Pass	No
4	NACK	Yes
5	New List	Yes
6	List Change Request	Yes
	Failure Recovery Packet	
7	Recovery Start	Yes
8	Recovery Vote	No
9	Recovery ACK New List	Yes
10	Recovery Abort	Yes

	Non Member Packet	
11	Non Member Data	No
12	Non Member ACK	No
13-15	Reserved	

* Packets requested by a NACK may be unicast

TOKEN RING ID: This uniquely identifies a token ring across the entire Internet. This number is formed from a combination of the identifier for the RMP process which created the current version of the token list with a unique counter for that host. The identifier for the process is the combination of its IP address and the port number it is using for receiving UDP/IP packets. This counter is initialized when a process starts up with the current time of that host and is incremented each time the process creates a TOKEN RING ID. The time is specified in milliseconds since the beginning of the year 1970, truncated to 32 bits. This is very similar to the way that some RPC systems generate unique ID's, except that they are generated on a pre-process basis instead of on a per-host basis. Under RMP, they are gauranteed unique as long as the clock on the host is stable and as long as no more than one ID is created by a process for each millisecond that the process is running.

OPTIONS: These variable length options are similar to those implemented by IP and TCP. These allow for future needs without penalizing current implementations.

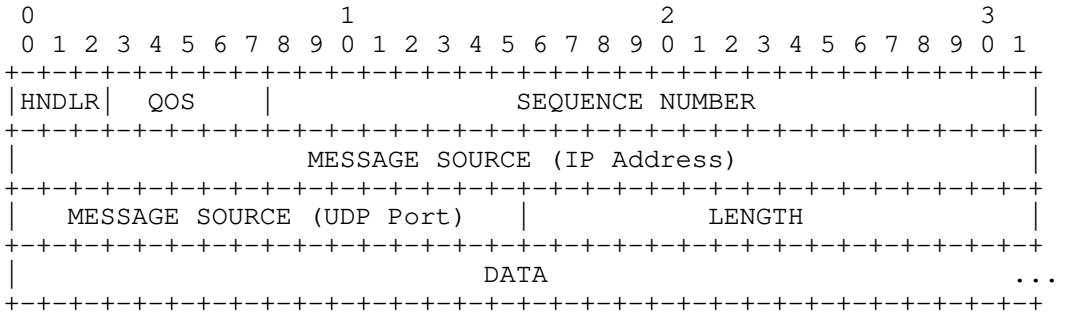
Defined Options

TODO:
Debugging Information
Error codes(?)
32 bit checksum (pull this from VMTP?)

Data Packet (Type 1)

Purpose: Data packets carry data from a member of the ring. A sender multicasts data packets out to the group, which are then acknowledged by the current token site with an ACK packet. If no ACK is received by the sender before the time-out period, the sender periodically resends the packet until it receives an ACK. This data packet may be retransmitted by a site in response to a NACK request from a destination.

Extra Fields:



HNDLR (Handler): This selects the handler, if any, for a data packet. There are six mutually exclusive locks included with each token ring. Based on the value of the HNDLR field, the process that holds one of these locks may be responsible for providing a response to it. Requesting that a packet be handled guarantees that at most one process will handle it. The valid values for HNDLR are:

Value	Required Handler
0	None
1-6	The process, if any, that holds the handler lock with the same number
7	The process, if any, which holds the highest priority handler lock, where 1 is the highest priority and 6 is the lowest.

QOS: This specifies the quality of service for this packet. The valid types are given below, along with a summary of when packets with that type will be delivered to clients at the destination. Types of service marked as optional do not have to be implemented. If a packet is received with an unsupported QOS, it should be delivered according to the next highest QOS. Except for the QOS levels of MAJORITY RESILIENT and TOTALLY RESILIENT, the defined QOS levels are strictly hierarchical, with each level including all guarantees of smaller QOS levels. The maximum K RESILIENT level for any given ring is equal to the number of members of the ring. If a packet is specified with a higher QOS than this, it will be delivered as if its QOS was equal to TOTALLY RESILIENT.

QOS	Name and Service Guarantees	Delivery Time
0	Reserved	Undefined
1	UNRELIABLE--Unreliable, completely unordered delivery	Immediately upon receipt of a data packet
2	UNORDERED--Reliable, completely unordered delivery	Immediately upon receipt of a data packet, with dropped packets being requested and delivered later
3	SOURCE ORDERED--Reliable, ordered with respect to each source	After all of the data packets from the same source and with smaller sequence numbers have been delivered
4	CAUSALLY ORDERED--Reliable, causally ordered **(Optional)**	After all of the causally preceding data packets have been delivered
5	TOTALLY ORDERED--Reliable, totally ordered	After all of the data packets with smaller virtual timestamps have been delivered
6-29	K RESILIENT--Reliable, totally ordered, and K-resilient atomic	After all of the data packets with smaller timestamps have been delivered and the token has been transferred (QOS-5) times
30	MAJORITY RESILIENT--Reliable, totally ordered, majority-resilient atomic	After the token has been transferred N/2 times (so that more than half the sites would have to fail to violate atomicity)
31	TOTALLY RESILIENT--Reliable, totally ordered, completely atomic	After the token has been transferred N-1 times

SEQUENCE NUMBER: Each process uses the sequence number field to sequentially order each Data Packet or List Change Request it creates. Each source keeps track of one sequence number for each token ring it is part of. This field is reset to 0 every time a member joins a ring and is incremented by one every time that source creates a Data Packet. Sequence numbers wrap around after 2^24. If for two sequence numbers A and B, A < B and (B-A) > 2^23, then A is considered to be larger than B. Because a new Token Ring ID is generated with each change to the membership list, delayed packets from members that leave and join again will correctly be identified as such. RMP requires that no more than 2^23 packets be created over a period equal to the maximum lifetime for a datagram packet in the network. This condition holds true when IP is used as the datagram service.

MESSAGE SOURCE: Each RMP process is uniquely identified by a six byte identifier, consisting of a four byte host identifier concatenated with a two byte process identifier. RMP uses the IP address to uniquely identify hosts, and the UDP/IP port which each RMP process must open to identify the process. Kernel implementations of RMP may choose different values for the process identifier, if desired. A data packet is uniquely identified over its maximum possible lifetime by its {SEQUENCE NUM, MESSAGE SOURCE} pair. These fields, along with its HNDLR and QOS fields, are copied into the ACK that timestamps the data packet.

LENGTH: The size of the data field, in bytes. The maximum value under this version is 8172 bytes.

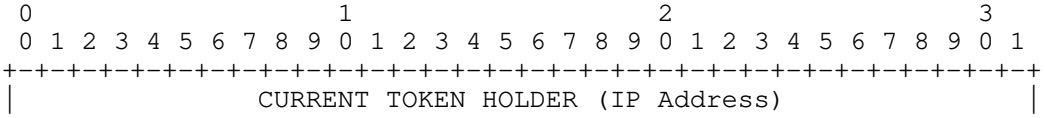
DATA: The data to be delivered.

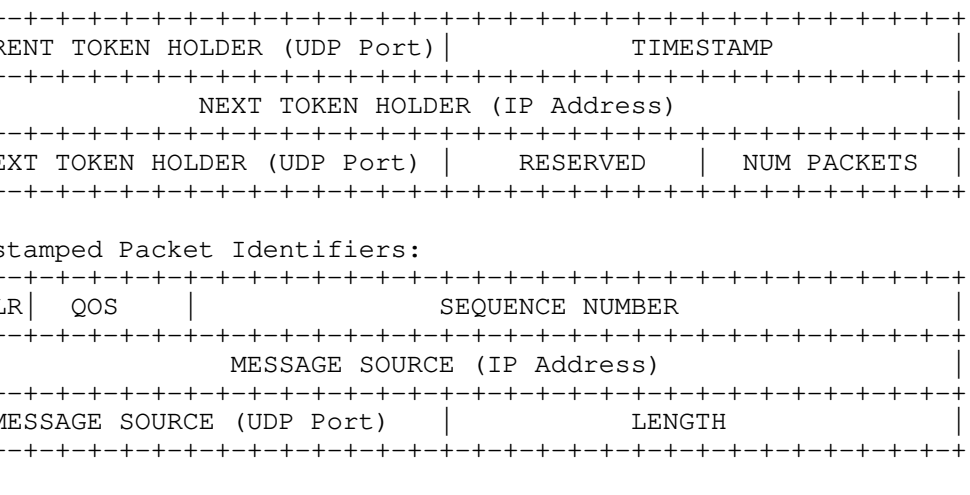
Control Packet -- ACK (Type 2)

Purpose: Each ACK serves two purposes. It passes the token to the next token site in the current token list, and it assigns timestamps to itself and zero or more Data Packets and/or Non Member Data Packets. New List packets (described below) also include their own timestamp.

These timestamps provide three functions. First, they provide a total ordering on packets of these types sent to a token ring. Missing packets in this total ordering are detected and requested with NACKs. Second, this total ordering makes token list changes be consistently ordered between members and allows data packets to be totally ordered. When requested (by packets with a QOS of at least SOURCE ORDERED), this total ordering provides "virtually synchronous"[BIRMAN REFERENCE] packet delivery. Finally, the receipt of an ACK with a timestamp for a Data Packet provides positive acknowledgment to the sender of it that the Data Packet has been received by the current token site.

Extra Fields:





CURRENT TOKEN HOLDER: The IP address and UDP/IP port number of the process which is the current token holder and which issued the ACK. In the case that this packet is stored and retransmitted, neither this field nor any of the others below are changed on retransmission.

TIMESTAMP: The timestamp for this ACK packet. Any other packets that are timestamped by this ACK are assigned consecutive timestamps following this one. Timestamps wrap around after 2^16, and must use wrap-around arithmetic for comparisons.

NEXT TOKEN HOLDER: The IP address and port number of the process that will be the new token site upon reception and handling of this packet.

RESERVED: Must be set to zero.

NUM PACKETS: The number of Data Packets and Non Member Data Packets timestamped by this packet.

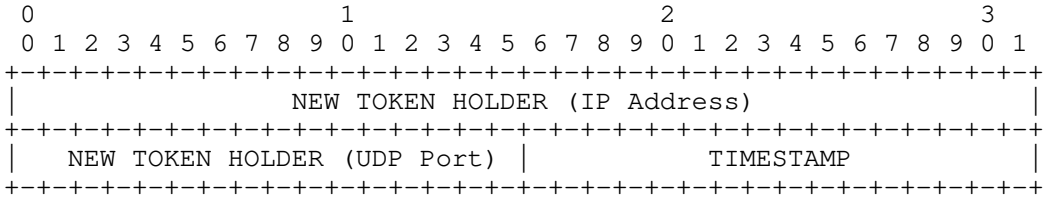
Timestamped Packet Identifiers: There are NUM PACKETS sets of these 12 byte identifiers. Each identifier is copied from the header of the Data Packet or Non Member Data Packet that it timestamps.

Control -- Confirm Token Pass (Type 3)

Purpose: A Confirm Token Pass packet is used to let the old token site know that the token has been accepted by a new token site. This is necessary because token passing requires positive acknowledgment, and the old token site will continually send out its last ACK until it receives this confirmation. Under heavy load, this message is not needed because this function is also served when the new token site sends out an ACK. It is sent out if a retransmission of an ACK naming it as the new token site is received or the site has no data packets to be acknowledged after a time-out period following the receipt of the

token. Unlike most packets, Confirm Token Pass packets are unicast to their destination instead of multicast.

Extra Fields:



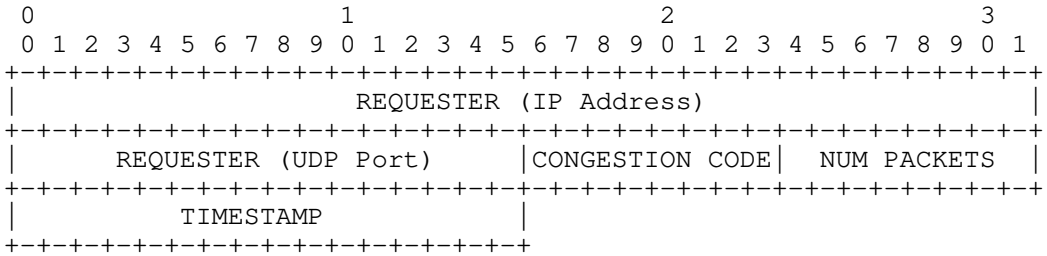
NEW TOKEN HOLDER: The identifier of the process which has accepted the token and is sending this packet.

TIMESTAMP: The timestamp of the ACK packet which passed the token to this site. In effect, this is an acknowledgment of that ACK packet.

Control -- NACK (Type 4)

Purpose: A Negative Acknowledgment (NACK) is used to request the retransmission of one or more missing ACKs, New Lists, and/or Data Packets. These are sent when a gap in the timestamps of received packets is detected.

Extra Fields:



REQUESTER: The IP address and port or the process which created this packet and is requesting retransmission of one or more packets.

CONGESTION CODE: This code provides extra information about the reason for this NACK. This is important, because flow control problems due to buffer overrun are handled differently than congestion in the network. The defined values are:

Number	Meaning
0	Reserved
1	Buffer overrun
2	Probable network congestion
3-255	Reserved

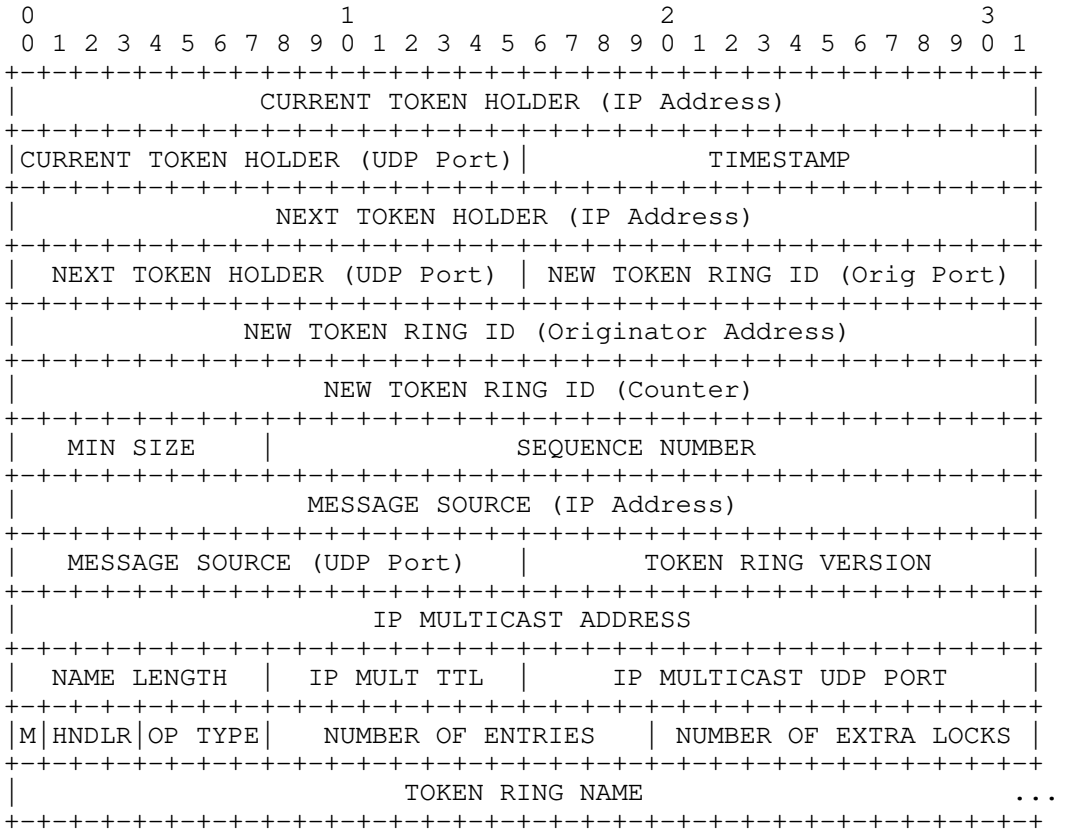
NUM PACKETS: The number of packets requested.

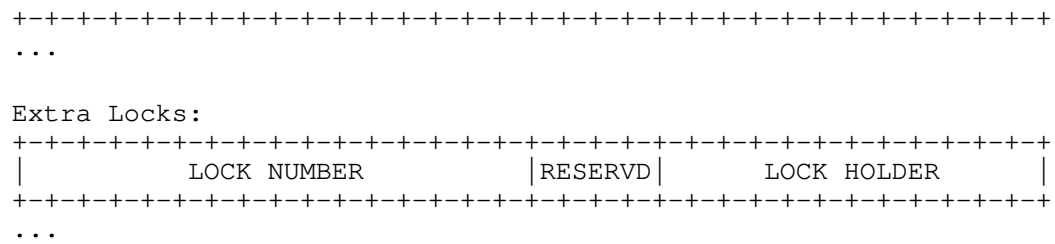
TIMESTAMP: The timestamp for the first missing packet.

Control-- New List (Type 5)

Purpose: A New List packet is the result of a List Change Request or a reformation. It can only be issued by the token site, as an alternative to an ACK. Like an ACK, it passes the token and timestamps itself, although this token pass is not included in the resiliency calculations which determine when a K resilient packet can be delivered. In addition, it contains a new token list, which when committed will result in a new set of members in the ring and/or changed ownership of the handler locks . The new token list is "delivered" or committed when all of the preceding ACKs and New List packets in the OrderingQ have been received and when all of the preceding Data Packets and Non Member Data Packets with QOS <= SOURCE ORDERED have been delivered. Because the token site which issues this New List must already have satisfied these conditions, membership change operations are atomic and totally ordered, and the handler locks granted by New List packets are mutually exclusive.

Extra Fields:





CURRENT TOKEN HOLDER: The IP address and UDP/IP port number of the process which is the current token holder and which issued the New List. In the case that this packet is stored and retransmitted, neither this field nor any of the others below are changed on retransmission.

TIMESTAMP: The timestamp for this New List packet.

NEXT TOKEN HOLDER: The IP address and UDP/IP port number of the process that will be the new token site upon reception and handling of this packet.

NEW TOKEN RING ID: The token ring ID to be used for all packets in this token ring following this one. Packets with token ring ID's for previous versions of the membership list must be received and handled for a short period of time after a new list has been committed, so while there are any packets in the OrderingQ with a given Token Ring ID, packets with that identifier must be handled.

MESSAGE SOURCE: The identifier for the process that issued the List Change Request this New List is a response to.

SEQUENCE NUMBER: The sequence number of the List Change Request packet that this New List is a response to.

MIN SIZE: The minimum number of sites allowed to be in a ring after a site failure. This provides varying levels of consistency in the face of site partitions. The minimum size for the token ring is the largest of the MIN SIZE REQ fields for each token ring member. As an example of its use, an application that needs strong consistency of distributed data will usually require that a token ring must, in order to proceed after a site failure or partition, contain the majority of members that were in it previous to the failure. On the other hand, a groupware or shared tool application will usually allow these minority partitions to proceed. In either case, once a partitioning of a group occurs, the failure reformation algorithm will never rejoin the partitions. The valid values for MIN SIZE are:

Value	Minimum Partition Size
0	Reserved
1-253	Equal to value
254	The majority of sites. Exactly half is not sufficient.
255	All sites. Disables fault tolerance.

TOKEN LIST VERSION: The version of the new token list. This is used to make sure failure reformation always proceeds forward. This is incremented once for every New List packet that is the result of a List Change Request. It is incremented at least once (possibly more) for every New List packet that is the result of a failure reformation.

IP MULTICAST ADDRESS: The IP Multicast address used by this token ring. This is used to notify members who join through a well known IP address and do not know the IP Multicast address and UDP/IP Multicast port for the token ring.

NAME LENGTH: The length of the TOKEN RING NAME field. This includes the trailing zero after the name. This must be an integer multiple of 4, and may be equal to zero.

IP MULT TTL: The IP Multicast TTL field for this token ring.

IP MULTICAST UDP PORT: The IP Multicast UDP port used by this token ring.

M (Multicast Capable): Is the RMP process specified by MESSAGE SOURCE multicast capable?

HNDLR: Optional, used only with Handler Lock operations (types 3-6). It specifies the handler number that the operation was to be performed on.

OP TYPE: The type of operation that was performed on the list. This field details why this new list was generated, and may also report errors. The valid types are:

Value	Operation Type
0	Reserved
1	Requesting Member Added to List
2	Requesting Member Removed from List
3	Member Received a Handler Lock
4	Member Released a Handler Lock
5	Member was Denied a Handler Lock
6	Member Attempted to Release a Handler Lock it Did Not Hold
7	Successful Reformation Occurred Due to a Failure
8	Reformation Occurred With Possible Atomicity Violations
9	Failed Reformation Because Partition is Not Large Enough
10	TTL of Token List ID Expired
11-15	Reserved

NUMBER OF ENTRIES: The number of Token List Entries.

NUMBER OF EXTRA LOCKS: The number of Extra Lock tuples, found at the end of the packet, after each of the token list entries.

TOKEN RING NAME: The null-terminated ASCII name for this token ring. This name must end on a word boundry, which may necessitate that 1-3 extra bytes of padding be included after the trailing zero of the name. These bytes can be set to any value.

Token List Entries: Each entry contains the information for one RMP process. These processes can be either members or non members. For non members, the M, HANDLERS, and MIN SIZE REQ fields will always be zero.

M (Multicast Capable): Is this member of the token ring a member of the associated IP Multicast address?

T (Token Ring Member): Is this process a member of the token ring? Non member sources get periodically flushed from the token ring.

HANDLERS: There are six bits in this field, each of which is mutually exclusive across token ring members. RMP guarantees that at most one source in each token ring has any given handler bit set. These bits can be used to provide fast and convenient locks between members of a token ring, and are also used to designate handlers for different classes of messages. The HNDLR field of Data Packets and Non Member Data Packets optionally selects at most one member of the token ring to handle the response to that data packet.

SEQUENCE NUMBER: The next expected sequence number of any data packet from MEMBER ID with QOS >= RELIABLE. The packet with SEQUENCE NUMBER-1 from that source must have been received before this list can be committed. This is required to provide virtual synchrony in the face of new members joining the ring.

MEMBER ID: The IP address and UDP/IP port for this RMP process.

RESERVED: Must be set to zero.

MIN SIZE REQ: This member's vote for the minimum size partition that is allowed to form in the face of failures.

Extra Locks: Each token list can contain an extra set of mutually exclusive locks. Up to 4095 total locks can be defined per token ring. The first 6 are handler locks, and should be used by applications that are going to handle requests from clients. Locks 7-4095 are extra locks, to be used for any purpose. These are included because each extra lock operation costs only a single round-trip time, on average.

LOCK NUMBER: The number of the lock, between 7 and 4095.

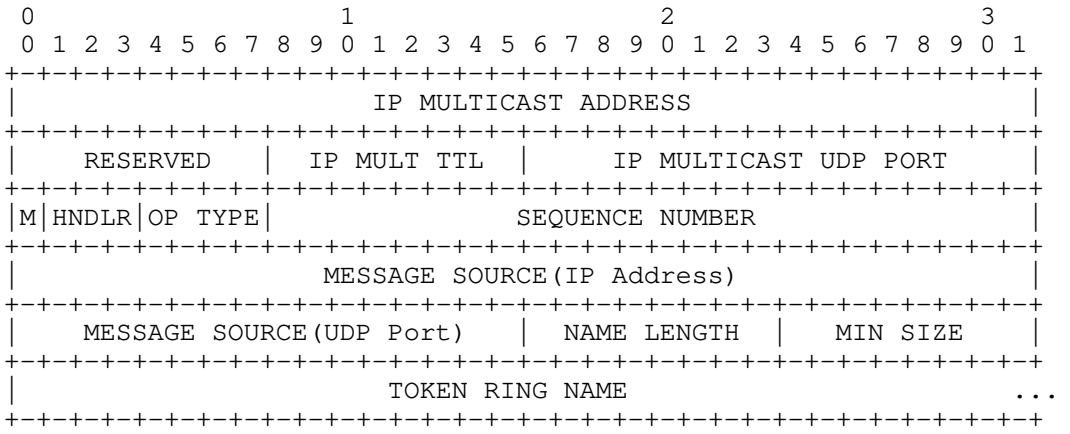
RESERVD: Must be set to zero.

LOCK HOLDER: The current holder in the lock. This is an index into the token list included in this packet. The first member in the list is denoted 1. This field should not ever be equal to zero, but if so, this indicates that the lock is not held by any site.

Control -- List Change Request (Type 6)

Purpose: A List Change Request packet is used by a process to request a change in the token list. The request can ask that a member be added, that a member be removed, that a lock be granted to a member, or that a lock be released by a member. This request is multicast repeatedly to the token ring until either the current token site responds with a New List packet granting the request, or (only for locks) a New List packet that invalidates the request.

Extra Fields:



IP MULTICAST ADDRESS: Optional, used only by Add Member To List operations. The IP Multicast address used by this token ring.

IP MULT TTL: The IP Multicast TTL field for this token ring. Also optional, used only by Add Member to List operations.

IP MULTICAST UDP PORT: Optional, used only by Add Member To List operations. The UDP port used for this IP Multicast address.

M (Multicast Capable): Is the RMP process specified by MESSAGE SOURCE multicast capable?

HNDLR: Optional, used only with Handler Lock operations (types 3 and 4). It specifies the handler number that the operation is to be performed on.

OP TYPE (Operation Type): The following types of requests are defined.

Value	Operation Type
0	Reserved
1	Add Member to List

2	Remove Member from List
3	Request Handler Lock
4	Release Handler Lock
5	TTL of Token List ID Expired -- Null Op
6-15	Reserved

MESSAGE SOURCE: The identifier for the process issuing this List Change Request

SEQUENCE NUMBER: Each process uses the sequence number field to sequentially order each Data Packet and List Change Request it creates. Each source keeps track of one sequence number for each token ring it is part of. This field is reset to 0 every time a member joins a ring and is incremented by one every time that source creates a Data Packet.

NAME LENGTH: The length of the TOKEN RING NAME field.

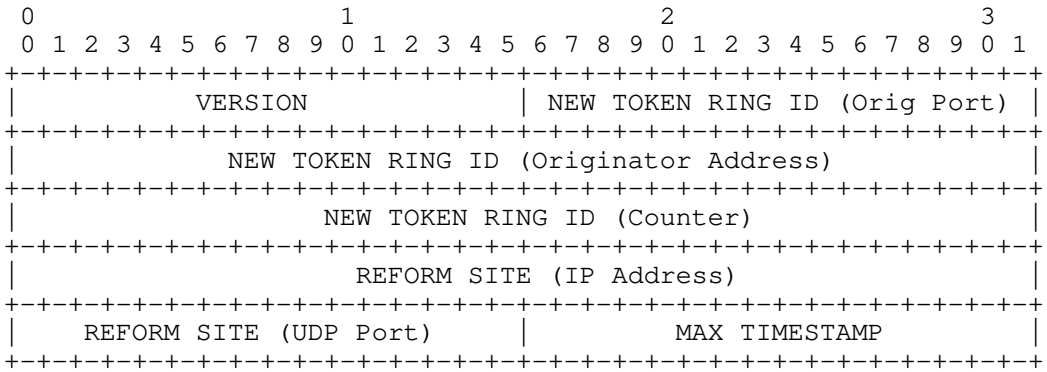
MIN SIZE: The minimum number of sites allowed to be in a ring after a site failure.

TOKEN RING NAME: Optional, used only with Add Member to List operations. The null-terminated ASCII name for this token ring. This is used when the member who wants to join does not know the Token Ring ID for the token ring. When this field is used, the Token Ring ID should be set to 0.

Recovery -- Recovery Start (Type 7)

Purpose: This packet is sent when a site detects a failure. It starts the reformation algorithm and notifies the other sites in the ring of the maximum timestamp it thinks they should have. It also includes the version of the new token list, which is at least one larger than that for any known previous list (even if the list was not committed). When a site receives this packet, they have not yet joined a new token list, and the version of this token list is greater than any previous one they had joined, they join this one and send back a Recovery Vote packet.

Extra Fields:



VERSION: The version of the token list. This is monotonically increasing, and it wraps around after 2^16.

NEW TOKEN RING ID: The ID for this new token ring if the reformation algorithm completes. This is generated by the Reform Site at the beginning of the reformation algorithm.

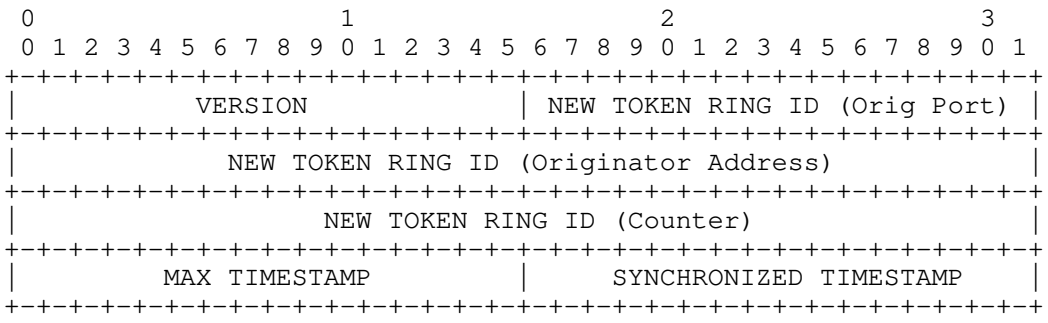
REFORM SITE: The IP address and UDP/IP port for the site that started the reformation algorithm. This is the site that is sending the Reform Start packet.

MAX TIMESTAMP: The maximum timestamp of the old list. All sites in the new list must have all packets (except for unreliable data packets) up to this timestamp before they are allowed to join the new list. If another member of the list has a higher timestamp than this value, it will send this timestamp back in its vote, and the recovery site will send a new Recovery Start packet with this value.

Recovery -- Recovery Vote (Type 8)

Purpose: When a site is in a normal state and it receives a valid Recovery Start packet, it joins (but does not commit) the token list denoted by the New Token ID of that packet. It then unicasts a Recovery Vote packet to the source. It does this repeatedly (after an alarm for the packet expires or another Recover Start from the same source is received) until it receives a New List packet from the recovery site for this new version or it decides to abort the reformation. The Recovery Vote packet notifies the reformation source of the highest timestamp it has for the old list, and indicates whether or not it is missing any packets (besides unreliable ones) before this timestamp.

Extra Fields:



VERSION: The highest known version of the old token list. The next reformation will have to start with a value at least one higher than VERSION.

NEW TOKEN RING ID: The ID for this new token ring if the reformation algorithm completes.

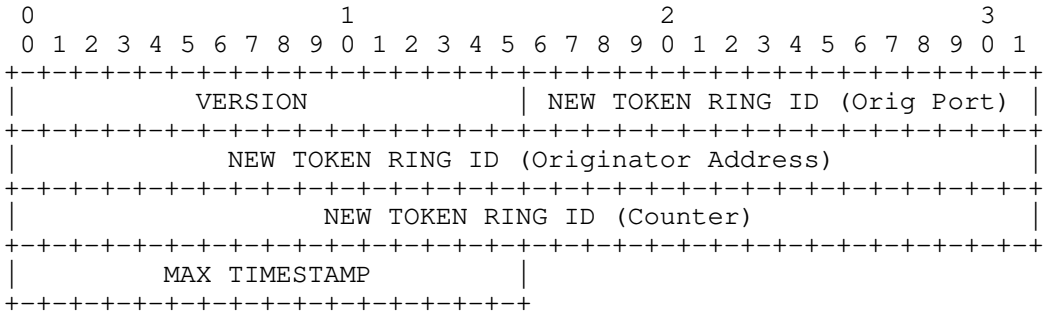
MAX TIMESTAMP: The maximum known timestamp of the old list. If this site has a higher timestamp for the old list than that of the Recovery Start packet(s) it received, it will include this value in the Reform Vote.

SYNCHRONIZED TIMESTAMP: The highest timestamp this site is synchronized up to. This site has all packets, except for unreliable data packets, with timestamps less than or equal to this value. A new list must have all sites synchronized to the same timestamp before a new list for it can be generated.

Recovery -- Recovery ACK New List (Type 9)

Purpose: This acknowledges that the sending RMP process has received the New List packet for the new list it joined, is still a member of this new list, and is synchronized up to MAX TIMESTAMP. Once the reform site receives an ACK New List from all of the other members of the new list, it commits the list locally, becomes the token site, and starts acknowledging packets. Upon receipt of an ACK or a non-reformation New List with the Token Ring ID for this new list, this site also commits the new list and starts processing packets as normal.

Extra Fields:



VERSION: The highest known version of the old token list. The next reformation will have to start with a value at least one higher than VERSION.

NEW TOKEN RING ID: The ID for this new token ring if the reformation algorithm completes.

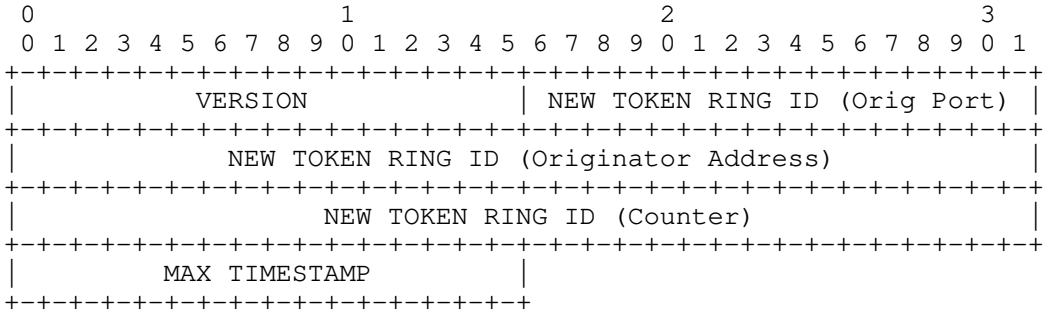
MAX TIMESTAMP: The maximum known timestamp of the old list. This must be the equal to one less than the timestamp of the token list contained in the New List packet this is a response to.

Recovery -- Recovery Abort (Type 10)

Purpose: This notifies members of a token ring that an error has occurred in the current reformation. This can occur due to the detection of two simultaneous reformations (with different Reform Sites) for the same token list, of the detection of a reformation that is proceeding with an out of date version number, or due to repeated timeouts of the same type by one site (indicating another site failure or partition). When received, all sites that

receive it will abort the recovery they were in and set a random timeout. If the site does not receive a new Recovery Start packet from a higher version list before this timeout, it will become the recovery site and start the process again itself.

Extra Fields:



VERSION: The highest known version of the old token list. The next reformation will have to start with a value at least one higher than VERSION.

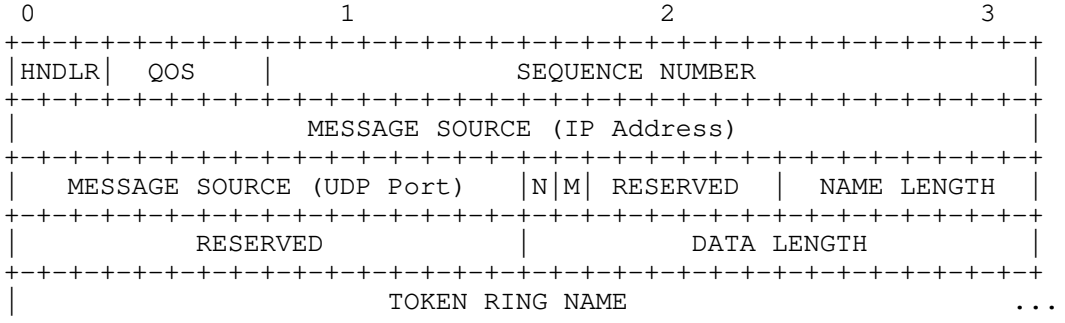
NEW TOKEN RING ID: The ID for the new token ring (which is being aborted).

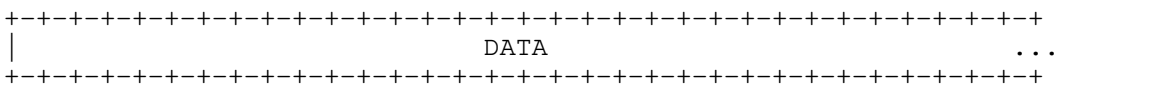
MAX TIMESTAMP: The maximum known timestamp of the old list.

Non Member Packet -- Non Member Data (Type 11)

Purpose: This packet is used to allow processes that are not members of a token ring to send data to a token ring, and optionally to have their packet handled by a member of the token ring. This packet is similar to a Data Packet, with the key difference that it may include the textual ID for a token ring instead of the 64-bit token ring ID. This packet is treated like a Data Packet, except that when a site that is the token site delivers the packet it may optionally send a Non Member ACK packet to the source. This positive ACK lets the source know that it has been delivered with the requested QOS. Another flag allows the packet to be passed to the handler multiple times. This allows the easy implementation of a multi-RPC service with all the features and guarantees of RMP. Like other packets, a non member data packet can either be sent to the IP Multicast address for the token ring or to the IP address of one of the members of the token ring (which will then multicast it to the ring).

Extra Fields:





HNDLR (Handler): This selects the handler, if any, for a data packet. There are six mutually exclusive locks included with each token ring. Based on the value of the HNDLR field, the process that holds one of these locks may be responsible for providing a response to it. Requesting that a packet be handled guarantees that at most one process will handle it. The valid values for this field are the same as for Data Packets.

QOS: This specifies the quality of service for this packet. The valid types and implementation details are the same as for Data Packets.

SEQUENCE NUMBER: Each process uses the sequence number field to sequentially order each Data Packet it creates. Each source keeps track of one sequence number for each token ring it is part of. This sequence number starts at zero and is incremented by one each time the source sends a Non Member Data Packet to that token ring. Sequence numbers wrap around after 2^24. If for two sequence numbers A and B, A < B and (B-A) > 2^23, then A is considered to be larger than B. RMP requires that no more than 2^23 packets be created over a period equal to the maximum lifetime for a datagram packet in the network. This condition holds true when IP is used as the datagram service.

MESSAGE SOURCE: Each RMP process is uniquely identified by a six byte identifier, consisting of a four byte host identifier concatenated with a two byte process identifier. RMP uses the IP address to uniquely identify hosts, and the UDP/IP port which each RMP process must open to identify the process. Kernel implementations of RMP may choose different values for the process identifier, if desired. A data packet is uniquely identified over its maximum possible lifetime by its {SEQUENCE NUM, MESSAGE SOURCE} pair. These fields, along with its HNDLR and QOS fields, are copied into the ACK that timestamps the data packet.

N (NO ACK): Do not send a Non Member ACK for this packet. This is used for RPC implementations on top of RMP that expect one or more members of the ring to send back a response to the packet.

M (MULTIPLE COPIES): Allow multiple copies of this packet to be delivered to the members of a ring. This makes the implementation of Client to Server RPC on top of RMP easier. When coupled with the N, QOS and HNDLR fields, this allows an atomic group RPC service with optional single-server handling to be easily implemented.

RESERVED: This must be set to zero.

NAME LENGTH: The length of the TOKEN RING NAME field. This includes the trailing zero after the name. In order to make the following fields word aligned, this length must be equal to an integer multiple of zero. This should be set to zero if the TOKEN RING NAME field is not used.

DATA LENGTH: The size of the data field, in bytes. The maximum value under this version is 8172 bytes.

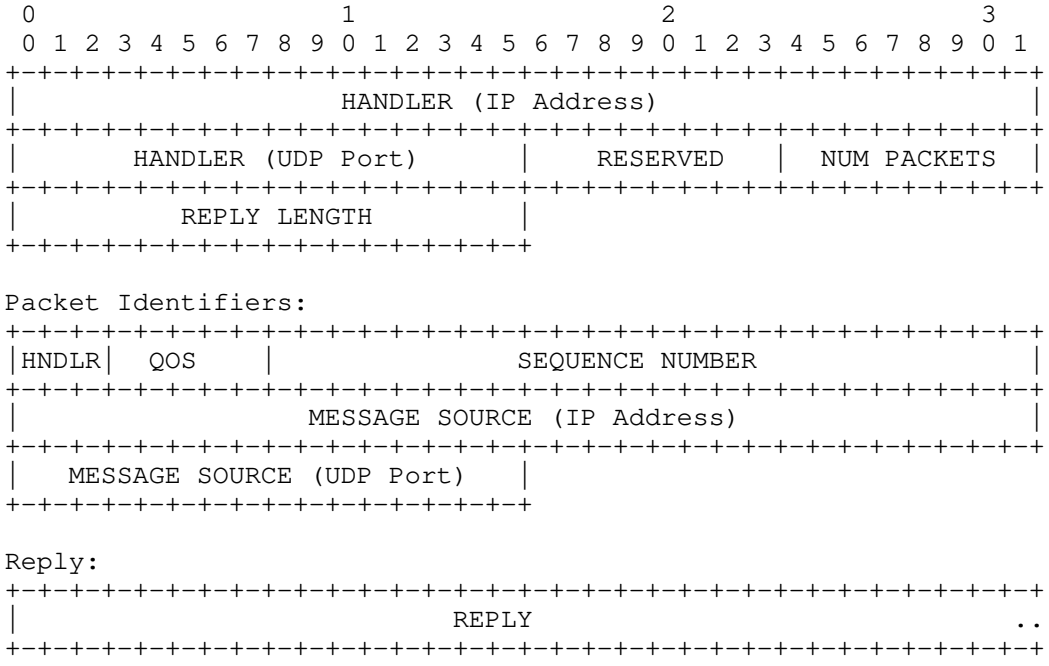
TOKEN RING NAME: The null-terminated ASCII name for this token ring. This name must end on a word boundry, which may necessitate that 1-3 extra bytes of padding be included after the trailing zero of the name. These bytes can be set to any value. This field is used when the non member does not know the Token Ring ID of the ring that it is sending to. Each Non Member ACK contains the Token Ring ID for the ring, so this can be cached and used for future Non Member Data Packets. If a packet is with a Token Ring ID and it does not get the expected Non Member ACK, then the non member source should assume that the Token Ring ID has changed for the ring and start using this field again. Exactly one of either the Token Ring ID or the first byte of the Token Ring Name fields must always be set to zero.

DATA: The data to be delivered.

Non Member Packet -- Non Member ACK (Type 12)

Purpose: This packet acknowledges the delivery of one or more Non Member Data Packets. This is either generated by the current token site when it delivers a Non Member Data Packet without the NO ACK bit set, or by the handler for a packet. For handled packets, the handling process may include an optional reply of up to 8162 bytes.

Extra Fields:



HANDLER: Optional. The identifier for the RMP process that generated this packet. This is only filled in for packets generated by clients in response to a Handler request.

NUM PACKETS: The number of packets being acknowledged. If REPLY LENGTH is not set to zero, this must be set to one.

REPLY LENGTH: The length of the optional reply field (used with handled packets). This can be between 0 and 8162.

Packet Identifiers: There are NUM PACKETS Packet Identifiers. Each one contains the header of one of the Non Member Data Packets that are being acknowledged.

REPLY: Optional. The data for the reply of a handled packet.