Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-rmt-gra-fspec-00.ps

RMT WG
Ken Calvert/UKY
Christos Papadopoulos/USC
Tony Speakman/Cisco
Don Towsley/UMASS
Swapna Yelamanchi/Cisco

13 July 2001
Expires: January 2002

Generic Router Assist - Functional Specification

Lexical Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

Abstract

This draft specifies the functional requirements which any implementation of GRA must meet. It broadly describes the context in which GRA operates, and it specifies the contents of GRA headers and GRA filter specifications. In the process, it specifies some principles of operation for GRA in the context of a router.

Table of Contents

1. Introduction

An architecture for Generic Router Assist (GRA) was introduced in [1]. This draft extends on that document by specifying the functionality of GRA in more concrete terms. It expands on the requirements on the larger context in which GRA must function by describing both GRA header contents and the filter specifications in routers against which those headers are matched and processed. In doing so, it specifies the requirements on the GRA control protocol for establishing filter specifications in routers, on the GRA signalling protocol for encoding and processing GRA headers, and on the general capabilities routers must have to be able to support GRA such as detecting and parsing GRA headers, managing GRA-related state, and handling (i.e., discarding, modifying, forwarding, etc.) GRA packets.

The intent of this draft is to provide a comprehensive list and broad description of all of the functions that must come together to implement GRA in a network: a control protocol to establish GRA filters in routers, GRA headers to be prepended to the transport headers of packets intended for GRA processing in routers, and GRA filter specifications to define the specific semantics of that processing.

2. Definitions

The definitions in this section apply in the text that follows.

Distribution Tree

The multicast distribution tree of routers defined by network-layer multicast routing information for a given multicast transport session rooted at a single root router (at the host that is the source of the data in the transport session), and fanning out (possibly through transit routers) to one or more leaf routers (at the hosts that are the receivers of the data in the transport session).

Downstream

In the direction away from the source and toward receivers.

Upstream

In the direction away from receivers and toward the source.

GRA hop

The single logical hop between any two GRA-capable routers adjacent (in a strictly upstream/downstream sense) to each other in the distribution tree (i.e., not separated in the distribution tree by any other GRA-capable router).

Neighbours

Two GRA-capable routers separated by a single GRA hop.

GRA packet

A packet bearing a GRA header.

Suppression

Potential transmitters of some packet refrain from the transmission of that packet because they see its duplicate.

Elimination

A potential forwarder of some packet refrains from forwarding that packet because it has a record of already having forwarded a duplicate of that packet.

Aggregation

Multiple identically sized operands are combined in a bit-wise fashion that results in a compositely encoded operand of the same size.

Accumulation

Multiple operands are concatenated together to form an operand list whose length is greater than length of any of the constituent operands.

3. Context and Model of Use

The assumption here is that GRA may be implemented in some fraction of the routers in a source-specific multicast distribution tree, that those GRA-capable routers discover their upstream neighbours through transport-session-specific announcements flowing down the distribution tree, that a GRA control protocol establishes filter specifications in those routers, and that the source and the receivers in the transport session direct GRA packets into the distribution tree. Those packets are detected by routers and processed against established filters the specifications of which determine, amongst other things, the subsequent fate of the packet.

In a bit more detail, GRA is designed for use primarily in the context of a source-specific multicast distribution tree, in which some subset of routers implement GRA services. Filter specifications are established in these routers for each session via GRA control protocol; each filter specification defines a set of GRA headers that may be borne by a subset of the session's packets and the processing to be applied to that subset.

Under some circumstances it may be feasible to establish the necessary session-related state without the control protocol. GRA packets are recognized as such during unicast or multicast forwarding, and are processed according to the appropriate filter specification as described below.

3.1. Establishing Filters and Sending GRA Packets

GRA filter specifications describe the (fixed) processing that can be applied to packets to implement some functionality. An example is the "duplicate elimination and subcasting" filter described in [1]. Router vendors choose which GRA services to support in their routers. The filter identifier space will be defined to accommodate both fixed, standard services, and dynamic, custom services.

Users (i.e. end system GRA implementations) control which services are applied to their packets, and (to the extent allowed by the chosen service) how that processing proceeds and affects their per-session state, by placing identifiers and operands in the GRA headers of packets sent via the distribution tree.

3.2. GRA Packet Processing

GRA packets are recognized as such during the unicast or multicast forwarding process. The GRA header is parsed to identify the session to which the packet belongs and the applicable filter specification from that session, if any. Each GRA packet matches at most one filter specification; GRA packets that fail to match any filter specification at a router are forwarded normally.

The processing defined by the service associated with the filter specification is then carried out, using the values from the GRA header and the state associated with the filter spec. This processing may modify the contents of the GRA header (only) by overwriting the values in some fields. When GRA processing concludes successfully, the packet is either discarded or forwarded, as specified by the selected service. There are two forwarding functions. Multicast forwarding on a group of interfaces or unicast forwarding to a network-layer destination.

Note that a session may have more than one filter specification. Each filter specification's state information is isolated from that of other filter specs in the same session or in other sessions. This allows different filter specs to specify the same GRA service within the same session, without interference.

4. GRA Neighbour Discovery

Since GRA requires upstream GRA neighbour information per transport session, any transport protocol that requires GRA services from routers must provide a per-transport-session upstream neighbour discovery mechanism that permits GRA-capable routers to associate any given transport session identifier with an upstream GRA neighbour. GRA-capable routers then use upstream neighbour information to address GRA packets returning from receivers toward the source on the exact reverse of the distribution tree.

Note that, due to unicast routing asymmetries, an upstream GRA packet that traverses a non-contiguous GRA hop and arrives at the neighbour on other than the interface to which it was addressed must, upon receipt, be reassociated with the interface to which it was addressed.

5. Filter Specifications

The GRA control protocol enables and disables GRA filter specifications in GRA-enabled routers. A GRA filter specification (discussed in more detail in Section 5 below) includes:

- the transport-session-specific filter identifier (TFID)

  (the predicate eliminating and subcasting filter from [1] is an example of a filter)
- the filter-specific housekeeping functions (a life timer for the availability of the filter, for example)
- the filter-specific service identifiers (FSIDs)

  (RCVR_UPDATE and FORWARD from [1] are examples of services)
- the filter-specific services:
  - the service-specific housekeeping functions (a life timer for the availability of the service, for example)
  - the service-specific GRA header formats including a (possibly null) sub-session-specific key (SKEY) (i.e., a key that further constrains the service to a specific attribute of a session such as a sequence number)

    (SQN from [1] is an examples of a sub-session-specific key)
  - the service-specific action(s) to take predicated on whether the SKEY, when specified, matches existing key-specific state; unconditionally otherwise

    (the specifications of predicates and their associated f(v)/f(s)/f(p) in [1] are examples of actions)
  - the action-specific predicates to be evaluated in applying the action-specific functions

- the action-specific functions for disposing of the GRA packet, and for transforming the (possibly key-specific) state including the (possibly key-specific) interface list
- the (possibly key-specific) state housekeeping functions

  (ET and LT from [1] are examples of key-specific state housekeeping functions)
- the key-specific state

6. The GRA Header

The GRA header is best conceived of as a pre-amble to the transport header and must be available at any layer in the communications stack at which the transport header is also available. The GRA header should be self-contained. That is, all parameters should be provided in the GRA header itself. This may necessitate duplicating information carried elsewhere in the packet, most obviously the transport session identifier.

Note that when a GRA header is present, a given transport protocol is free to refer to the GRA header for values that normally would be carried elsewhere in the packet.

The GRA header must carry the following parameters:

```
              +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       Demux  |         GHTYPE/GHSIZE             |
             /+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            / |              TSI                  |
           /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 Identifiers  |              TFID                 |
          \   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           \  |              FSID                 |
            \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             /|              SKEY                 |
            / +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
           /  |            operand 1              |
          /   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    Operands  |            operand 2              |
         \    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          \   |            operand 3              |
           \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
            \ |              .....                |
             \+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1 - Schematic GRA Header Format

GHTYPE/GHSIZE

GRA Header Type/ GRA Header Size

TSI

Transport Session Identifier TFID

Transport-session-specific Filter Identifier

FSID

Filter-specific Service Identifier

SKEY

Sub-session-specific label corresponding to key-specific state

6.1. Transport Session Identifier (TSI)

The TSI must be globally unique.

If GRA is to be truly generic and independent of even network-layer protocol context, this uniqueness must be established by the TSI alone without reference to any other qualifiers such as, say, a transport protocol identifier.

Alternatively, an implementation of GRA may be network-protocol specific and make use of network-layer protocol context to scope TSIs by the relevant network-layer protocol's transport protocol identifiers. To the extent that an implementation of GRA is network-layer-protocol specific, it is incompatible with a fully generic implementation of GRA.

The TSI must be large enough to name the universe (scoped or not) of transport sessions, but still be scalable enough to be efficiently sorted and searched in the forwarding path in routers. A TSI scoped by a transport protocol identifier must number 2^48 sessions.

The TSI is used to locate the filter specifications and filter-specific state for a given transport session.

6.2. Filter Identifier (TFID)

The TFID is the transport-session-specific name of the filter specification against which the GRA header must be matched.

The TFID must number on the order of 256 filters per transport session.

6.3. Service Identifier (FSID)

The FSID is the filter-specific name of the service to be used to process the GRA header.

The FSID must number on the order of 16 different services per filter.

6.4. Key (SKEY)

The SKEY is the sub-session-specific label to be attached to any key-specific state established by the service used to process the GRA header. It may also be treated as an operand by the service.

The SKEY must be large enough to provide substantial granularity within the transport session, but, like the TSI, still be scalable enough to be efficiently sorted and searched in the forwarding path in routers.

6.5. Operands

An implementation of GRA must support GRA headers with either a fixed number of operands in fixed formats, or a variable number of operands in variable formats. Fixed format GRA headers should be used in filter specifications where forwarding-time processing performance is the priority. Variable format GRA headers should be used in filter specifications where syntactic flexibility is the priority. Variable format GRA headers must be self-describing to the extent that the type, length, and value of all operands can be determined dynamically.

While operands may represent a whole range of meaningful (to the transport) attributes, they are not interpreted by GRA in other than the context of the operators of each type supported by GRA. That is, the numerical and relational operators treat operands as unsigned integers, and the logical operators treat operands as bit fields.

7. More on Filter Specifications

While GRA headers carry only inert identifiers and operands, it is the filter specifications in routers that embody the procedural power of GRA. A filter specification requires a fairly rich language of operators for operating on the operands in the GRA header, for forming predicates, for constructing expressions, for binary control decisions, and for maintaining state.

Numerical Operators

Addition and subtraction.
"+, -"

Logical Operators

Bit-wise AND, inclusive OR, exclusive OR, and one's complement.
"&, |, ^, ~"

Relational Operators

EQ, GT, GTE, LT, LTE.
"==, >, >=, <, <="

Expressions

Logical AND, inclusive OR, and NEGATION.
"&&, ||, !"

Control Flow

Non-compoundable "if/else".

Assignment

" = "

7.1. Identifiers

A filter specification consists first of specific values for each of the identifiers described above: a TSI for the transport session, a TFID for the specific filter within that session, and an FSID for each service provided by the filter.

7.2. Filter, Service, and Key State

Any state associated with each filter and each service provided must also be specified. For services whose actions are predicated on matching SKEY against existing key state, that key state must be specified.

Such state may consist of timers and variables. Timers may be used to expire a filter or service or key, or for other purposes. Variables may be used to maintain filter-,service-, and key-specific state. These timers and variables may be used in housekeeping functions as well as in action specifications.

7.3. Housekeeping Functions:

The only non-data driven events associated with GRA are the expirations of life timers on local state. That is, once established, a filter specification does not change and results in no non-data-driven activity other than the expiry of timers that discard the filter, its services, or any key state associated with those services.

7.4. GRA Header Formats

The specification then provides the format of the GRA header for each service possibly including an SKEY. (SKEY is just an operand that happens to be used to label filter state within the session.) Fixed format headers specify the fixed sequence and size of the operands. Variable format headers specify the syntax and size of the operand descriptors.

7.5. Actions

A single service-specific action must be specified for each service. If the service is conditional on SKEY matching existing state, two actions must be specified, one for the match and the other for the miss.

An action consists of a single predicate that may evaluate GRA header operands and local state including variables and timers.

Conditional upon the predicate, are 3 functions.

The first function specifies how to transform local state which may include filter state, service state, or key state. Transformations include assigning values to variables, setting, resetting, or clearing timers, and discarding key state.

The second function specifies how to transform the interface list associated with either the service or the key. Transformations include adding or deleting interfaces in the list, and assigning values to variables associated with interfaces in the list.

The third function specifies how to handle the GRA packet. It may be discarded (DISCARD), it may be forwarded downstream on one or more outgoing interfaces (FORWARD), or it may be forwarded upstream to the upstream GRA neighbour for the session (REVERSE). The

FORWARD function must be capable of iterating on a list of outgoing interfaces.

Before being forwarded in either direction, the operands (and only the operands) in the GRA header may be modified. Modifications consists solely of assigning new values to existing operands in the GRA header without altering the format of that header in any way.

8. Contextual State

To carry out a particular GRA action, both the identity of the incoming interface upon which a GRA packet was received and of the upstream GRA neighbour for the transport session must be available to GRA.

9. Implosion Control

10. Security Considerations

11. IANA Considerations

No information in this specification is subject to IANA registration.

Intellectual Property Issues

Acknowledgments

References

[1] Cain, B., Speakman, T., and Towsley, D., "Generic Router Assist (GRA) Building Block, Motivation and Architecture", Internet Draft draft-ietf-rmt-gra-arch-02.txt, a work in progress.

Authors' Addresses

Ken Calvert
calvert@netlab.uky.edu

Christos Papadopoulos
christos@catarina.usc.edu

Tony Speakman
speakman@cisco.com

Don Towsley
towsley@cs.umass.edu

Swapna Yelamanchi
syelaman@cisco.com