Join failure notification for PIM-SM multicast routing

draft-hilt-pim-tree-unreachability-00.txt

may 2007

updated/modified version of draft-hoerdt-pim-group-unreachable-00

B.HILT	MIPS	Haute Alsace University, Colmar - France
JJ. Pansiot	LSIIT	Louis Pasteur University, Strasbourg - France
M.Hoerdt	IL21	Lancaster University, Lancaster - UK

Outline

Background

The needs

Our proposition : a PIM-Tree Unreachable message

A short evaluation

Summary

Background on Multicast routing management

PIM-SM is

- receiver driven
- one way without feedback

This means that :

- receivers create states in routers
  malicious receivers can easily launch DDoS
  on PIM-SM control plane
- if a PIM-join fails
  (transient routing problem, misconfiguration, user error)
- then
  + netadmin and users are not informed
  + useless trees, states, cyclic joins are maintained until
    problem is fixed or receivers quit the group

Need an “ICMP-like” feedback

The needs

There is a need to

- help netadmins on the receiver side
  - making failure location and reason available
  - => to inform users and/or fix problem
- help automatically flush useless trees
  - especially important in case of DDoS

In this draft we deal only with control-plane problems,
not data plane problems (TTL problems, congestion, ...)

A simple example : DDoS attack using RPembedded

N attackers (botnet) launch an attack against a prefix P

each attacker randomly generates *k* RP embedded addresses G such that the RP address embedded in G, say R, has prefix P

For example if P is a /48, there are 2**28 syntactically correct possibilities for R (64 - 48 bits in prefix part, 4 bits in RIID part)

each attacker joins its *k* ASM groups

=> *N*k* trees (states) created in the access router for prefix P

With N = 2000 and k = 50 => 100 000 trees

=> may well overwhelm routers (and deny legitimate multicast users)

=> hard to detect on the attacker side (only *k* joins)

Similar attacks with SSM (V4 or V6) choosing *k* random source addresses

Our proposition (1/4)

A new PIM-SM message Called **PIM-TU** for **PIM-Tree Unreachability**

- containing unreachability information for one or several trees
- generated by a Pim router detecting an error/anomaly(DDoS)
- forwarded hop by hop on the outgoing interfaces of the failed tree
- Note: sent to downstream routers, **not** to the failed group address

Possibility to aggregate error information for several trees

- effective for ASM and SSM mode,
- similar messages for Ipv4 and Ipv6.

A router receiving a PIM-TU for a group/channel existing in its TIB

- flags the corresponding TIB entry
- forwards the TU to each outgoing interface of this TIB entry
  if there is a trusted PIM neighbor on this interface
- caches the TU for some duration if it is an **Edge** router for group:
  if it has directly attached receivers
  or it has an “untrusted” (eg not using TU) downstream router

Our proposition (2/4)

Usage of this PIM-TU message: inform and/or flush

Inform:

- unreachability conditions are propagated to edge routers
- they can be logged
- network admin has information on
  - which: group/channel
  - where: router unable to forward join (or unwilling)
  - why: reason of failure
- depending on the location and reason of failure
  - network admin may try to solve problem, inform users, ...

Our proposition (3/4)

Usage of this PIM-TU message: inform and/or flush

Flush:

an edge router keeps in cache the PIM-TU message

depending on the error condition,

may stop sending PIM-join messages

=> this will flush the tree upstream for the caching time

(Note: edge router could send a prune)

Particular (but important) case : DDoS

If the reason for failure indicated in the PIM-TU is DDoS

- logging with high severity may be used
- new cyclic joins may be suppressed for a long time
- IGMP-Reports from the offending interface (or host) may be filtered altogether

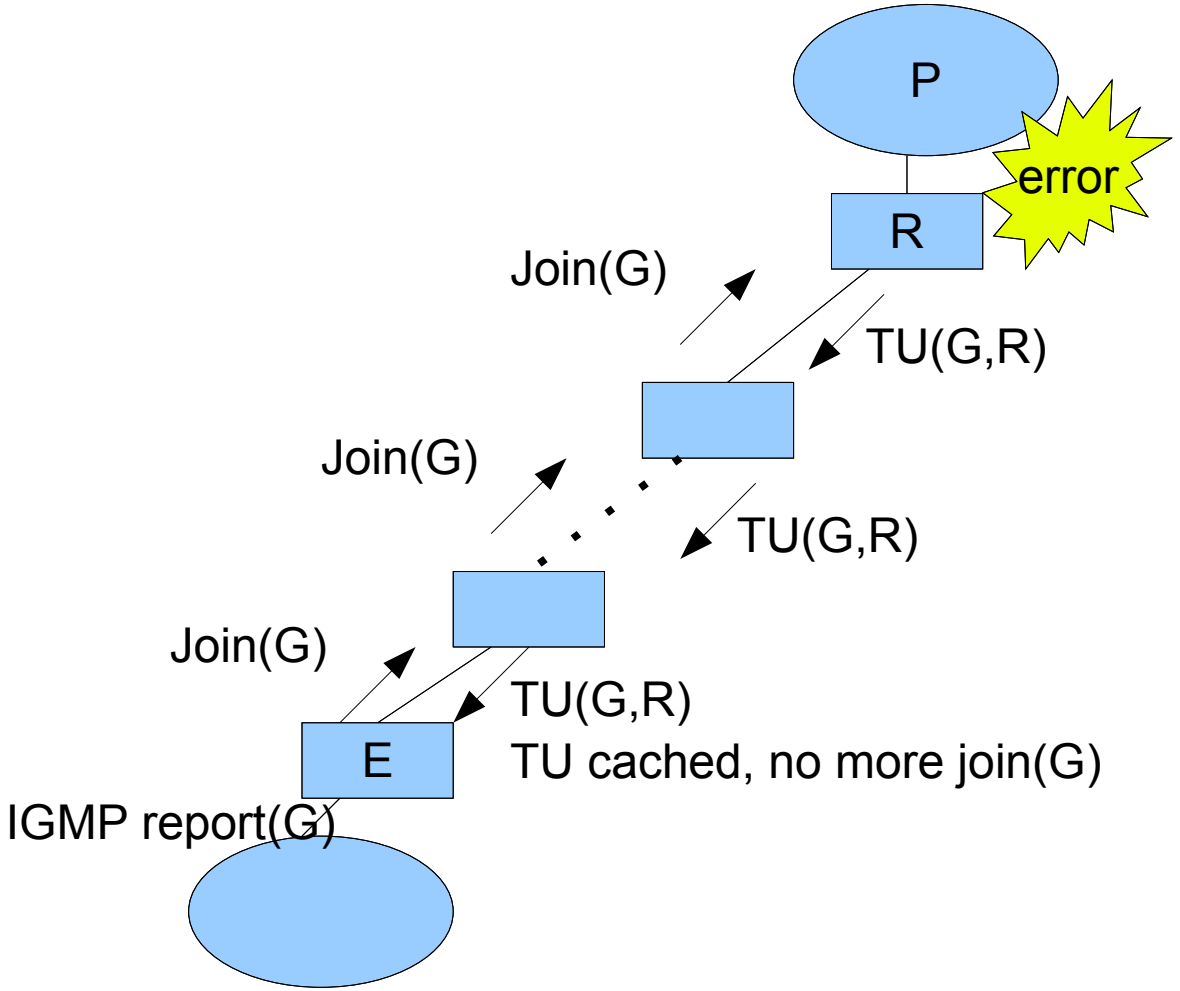
Our proposition (4/4)

Cost vs Benefit

Cost
- signaling:
  - number of PIM-TU a fraction of number of useless PIM-join messages
- memory : adds a few words per TIB entry
  - in non edge routers these entries are flushed: low cost
  - in edge routers these entries are kept a longer time
  - => the burden is on edge routers

Benefit
- less states and signaling in core (non edge) routers
- debugging information available through edge routers

DDoS example revisited



Host starts sending IGMP-Report(G) embedding address RP with prefix P

Pim-join(G) forwarded toward P
(*,G) state in intermediate routers

R detects an error
for example *RP* not a valid RP
R sends TU(G,R) to neighbor
on outgoing interface(s) for G

PIM-TU propagated hop by hop downstream
PIM-TU arrives at edge router E
E puts PIM-TU(G,R) in cache
E suppresses periodic PIM-join(G)
States for G disappears
in all routers but E
during caching duration

Relationship with other mechanisms

PIM attribute

In order to determine if a PIM neighbor implements the PIM-TU mechanism
one could use a PIM-join attribute as in
draft-ietf-pim-join-attributes-03

Relationship with mtrace (recently re-activated)
draft-asaeda-mboned-mtrace-v2-00

	MTRACE	PIM-TU
needs router participation	yes	yes
routing protocol	any	PIM
initiator	netadmin (manual)	upstream routers (automatic)
error diagnostic	yes	yes
data plane error TTL/congestion	yes	no
DDoS detection and filter	no	yes

Seems that the two tools are complementary,
could share some common error codes

Summary

Our proposition of a PIM-TU feedback message allows to:

- suppress useless trees branches (depending on failure reason)
- block DDoS attacks as close as possible to attackers
- give administrators helpful debugging information
- users may get failure information from their local netadmin
- or possibly from a local looking glass

Relatively simple mechanism

Keypoint: find good values for cache timers