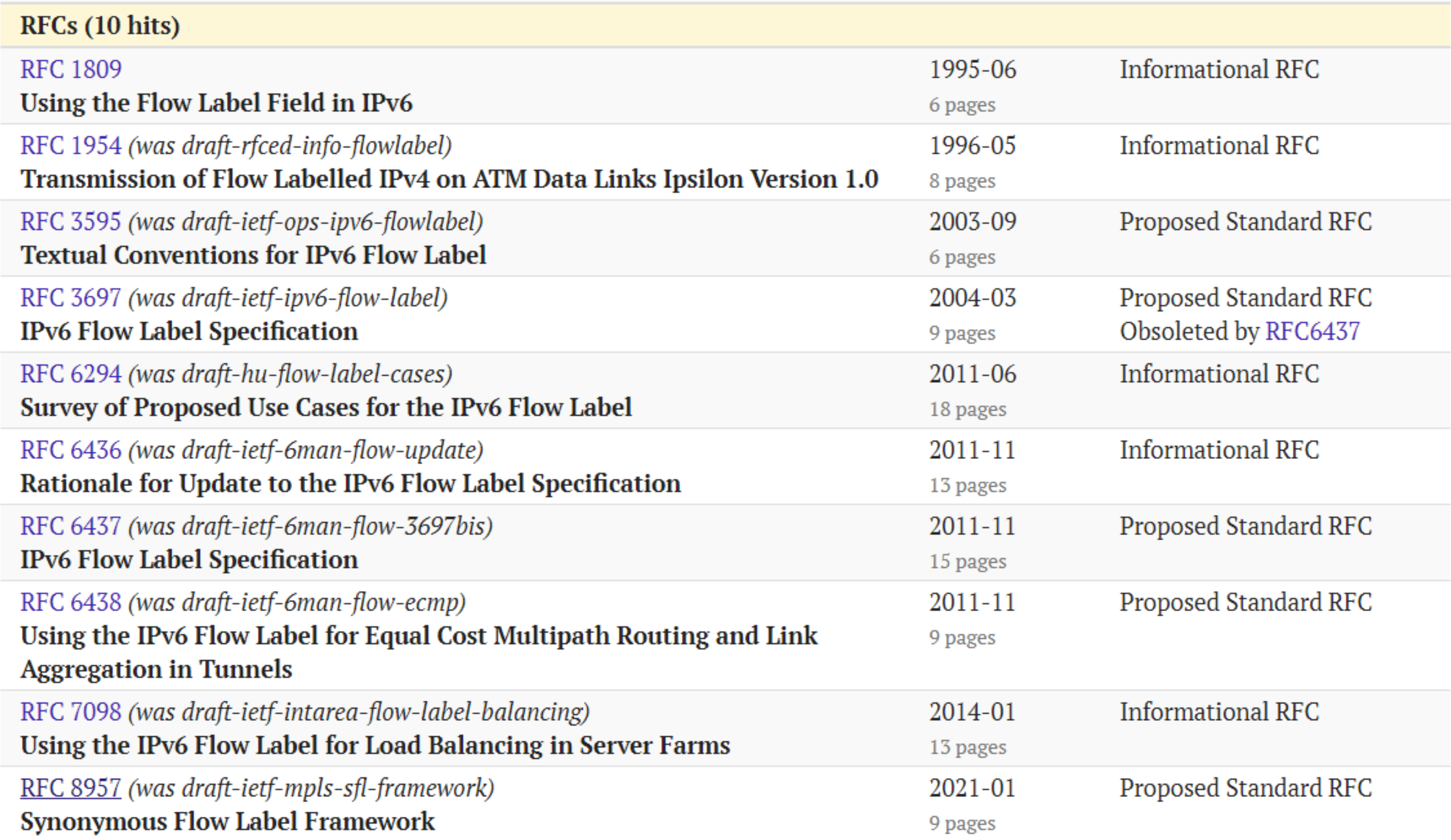
TCP Socket Hash & Flow Label

Alexander Azimov, Yandex

a.e.azimov@gmail.com

Homework



RFCs (10 hits)		
RFC 1809 **Using the Flow Label Field in IPv6**	1995-06 6 pages	Informational RFC
RFC 1954 *(was draft-rfced-info-flowlabel)* **Transmission of Flow Labelled IPv4 on ATM Data Links Ipsilon Version 1.0**	1996-05 8 pages	Informational RFC
RFC 3595 *(was draft-ietf-ops-ipv6-flowlabel)* **Textual Conventions for IPv6 Flow Label**	2003-09 6 pages	Proposed Standard RFC
RFC 3697 *(was draft-ietf-ipv6-flow-label)* **IPv6 Flow Label Specification**	2004-03 9 pages	Proposed Standard RFC Obsoleted by RFC6437
RFC 6294 *(was draft-hu-flow-label-cases)* **Survey of Proposed Use Cases for the IPv6 Flow Label**	2011-06 18 pages	Informational RFC
RFC 6436 *(was draft-ietf-6man-flow-update)* **Rationale for Update to the IPv6 Flow Label Specification**	2011-11 13 pages	Informational RFC
RFC 6437 *(was draft-ietf-6man-flow-3697bis)* **IPv6 Flow Label Specification**	2011-11 15 pages	Proposed Standard RFC
RFC 6438 *(was draft-ietf-6man-flow-ecmp)* **Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels**	2011-11 9 pages	Proposed Standard RFC
RFC 7098 *(was draft-ietf-intarea-flow-label-balancing)* **Using the IPv6 Flow Label for Load Balancing in Server Farms**	2014-01 13 pages	Informational RFC
RFC 8957 *(was draft-ietf-mpls-sfl-framework)* **Synonymous Flow Label Framework**	2021-01 9 pages	Proposed Standard RFC

RFC3697 - RFC6437
IPv6 Flow Label Specification

…flow is not necessarily 1:1 mapped to a transport connection.

A specific goal is to enable and encourage the use of the flow label for various forms of stateless load distribution…

Once set to a non-zero value, the Flow Label is expected to be delivered unchanged to the destination node(s)

It is therefore RECOMMENDED that source hosts support the flow label by setting the flow label field for all packets of a given flow to the same value chosen from an approximation to a discrete uniform distribution.

RFC6294

Survey of Proposed Use Cases for the IPv6 Flow Label

…The 3-tuple {source address, destination address, flow label} uniquely identifies which packets belong to which particular flow.

…By using the 3-tuple, we only use the IP layer to classify packets, without needing any transport-layer information.

RFC6436

Rationale for Update to the IPv6 Flow Label Specification

…a router is allowed to combine the flow label value with other data in order to produce a uniformly distributed hash.

…flow label for various forms of stateless load distribution is the best simple application for it.

The flow label is no longer unrealistically asserted to be strictly immutable;

RFC6438
Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels

…the term "flow" to represent a sequence of packets that may be identified by either the source and destination IP addresses alone {2-tuple} or the source IP address, destination IP address, protocol number, source port number, and destination port number {5-tuple}.

The sending TEP MAY perform stateless flow label assignment by using a suitable 20-bit hash of the inner IP header's 2-tuple or 5-tuple as the flow label value.

RFC7098

Using the IPv6 Flow Label for Load Balancing in Server Farms

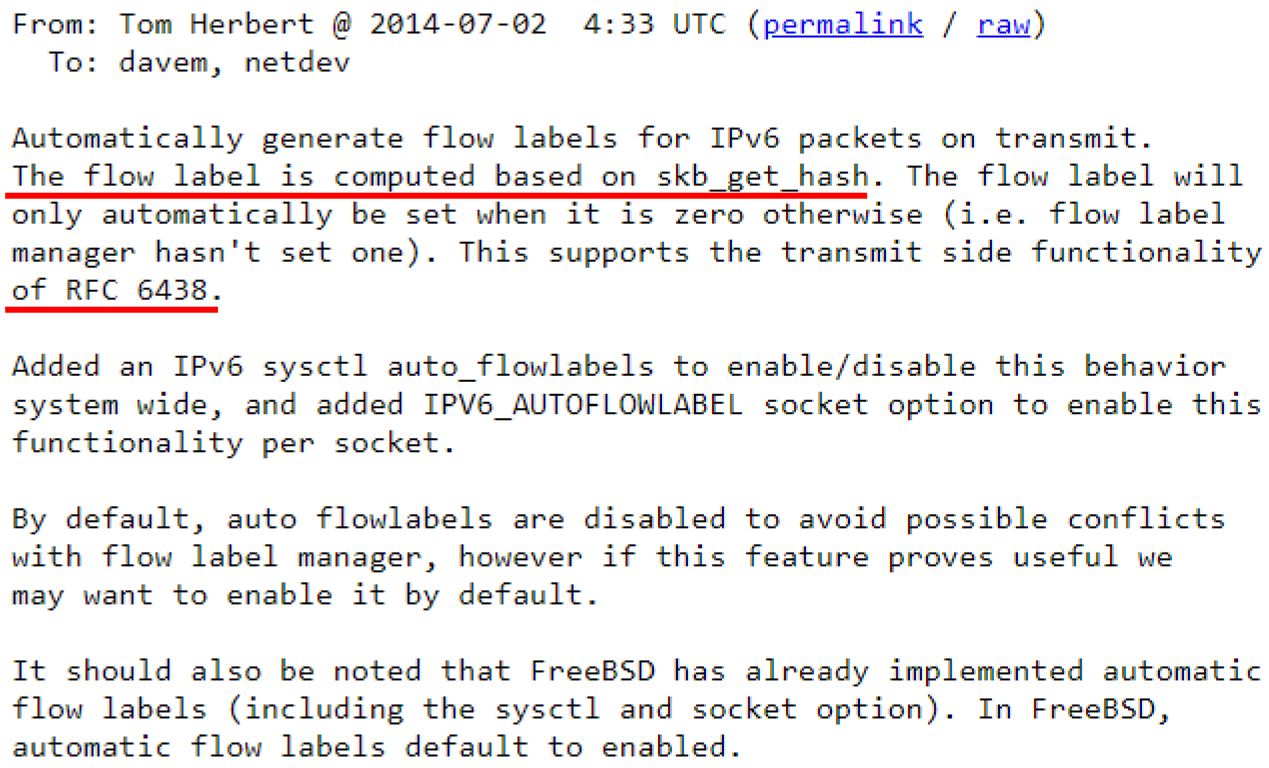
The motivation for this approach is to improve the performance of most types of layer 3/4 load balancers, especially for traffic including multiple IPv6 extension headers and in particular for fragmented packets.

…flow label should be set to a constant value for a given traffic flow

…flow label value must be constant for a given transport session, normally identified by the IPv6 and Transport header 5-tuple

Linux Kernel

2014



From: Tom Herbert @ 2014-07-02 4:33 UTC (permalink / raw)
  To: davem, netdev

Automatically generate flow labels for IPv6 packets on transmit. The flow label is computed based on skb_get_hash. The flow label will only automatically be set when it is zero otherwise (i.e. flow label manager hasn't set one). This supports the transmit side functionality of RFC 6438.

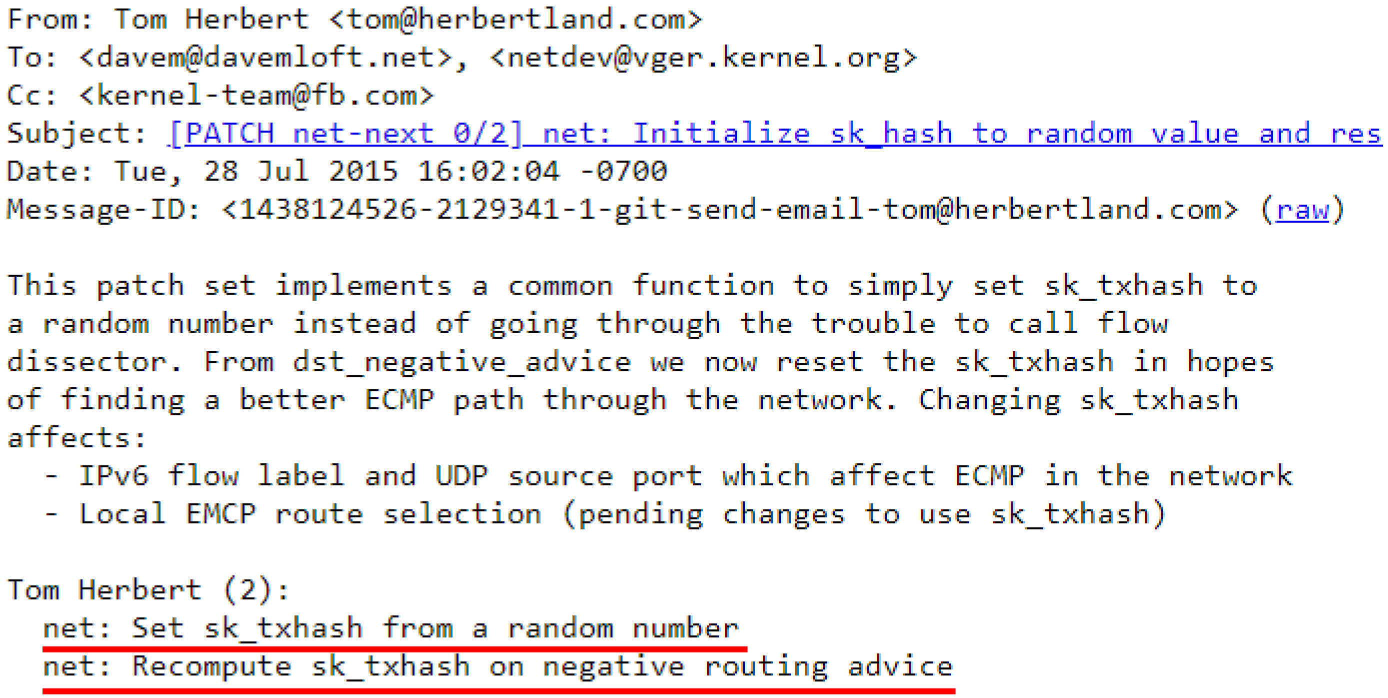
Added an IPv6 sysctl auto_flowlabels to enable/disable this behavior system wide, and added IPV6_AUTOFLOWLABEL socket option to enable this functionality per socket.

By default, auto flowlabels are disabled to avoid possible conflicts with flow label manager, however if this feature proves useful we may want to enable it by default.

It should also be noted that FreeBSD has already implemented automatic flow labels (including the sysctl and socket option). In FreeBSD, automatic flow labels default to enabled.

Linux Kernel

2015



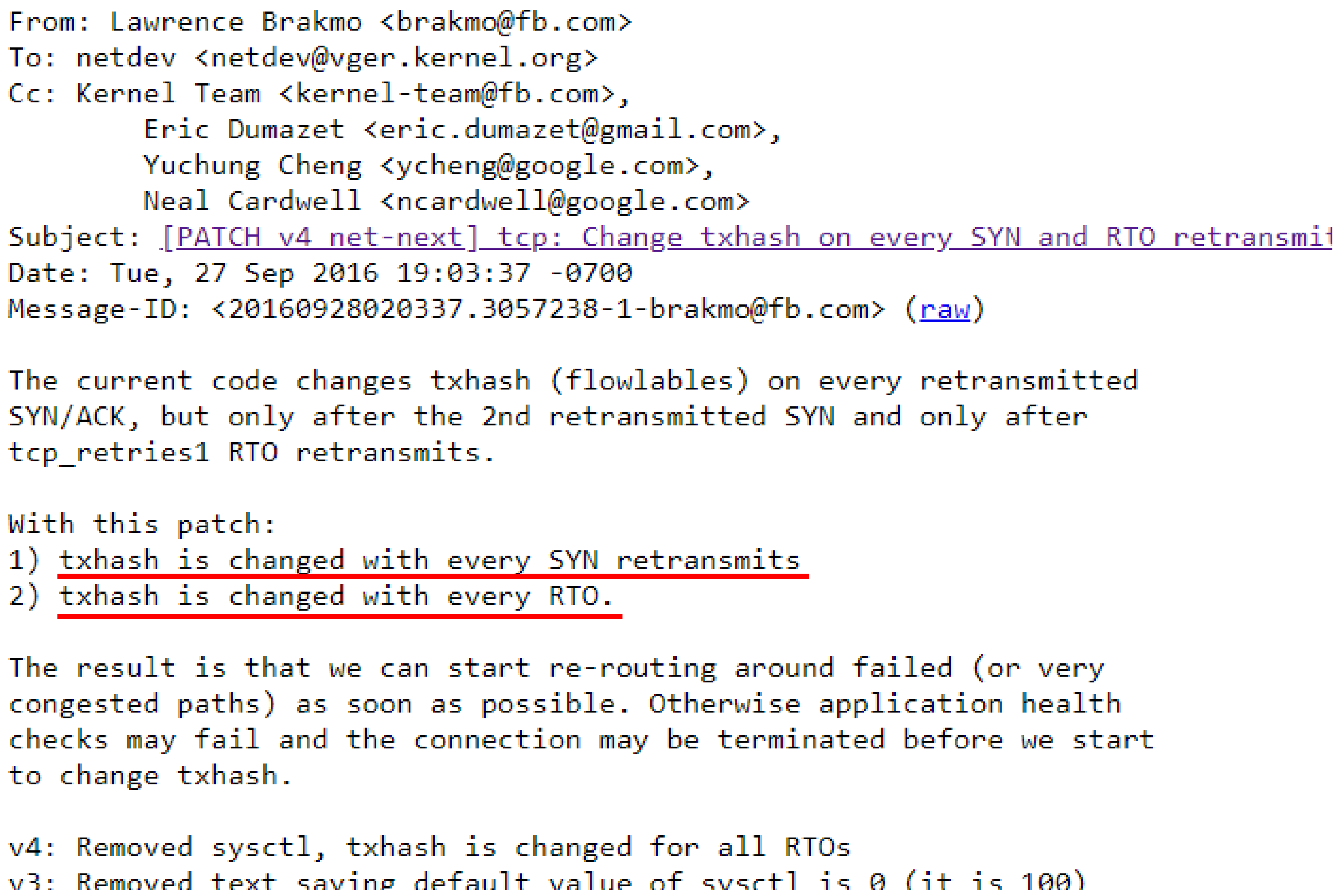
```
From: Tom Herbert <tom@herbertland.com>
To: <davem@davemloft.net>, <netdev@vger.kernel.org>
Cc: <kernel-team@fb.com>
Subject: [PATCH net-next 0/2] net: Initialize sk_hash to random value and res
Date: Tue, 28 Jul 2015 16:02:04 -0700
Message-ID: <1438124526-2129341-1-git-send-email-tom@herbertland.com> (raw)

This patch set implements a common function to simply set sk_txhash to
a random number instead of going through the trouble to call flow
dissector. From dst_negative_advice we now reset the sk_txhash in hopes
of finding a better ECMP path through the network. Changing sk_txhash
affects:
  - IPv6 flow label and UDP source port which affect ECMP in the network
  - Local EMCP route selection (pending changes to use sk_txhash)

Tom Herbert (2):
  net: Set sk_txhash from a random number
  net: Recompute sk_txhash on negative routing advice
```

Linux Kernel

2016



From: Lawrence Brakmo <brakmo@fb.com>
To: netdev <netdev@vger.kernel.org>
Cc: Kernel Team <kernel-team@fb.com>,
Eric Dumazet <eric.dumazet@gmail.com>,
Yuchung Cheng <ycheng@google.com>,
Neal Cardwell <ncardwell@google.com>
Subject: [PATCH v4 net-next] tcp: Change txhash on every SYN and RTO retransmit
Date: Tue, 27 Sep 2016 19:03:37 -0700
Message-ID: <20160928020337.3057238-1-brakmo@fb.com> (raw)

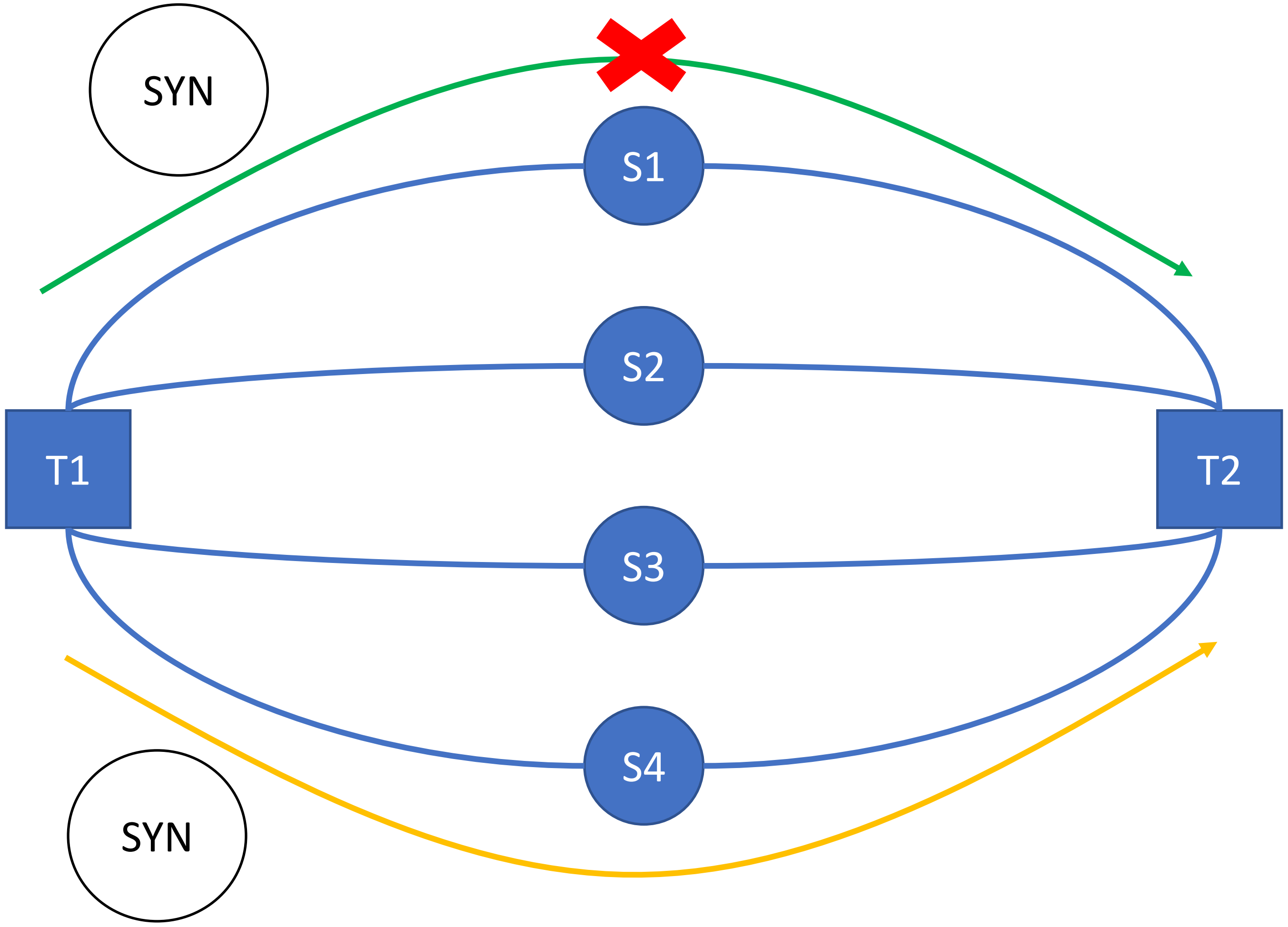
The current code changes txhash (flowlables) on every retransmitted SYN/ACK, but only after the 2nd retransmitted SYN and only after tcp_retries1 RTO retransmits.

With this patch:
1) txhash is changed with every SYN retransmits
2) txhash is changed with every RTO.

The result is that we can start re-routing around failed (or very congested paths) as soon as possible. Otherwise application health checks may fail and the connection may be terminated before we start to change txhash.

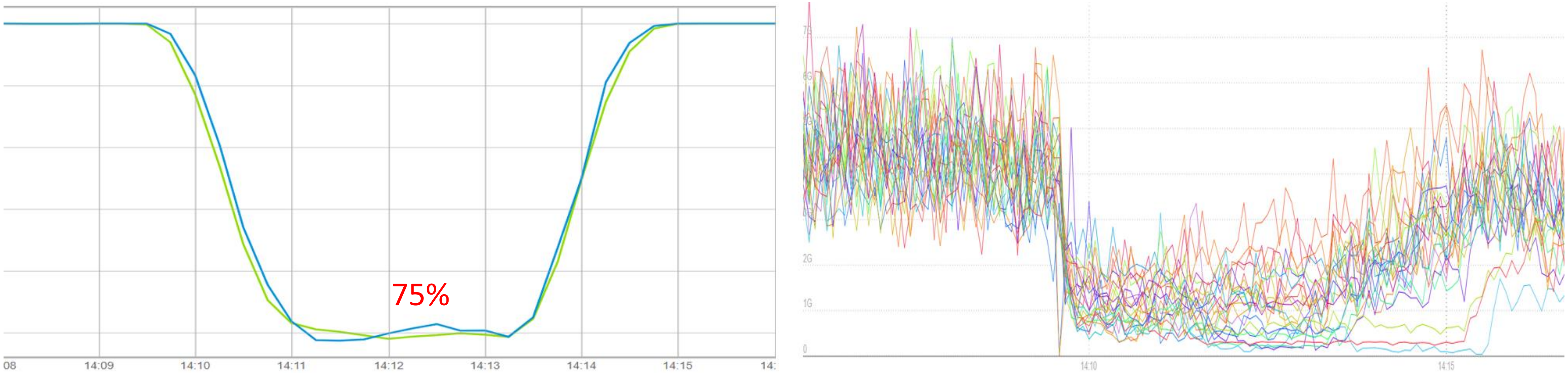
v4: Removed sysctl, txhash is changed for all RTOs
v3: Removed text saying default value of sysctl is 0 (it is 100)

Self-healing Network



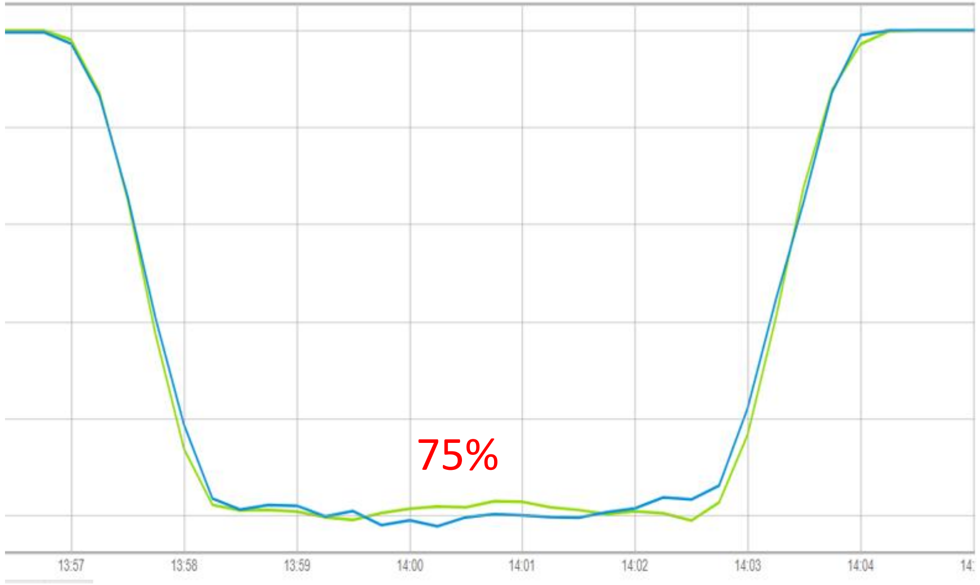
hash(proto, src ip, dst ip, src port, dst port, flow label)

Evaluation: Flow Label Balancing Off

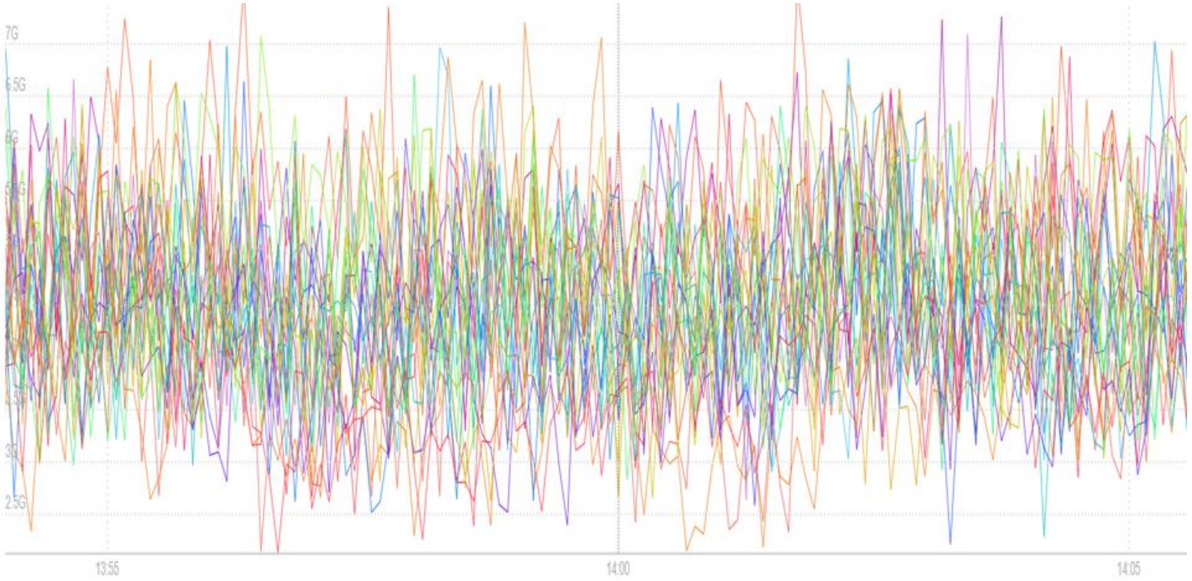


One of four ToR uplinks drops packets, significant service degradation

Evaluation: Flow Label Balancing On

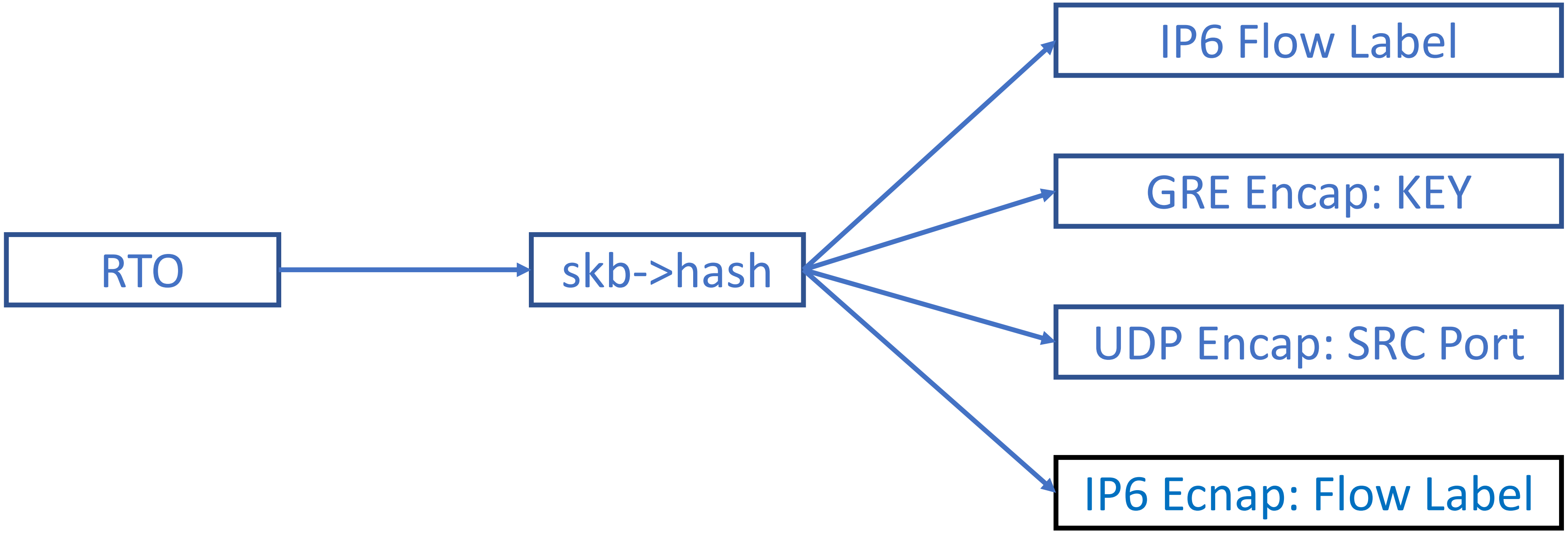


75%



One of four ToR uplink drops packets, no effect on the service!

TCP RTO & skb->hash



auto_flowlabels

0: automatic flow labels are completely disabled

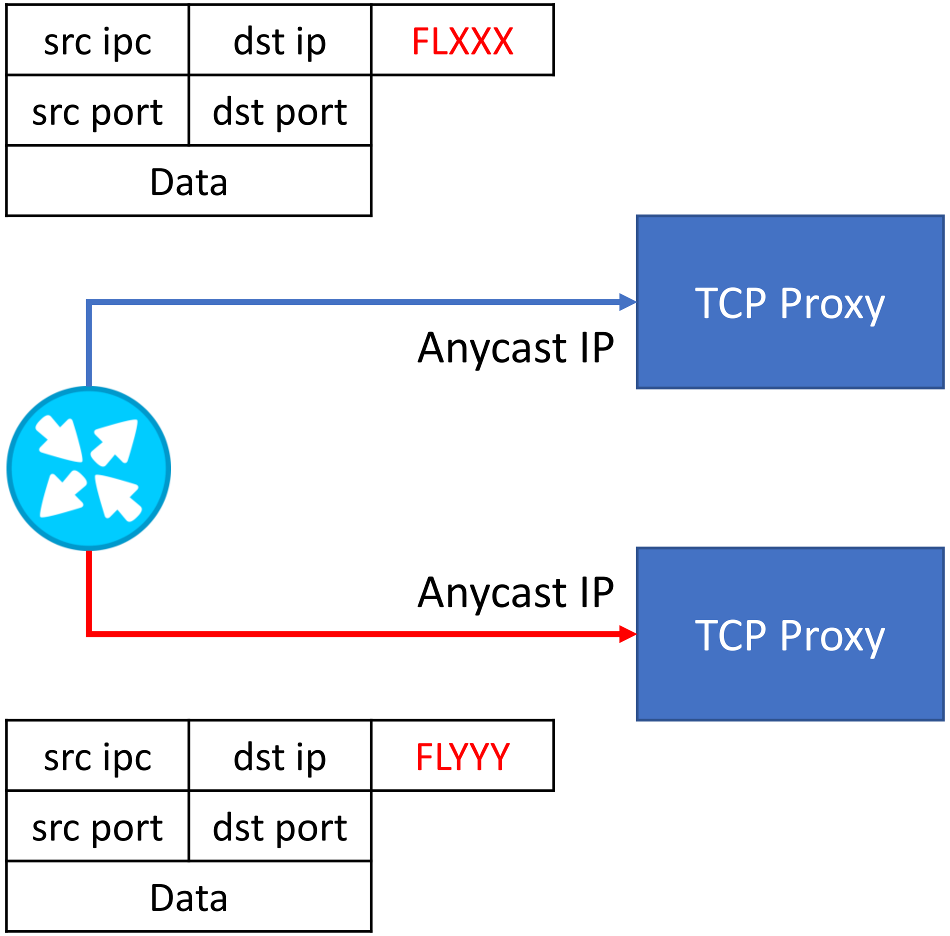
1: automatic flow labels are enabled by default, they can be disabled on a per socket basis using the IPV6_AUTOFLOWLABEL socket option

2: automatic flow labels are allowed, they may be enabled on a per socket basis using the IPV6_AUTOFLOWLABEL socket option

3: automatic flow labels are enabled and enforced, they cannot be disabled by the socket option

Default: 1

Side Effect



src ipc	dst ip	FLXXX
src port	dst port	
Data		

src ipc	dst ip	FLYYY
src port	dst port	
Data		

Hash change at client may break TCP connection!

TCP Hash: Safe Mode

Client – sends SYN, Server – responds with SYN&ACK

- In case of SYN_RTO or RTO events Server SHOULD recalculate its TCP socket hash, thus change Flow Label. This behavior MAY be switched on by default;
- In case of SYN_RTO or RTO events Client MAY recalculate its TCP socket hash, thus change Flow Label. This behavior MUST be switched off by default;

Flow Label: Status

- Flow Label isn’t used in stateful load balancing;
- Flow label is actively used in stateless load balancing;
- The 1:1 mapping between TCP flows and flow label was never guaranteed and doesn’t really exist;
- TCP hash calculation isn’t standardized, though actively used;
- Current TCP hash calculation defaults can cause session timeout;
- Some related RFCs look obsolete.