Independent Stream
Internet-Draft
Intended status: Informational
Expires: 3 June 2024

J. Evans
O. Pylypenko
Amazon
J. Haas
Juniper Networks
A. Kadosh
Cisco Systems, Inc.
1 December 2023

An Information Model for Packet Discard Reporting
draft-opsawg-evans-discardmodel-01

Abstract

~~Router~~ Router-reported packet loss is ~~the~~ among key ~~primary~~ signals that are used to infer ~~of~~ when a network is not ~~doing its job~~behaving as expected. Some packet loss is normal or intended in ~~TCP/~~IP networks, however. ~~To minimise network packet loss through~~For the sake of highly automated network operations ~~requires~~, it is required to expose clear, reliable, and accurate signals of all packets which are dropped and the reasons why. This document defines an information model for packet loss reporting, which classifies these signals to enable automated network mitigation of unintended packet loss. Interpretation of these signals and how they trigger mitigation actions are out of scope, though.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 June 2024.

Copyright Notice

Commenté [BMI1]: I don't think you meant https://www.rfc-editor.org/about/independent/

Commenté [BMI2]: That would be ideal, however som filtering may be needed to avoid expose too "noise" to applications that will consume the data. This can also be a function of the services that are provided over a specific network.

provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The primary function of a network is to transport packets. Understanding both where and why packet loss occurs is essential for effective network operation. Router-reported packet loss is the most direct signal for network operations to identify customer impact resulting from unintended packet loss. Accurate accounting of packet loss is not enough, however, as some level of packet loss is normal in ~~TCP/~~IP networks. In automating network operations, there are ~~only a relatively small number~~a set of automated actions that can be taken to mitigate customer-impacting packet loss. Action triggering depends on the diagnostic outcome. Hence, precise classification of packet loss signals is crucial both to ensure that ~~customer impacting~~specific packet loss is detected and that the right action(s) is taken to mitigate the impact, as taking the wrong action can make problems worse.

The existing reported metrics for packet loss, as defined in [RFC1213] - namely ifInDiscards, ifOutDiscards, ifInErrors, ifOutErrors - do not provide sufficient precision to automatically identify the cause of the loss and mitigate the impact. Concretely, these metrics can be used to detect abnormal network symptoms but do not help much in identifying root causes. From a network operator's perspective, ifInDiscards can represent both intended packet loss (~~i.e.~~e.g., packets discarded due to policy) and unintended packet loss (e.g., packets dropped in error). Furthermore, these definitions are ambiguous, as vendors can and have implemented them differently. In some implementations, ifInErrors accounts only for errored packets that are dropped, while in others, it accounts for all errored packets, whether they are dropped or not. Many implementations support more discard metrics than these; where they do, they have

Commenté [BMI3]: Idem as for the comments in the abstract.

Commenté [BMI4]: I agree this is the more visible one, but other types of traffic is important to monitor.

Commenté [BMI5]: To differentiate these metrics vs measured perf metrics such as https://datatracker.ietf.org/doc/html/rfc7680

been inconsistently implemented due to the lack of a clearly defined classification scheme and semantics for packet loss reporting.

Hence, this document defines an information model for packet loss reporting, aiming to address these issues by presenting a packet loss classification scheme that can enable automated mitigation of unintended packet loss, in particular. Consistent with [RFC3444], ~~This~~ this information model is independent of any specific implementations or protocols used to transport the data ~~[RFC3444]~~. There are multiple ways that this information model could be implemented (i.e., data models), including SNMP [RFC1157], NETCONF [RFC6241] / YANG [RFC7950], and IPFIX [RFC5153], but they are outside of the scope of this document.

Section 2 describes the problem. Section 3 defines the information model and semantics with examples. Section 4 provides examples of discard signal-to-cause-to-auto-mitigation action mapping. Appendix B details the authors' experience from implementing this model.

The terms 'packet drop' and 'discard' are considered equivalent and are used interchangeably in this document.

X. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Problem statement

Working backwards from the goal of auto-mitigation of unintended packet loss, there are only a relatively small number of potential actions than can be taken to auto-mitigate customer impacting packet loss:

1. Take a device, link, or set of devices and/or links out of service.
2. Return a device, link, or set of devices and/or links back into service.
3. Move traffic to other links or devices.
4. Roll back a recent change to a device that might have caused the problem.
5. Escalate to a human (e.g., network operator) as a last resort.

A precise signal of impact is crucial, as taking the wrong action can be worse than taking no action. For example, taking a congested device out of service can make congestion worse by moving the traffic to other links or devices, which are already congested.

To detect whether router-reported packet loss is a problem and to

Commenté [BMI6]: You may mention for example that some reporting means such as https://www.rfc-editor.org/rfc/rfc7270.html (frowarding status) lack clarity. See for example the values under **Status 10b: Dropped**

Commenté [BMI7]: This is not specific to this IM, IMs a not DMs as per RFC3444.

Commenté [BMI8]: This is tricky as sometimes the issu related to suboptimal configuration parameters or becaus some default value was "blindly" used.
Testing and validation may soften some of these issues, b do not nullify them.

determine what actions should be taken to mitigate the impact and remediate the cause, depends on four primary features of the packet loss signal:

1. The cause of the loss.

2. The rate and/or degree of the loss.

3. The duration of the loss.

4. The location of the loss.

Features 2, 3, and 4 are already addressed with passive monitoring statistics, for example, obtained with SNMP [RFC1157] / MIB-II [RFC1213] or NETCONF [RFC6241] / YANG [RFC7950]. Feature 1, however, is dependent on the classification scheme used for packet loss reporting. In the next section, we define a new classification scheme to address this problem.

3. Information ~~model~~Model

The classification scheme is defined as a tree which follows the structure component/direction/type/layer/sub-type/sub-sub-type/.../metric, where:
a. component can be interface|device|control_plane|flow
b. direction can be ingress|egress
c. type can be traffic|discards, where traffic accounts for packets successfully received or transmitted, and discards account for packet drops
d. layer can be l2|l3

```
.
|-- interface/
|   |-- ingress/
|   |   |-- traffic/
|   |   |   |-- l2/
|   |   |   |   |-- frames
|   |   |   |   `-- bytes
|   |   |   |-- l3/
|   |   |   |   |-- v4/
|   |   |   |   |   |-- packets
|   |   |   |   |   |-- bytes
|   |   |   |   |   |-- unicast/
|   |   |   |   |   |   |-- packets
|   |   |   |   |   |   `-- bytes
|   |   |   |   |   `-- multicast/
|   |   |   |   |       |-- packets
|   |   |   |   |       `-- bytes
|   |   |   |   `-- v6/
|   |   |   |       |-- packets
|   |   |   |       |-- bytes
|   |   |   |       |-- unicast/
|   |   |   |       |   |-- packets
|   |   |   |       |   `-- bytes
|   |   |   |       `-- multicast/
|   |   |   |           |-- packets
|   |   |   |           `-- bytes
|   |   |   `-- qos/
```

Commenté [BMI9]: As the data my be be aggregated a filtered (per service in some deployments), I wonder whether the document can discuss how such aggregation (network--node--if) can be done.

Commenté [BMI10]: Should time matters be included the IM (e.g., discontinuity-time)? That information will trigger how the reported information will be consumed

```
|   |   |       |-- class_0/
|   |   |       |   |-- packets
|   |   |       |   `-- bytes
|   |   |       |-- ...
|   |   |       `-- class_n/
|   |   |           |-- packets
|   |   |           `-- bytes
|   |   `-- discards/
|   |       |-- l2/
|   |       |   |-- frames
|   |       |   `-- bytes
|   |       |-- l3/
|   |       |   |-- v4/
|   |       |   |   |-- packets
|   |       |   |   |-- bytes
|   |       |   |   |-- unicast/
|   |       |   |   |   |-- packets
|   |       |   |   |   `-- bytes
|   |       |   |   `-- multicast/
|   |       |   |       |-- packets
|   |       |   |       `-- bytes
|   |       |   `-- v6/
|   |       |       |-- packets
|   |       |       |-- bytes
|   |       |       |-- unicast/
|   |       |       |   |-- packets
|   |       |       |   `-- bytes
|   |       |       `-- multicast/
|   |       |           |-- packets
|   |       |           `-- bytes
|   |       |-- errors/
|   |       |   |-- l2/
|   |       |   |   `-- rx/
|   |       |   |       |-- frames
|   |       |   |       |-- crc_error/
|   |       |   |       |   `-- frames
|   |       |   |       |-- invalid_mac/
|   |       |   |       |   `-- frames
|   |       |   |       |-- invalid_vlan/
|   |       |   |       |   `-- frames
|   |       |   |       `-- invalid_frame/
|   |       |   |           `-- frames
|   |       |   |-- l3/
|   |       |   |   |-- rx/
|   |       |   |   |   |-- packets
|   |       |   |   |   |-- checksum_error/
|   |       |   |   |   |   `-- packets
|   |       |   |   |   |-- mtu_exceeded/
|   |       |   |   |   |   `-- packets
|   |       |   |   |   |-- invalid_packet/
|   |       |   |   |   |   `-- packets
|   |       |   |   |   `-- ttl_expired/
|   |       |   |   |       `-- packets
|   |       |   |   `-- no_route/
|   |       |   |       `-- packets
|   |       |   `-- local/
|   |       |       |-- packets
|   |       |       `-- hw/
```

Commenté [BMI11]: Some network devices act on L4 (NAT64, for example). Are those devices in scope?

```
|   |       |           |-- packets
|   |       |           `-- parity_error/
|   |       |               `-- packets
|   |       |-- policy/
|   |       |   `-- l3/
|   |       |       |-- packets
|   |       |       |-- acl/
|   |       |       |   `-- packets
|   |       |       |-- policer/
|   |       |       |   |-- packets
|   |       |       |   `-- bytes
|   |       |       |-- null_route/
|   |       |       |   `-- packets
|   |       |       `-- rpf/
|   |       |           `-- packets
|   |       `-- no_buffer/
|   |           |-- class_0/
|   |           |   |-- packets
|   |           |   `-- bytes
|   |           |-- ...
|   |           `-- class_n/
|   |               |-- packets
|   |               `-- bytes
|   `-- egress/
|       |-- traffic/
|       |   |-- l2/
|       |   |   |-- frames
|       |   |   `-- bytes
|       |   |-- l3/
|       |   |   |-- v4/
|       |   |   |   |-- packets
|       |   |   |   |-- bytes
|       |   |   |   |-- unicast/
|       |   |   |   |   |-- packets
|       |   |   |   |   `-- bytes
|       |   |   |   `-- multicast/
|       |   |   |       |-- packets
|       |   |   |       `-- bytes
|       |   |   `-- v6/
|       |   |       |-- packets
|       |   |       |-- bytes
|       |   |       |-- unicast/
|       |   |       |   |-- packets
|       |   |       |   `-- bytes
|       |   |       `-- multicast/
|       |   |           |-- packets
|       |   |           `-- bytes
|       |   `-- qos/
|       |       |-- class_0/
|       |       |   |-- packets
|       |       |   `-- bytes
|       |       |-- ...
|       |       `-- class_n/
|       |           |-- packets
|       |           `-- bytes
|       `-- discards/
|           |-- l2/
|           |   |-- frames
```

Commenté [BMI12]: Do we include inline ddos protect policies here?

Commenté [BMI13]: For IPv6, there are drops related extension headers.

Commenté [BMI14]: Can be layer 2 as well.

```
|           |   `-- bytes
|           |-- l3/
|           |   |-- v4/
|           |   |   |-- packets
|           |   |   |-- bytes
|           |   |   |-- unicast/
|           |   |   |   |-- packets
|           |   |   |   `-- bytes
|           |   |   `-- multicast/
|           |   |       |-- packets
|           |   |       `-- bytes
|           |   `-- v6/
|           |       |-- packets
|           |       |-- bytes
|           |       |-- unicast/
|           |       |   |-- packets
|           |       |   `-- bytes
|           |       `-- multicast/
|           |           |-- packets
|           |           `-- bytes
|           |-- errors/
|           |   |-- l2/
|           |   |   `-- tx/
|           |   |       `-- frames
|           |   `-- l3/
|           |       `-- tx/
|           |           `-- packets
|           |-- policy/
|           |   `-- l3/
|           |       |-- acl/
|           |       |   `-- packets
|           |       `-- policer/
|           |           |-- packets
|           |           `-- bytes
|           `-- no_buffer/
|               |-- class_0/
|               |   |-- packets
|               |   `-- bytes
|               |-- ...
|               `-- class_n/
|                   |-- packets
|                   `-- bytes
`-- control_plane/
    `-- ingress/
        |-- traffic/
        |   |-- packets
        |   `-- bytes
        `-- discards/
            |-- packets
            |-- bytes
            `-- policy/
                `-- packets
```

For additional context, Appendix A provides an example of where packets may be dropped in a device.

3.1. Discard Class Descriptions

discards/policy/
These are intended discards, meaning packets dropped due to a configured policy. There are multiple sub-classes.

discards/error/l2/rx/
Frames dropped due to errors in the received L2 frame. There are multiple sub-classes, such as those resulting from failing CRC, invalid header, invalid MAC address, or invalid VLAN.

discards/error/l3/rx/
These drops occur due to errors in the received packet, indicating an upstream problem rather than an issue with the device dropping the errored packets. There are multiple sub-classes, including header checksum errors, MTU exceeded, and invalid packet, i.e., due to incorrect version, incorrect header length, or invalid options.

discards/error/l3/rx/ttl_expired
There can be multiple causes for TTL-exceed (or Hop Limit) drops: i) trace-route;
ii) TTL (Hop Limit) set too low by the end-system; iii) routing loops.

discards/error/l3/no_route/
Discards occur due to a packet not matching any route.

discards/error/local/
A device may drop packets within its switching pipeline due to internal errors, such as parity errors. Any errored discards not explicitly assigned to the above classes are also accounted for here.

discards/no_buffer/
Discards occur due to no available buffer to enqueue the packet. These can be tail-drop discards or due to an active queue management algorithm, such as RED [RED93] or CODEL [RFC8289].

3.2. Semantics

Rules 1-10 relate to packets forwarded by the device; rule 11 relates to packets destined to/from the device:

1. All instances of frame or packet receipt, transmission, and drops MUST be reported.

2. All instances of frame or packet receipt, transmission, and drops SHOULD be attributed to the physical or logical interface of the device where they occur.

3. An individual frame MUST only be accounted for by either the L2 traffic class or the L2 discard classes within a single direction, i.e., ingress or egress.

4. An individual packet MUST only be accounted for by either the L3 traffic class or the L3 discard classes within a single direction, i.e., ingress or egress.

5. A frame accounted for at L2 MUST NOT be accounted for at L3 and vice versa.

6. The aggregate L2 and L3 traffic and discard classes MUST account

Commenté [BMI15]: You may clarify when this is not possible/required.

Commenté [BMI16]: Should we set a requirement for implems to expose the logic they follow (L2 or L3)?

for all underlying packets received, transmitted, and dropped across all other classes.

7. The aggregate ~~qos~~ QoS traffic and discard (no buffer) classes MUST account for all underlying packets received, transmitted, and dropped across all other classes.

8. In addition to the L2 and L3 aggregate classes, an individual dropped packet MUST only account against a single error, policy, or no_buffer discard subclass.

9. When there are multiple drop reasons for a packet, the ordering of discard class reporting MUST be defined.

10. If Diffserv [RFC2475] quality of service (~~QOS~~QoS) is not used, no_buffer discards SHOULD be reported as class0.

11. Traffic to the device control plane has its own class, however, traffic from the device control plane SHOULD be accounted for in the same way as other egress traffic.

3.3. Examples

Assuming all the requirements are met, a "good" unicast IPv4 packet received would increment: - interface/ingress/traffic/l3/v4/unicast/packets
- interface/ingress/traffic/l3/v4/unicast/bytes
- interface/ingress/traffic/qos/class_0/packets
- interface/ingress/traffic/qos/class_0/bytes

A received unicast IPv6 packet dropped due to ~~TTL~~ Hop Limit expiry would increment:
- interface/ingress/discards/l3/v6/unicast/packets
- interface/ingress/discards/l3/v6/unicast/bytes
- interface/ingress/discards/l3/rx/ttl_expired/packets

An IPv4 packet dropped on egress due to no buffers would increment: - interface/egress/discards/l3/v4/unicast/packets
- interface/egress/discards/l3/v4/unicast/bytes
- interface/egress/discards/no_buffer/class_0/packets
- interface/egress/discards/no_buffer/class_0/bytes

4. A Possible Signal-Cause-Mitigation Mapping

Example discard signal-to-cause-to-mitigation mappings are shown in the table below:

Discard class	Cause	Discard	Unintended?	Possible actions	Discard rate	duration

Commenté [BMI17]: The MUST assumes that there is discontinuity time in a device or counters recycling back.

Commenté [BMI18]: How to demux IPv4 vs. IPv6?

I would use hoplimit counter for IPv6

| ingress/discards/errors/l2/rx | Upstream device |
>Baseline | O(1min) | Y | Take upstream link or |
		or link errror
		device out-of-service
ingress/discards/errors/l3/rx/ttl_expired	Tracert
<=Baseline		N	no action
ingress/discards/errors/l3/rx/ttl_expired	Convergence
>Baseline	O(1s)	Y	no action
ingress/discards/errors/l3/rx/ttl_expired	Routing loop
>Baseline	O(1min)	Y	Roll-back change
.*/policy/.*	Policy
	N	no action
ingress/discards/errors/l3/no_route	Convergence
>Baseline	O(1s)	Y	no action
ingress/discards/errors/l3/no_route	Config error
>Baseline	O(1min)	Y	Roll-back change
ingress/discards/errors/l3/no_route	Invalid destination
>Baseline	O(10min)	N	Escalate to operator
ingress/discards/errors/local	Device errors
>Baseline	O(1min)	Y	Take device
		out-of-service
egress/discards/no_buffer	Congestion
<=Baseline		N	no action
egress/discards/no_buffer	Congestion
>Baseline	O(1min)	Y	Bring capacity back
		into service or move
		traffic

The 'Baseline' in the 'Discard Rate' column is network dependent.

5. Security Considerations

There are no new security considerations introduced by this document.

6. IANA Considerations

There are no new IANA considerations introduced by this document.

Commenté [BMI19]: I'm not sure yet about the maintenance of the classification model and how to supp future classes/subclasses.

~~7. Terminology~~

~~The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.~~

8. Contributors

Nadav Chachmon
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
United States of America

Email: nchachmo@cisco.com

9. Acknowledgments

The content of this draft has benefitted from feedback from JR Rivers, Ronan Waide, Chris DeBruin, and Marcoz Sanz.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

10.2. Informative References

[RED93] Jacobson, V., "Random Early Detection gateways for Congestion Avoidance", n.d..

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, <https://www.rfc-editor.org/rfc/rfc1157>.

[RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, DOI 10.17487/RFC1213, March 1991, <https://www.rfc-editor.org/rfc/rfc1213>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/rfc/rfc2475>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/rfc/rfc3444>.

[RFC5153] Boschi, E., Mark, L., Quittek, J., Stiemerling, M., and P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines", RFC 5153, DOI 10.17487/RFC5153, April 2008, <https://www.rfc-editor.org/rfc/rfc5153>.

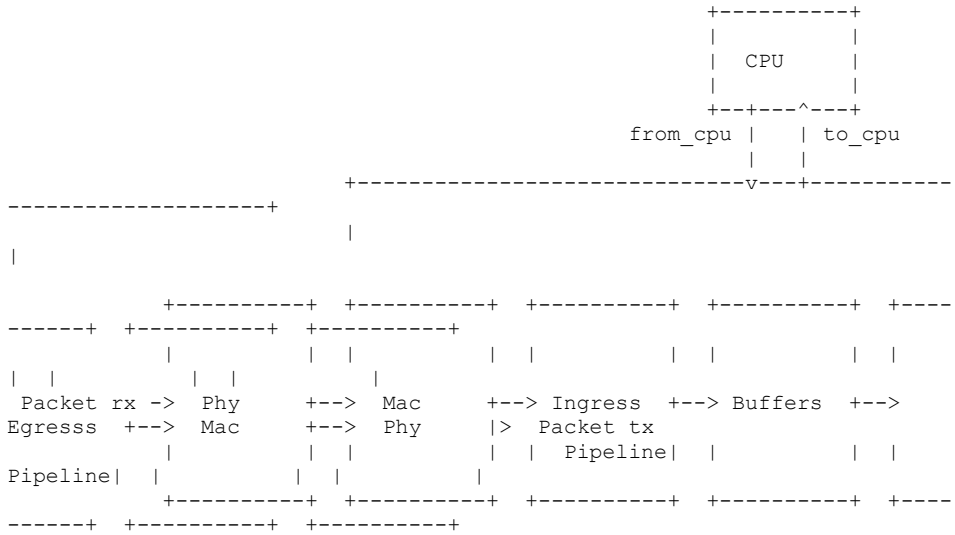
[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC8289] Nichols, K., Jacobson, V., McGregor, A., Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management", RFC 8289, DOI 10.17487/RFC8289, January 2018, <https://www.rfc-editor.org/rfc/rfc8289>.

Appendix A. Where do packets get dropped?

The diagram below is an example of where and why packets may be dropped in a typical single ASIC, shared buffered type device, where packets ingress on the left and egress on the right.



Appendix B. Implementation Experience

This appendix captures the authors' experience gained from implementing and applying this information model across multiple vendors' platforms, as guidance for future implementers.

1. The number and granularity of classes described in Section 3 represent a compromise. It aims to offer sufficient detail to enable appropriate automated actions while avoiding excessive detail which may hinder quick problem identification.

Additionally, it helps constrain the quantity of data produced per interface to manage data volume and device CPU impacts. Although further granularity is possible, the scheme described has generally proven to be sufficient.

2. There are multiple possible ways to define the discard classification tree. For example, we could have used a multi-rooted tree, rooted in each protocol. ~~Instead~~Instead, we opted to define a tree where protocol discards and causal discards are accounted for orthogonally. This decision reduces the number of combinations of classes and has proven sufficient for determining mitigation actions.

3. NoBuffer discards can be realized differently with different memory architectures. Hence, whether a NoBuffer discard is attributed to ingress or egress can differ accordingly. For successful auto-mitigation, discards due to egress interface congestion should be reported on egress, while discards due to device-level congestion (exceeding the device forwarding rate) should be reported on ingress.

4. Platforms often account for the number of packets dropped where the TTL has expired, and the CPU has returned an ICMP Time Exceeded message. There is typically a policer applied to limit the number of packets sent to the CPU, however, which implicitly limits the rate of TTL discards that are processed. One method to account for all packet discards due to TTL exceeded, even those that are dropped by a policer when being forwarded to the CPU, is to use accounting of all ingress packets received with TTL=1.

5. Where no route discards are implemented with a default null route, separate discard accounting is required for any explicit null routes configured, in order to differentiate between interface/ingress/discards/policy/null_route/packets and interface/ingress/discards/errors/no_route/packets.

6. It is useful to account separately for transit packets dropped by transit ACLs or policers, and packets dropped by ACLs or policers which limit the number of packets to the device control plane.

7. It is not possible to identify a configuration error - e.g., when intended discards are unintended - with device packet loss metrics alone. For example, to determine if ACL drops are intended or due to a misconfigured ACL some other method is needed, i.e., with configuration validation before deployment or by detecting a significant change in ACL drops after a change compared to before.

8. Where traffic byte counters need to be 64-bit, packet and discard counters that increase at a lower rate may be encoded in fewer bits, e.g., 48-bit.

9. In cases where the reporting device is the source or destination of a tunnel, the ingress protocol for a packet may differ from the egress protocol; if IPv4 is tunneled over IPv6 for example.

Some implementations may attribute egress discards to the ingress protocol.

10. While the classification tree is seven layers deep, a minimal implementation may only implement the top six layers.

Authors' Addresses

John Evans
Amazon
1 Principal Place, Worship Street
London
EC2A 2FA
United Kingdom
Email: jevanamz@amazon.co.uk

Oleksandr Pylypenko
Amazon
410 Terry Ave N
Seattle, WA 98109
United States of America
Email: opyl@amazon.com

Jeffrey Haas
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: jhaas@juniper.net

Aviran Kadosh
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
United States of America
Email: akadosh@cisco.com