Application Enabled Open Networking (AEON)

Charles Eckel, Cisco Systems, eckelcu@cisco.com

Abstract

Identification and treatment of application flows are important to many application providers and network operators. They often rely on these capabilities to deploy and/or support a wide range of applications. These applications, and the packet flows they generate and consume, may have specific connectivity requirements that can be met if made known to the network. Historically, this functionality has been implemented to the extent possible using heuristics, which inspect and infer flow characteristics. Heuristics may be based on port ranges, network separation (e.g. subnets or VLANS), or Deep Packet Inspection (DPI). But many application flows in current usages, and those corresponding to web based applications in general, are dynamic, adaptive, time-bound, encrypted, peer-to-peer, asymmetric, used on multipurpose devices, and have different priorities depending on direction of flow, user preferences, and other factors. Any combination of these properties renders heuristic based techniques less effective and may result in compromises to application security or user privacy.

Introduction

Networks today, whether public or private, are challenged with demands to support rapidly increasing amounts of traffic. New channels for creating and consuming rich media are deployed at a rapid pace. Pervasive video and access on demand are becoming second nature to consumers. Communication applications make extensive use of rich media, placing unprecedented quality of experience expectation on the underlying network. These trends present challenges for network forecast and planning operations.

Now more so than ever before, identification and treatment of application flows are critical for the successful deployment and operation of a growing number of business and household applications. These applications are based on wide range of signaling protocols and deployed by a diverse set of application providers that is not necessarily affiliated with the network providers across which the applications are used.

Historically, identification of application flows has been accomplished using heuristics, which inspect and infer flow characteristics based on port ranges, network separation, or DPI. Each of these techniques suffers from a set of limitations, particularly in the face of challenges on the network outlined previously.

Heuristic-based approaches may not be efficient and require continuous updates of application signatures. Port based solutions suffer from port overloading and inconsistent port usage. Network separation techniques like IP subnetting are error prone and increase network management complexity. DPI is computationally expensive and becomes a greater challenge with the wider adoption of encrypted signaling and secured media. An additional drawback of DPI is that any insights are not available, or need to be recomputed, at network nodes further down the application flow path.

As the W3C and IETF establish default behaviors that thwart pervasive surveillance, it will be important to provide mechanisms for applications to protect the contents of their flows, yet have the ability to opt-in to information exchanges that provide a more precise allocation of network resources and thus a better user experience.

Typical Workflows

Various heuristic based approaches are used prevalently and successfully for two types of workflows:

1. Provide network operators with visibility into traffic for troubleshooting, capacity planning, accounting and billing, and other off network workflows. Typically done by exporting observed traffic via protocols such as IPFIX [RFC7011] and SNMP [RFC3416].
2. Provide differentiated network services for specific traffic according to network operator defined policies, including traffic classification, policing and shaping (e.g. [RFC2475]), providing admission control (e.g. [RFC6601]), impacting routing, and permitting passage of specific traffic (e.g. firewall functions).

Limitations of Heuristic Based Solutions

These typical work flows, visibility and differentiated network services, are critical in many networks. However, their reliance on inspection and observation limits the ability to enable these workflows more widely. Reasons for this include the following:

- Simple observation based classification based on TCP/UDP port numbers often result in incorrect results due to port overloading (i.e. ports used by applications other than those claiming the port with IANA).
- More and more traffic is encrypted, rendering DPI impossible, inefficient, or much more complex, and sometimes at the expense of privacy or security (e.g. need to share encryption keys with intermediary proxy performing DPI).
- Visibility generally requires inspecting the signaling traffic of applications. This traffic may flow through a different network path than the actual application data traffic. Impacting the traffic behavior is ineffective in those scenarios.
- Extensions to signaling protocols can result in false negatives or false positives during inspection.
- Network services leveraging DPI traffic classification impact the application behavior by impacting its traffic, but they do not provide explicit feedback to the application. This results in a lost opportunity for the application to gain insight and adjust its operation accordingly.

Limitations of Existing Signaling Mechanisms

The IETF has standardized several mechanisms involving explicit signaling between applications and the network that may be used to support visibility and differentiated network services workflows. Unfortunately, none of these has experienced widespread deployment success, nor are they well suited for the applications usages described previously. Existing signaling options include the following:

- RSVP [RFC2205]is the original on-path signaling protocol standardized by the IETF. It is transported out-of-band and could be used to signal information about any transport protocol traffic (it currently supports TCP and UDP). Its original goal was to provide admission control. Its requirement for explicit reservation of resources end to end proved too heavy for most network environments. Its success was further impacted by its reliance on router-alert, which often leads to RSVP packets being filtered by intervening networks, and by its requirement for access to a raw socket, something that is generally not available to applications running in user space. To date, more lightweight signaling workflows utilizing RSVP have not been standardized within the IETF.
- NSIS (next Steps in Signaling) [RFC5978]is the next iteration of RSVP-like signaling defined by the IETF. It focused on the same fundamental workflow as RSVP admission control as its main driver, and because it

did not provide significant enough use-case benefits over RSVP, it has seen even less adoption than RSVP.

- DiffServ style packet marking [RFC4594] can help provide QoS in some environments but DSCP markings are often modified or removed at various points in the network or when crossing network boundaries. There are additional limitations when using DiffServ with real-time communications applications, and the DART working group has been chartered to write a document that explains the limitations that exist with DiffServ when used with RTP in general as well in the specific WebRTC cases.

Basis for a Solution

Not surprisingly, there are several evolving proposals that aim to address the visibility and differentiated network services workflows where existing approaches are not sufficient. Protocol specific extensions are being defined, creating duplicate or inconsistent information models. This results in duplicate work, more operational complexity and an inability to convert information between protocols to leverage the best protocol option for each specific use case.

Rather than encourage independent, protocol specific solutions to this problem, application enabled open networking (AEON) advocates a protocol and application independent information model and individual data models that enable explicit communication between applications and the networks on which they are used.

Key aspects of this communication include the following:

1. Allow applications to explicitly signal their flow characteristics to the network.
2. Provide network nodes visibility to application flow characteristics.
3. Enable network nodes to contribute to application flow descriptions.
4. Allow applications to receive resulting flow descriptions as feedback from the network.
5. Complement existing heuristic based mechanisms.
6. Provide differentiated service for both directions of a flow, including flows that cross administrative boundaries.
7. Provide mechanism to authenticate and authorize endpoints/applications to signal flow characteristics, including 3rd party authentication and authorization for over-the-top (OTT) applications.
8. Provide mechanism for integrity protection and replay protection of messages exchanged between the application and the network.

The information communicated should include an application independent description of the media flows associated with an application, e.g.

- Delay Tolerance
- Loss Tolerance
- Jitter Tolerance
- Bandwidth
- Relative Priority

In this way, the application describes the flows it intends to send and receive. A network node can use this information to utilize its available resources to accommodate the flows, and it can provide feedback to the application informing it of the anticipated handling of its flows. The application can use this information to adjust its operation to better align with expected network conditions, including setting appropriate user expectations, checking availability of alternate network interfaces, etc.

This communication may occur using any of a variety of protocols, including the following:

- STUN [RFC5389] is an on-path, in-band signaling protocol that could be extended to provide signaling to on-path network devices. It provides an easily inspected packet signature, at least for transport protocols such as UDP. Through its extensions TURN [RFC5766] and ICE [RFC5245], it is becoming prevalent in application signaling driven by the initial use-case of providing NAT and firewall traversal capabilities and detecting local and remote candidates for peer-to-peer media sessions. The TRAM working group is chartered to update TURN and STUN to make them more suitable for WebRTC.
- Port Control Protocol (PCP) [RFC6887] provides a mechanism to describe a flow to the network. The primary driver for PCP is creating port mappings on NAT and firewall devices. When doing this, PCP pushes flow information from the host into the network (specifically to the network's NAT or firewall device), and receives information back from the network (from the NAT or firewall device). It is not meant to be used end-to-end but rather independently on one "edge" of a flow. It is therefore an attractive alternative because it allows the introduction of application to network signaling without relying on the remote peer. This is especially useful in multi-domain communications.
- RESTCONF [I-D.ietf-netconf-restconf] is a REST-like protocol that provides a programmatic interface over HTTP for accessing data defined in YANG, using the datastores defined in NETCONF [RFC6241]. It is meant to provide a standard mechanism for web applications to access the configuration data, operational data, data-model specific protocol operations, and notification events within a networking device, in a modular and extensible manner.

Conclusion

Identification and treatment of application flows are critical for the successful deployment and operation of applications. These applications are based on a wide range of signaling protocols and deployed by a diverse set of application providers that is not necessarily affiliated with the network providers across which the applications are used. Reliance of existing solutions on heuristics and inspection limits their effectiveness and/or compromises security and user privacy. AEON offers a path forward, whereby applications explicitly signal their flows to the network and the network provides feedback to applications. This communication provides visibility that enables networks to better accommodate application flows and enables applications to adjust their operation to network conditions.

References

[I-D.ietf-netconf-restconf]	Bierman, A., Bjorklund, M., Watsen, K. and R. Fernando, "RESTCONF Protocol", Internet-Draft draft-ietf-netconf-restconf-00, March 2014.
[I-D.ietf-rtcweb-use-cases-and-requirements]	Holmberg, C., Hakansson, S. and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", Internet-Draft draft-ietf-rtcweb-use-cases-and-requirements-14, February 2014.
[IEEE-802.1Q]	IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", IEEE 802.1Q, 2005.
[RFC2205]	Braden, B., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
[RFC2475]	Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
[RFC3416]	Presuhn, R., "Version 2 of the Protocol Operations for the Simple Network

Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.

[RFC4594] Babiarz, J., Chan, K. and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P. and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5766] Mahy, R., Matthews, P. and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC5978] Manner, J., Bless, R., Loughney, J. and E. Davies, "Using and Extending the NSIS Protocol Family", RFC 5978, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J. and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6601] Ash, G. and D. McDysan, "Generic Connection Admission Control (GCAC) Algorithm Specification for IP/MPLS Networks", RFC 6601, April 2012.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R. and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

[RFC7011] Claise, B., Trammell, B. and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.