Internet Engineering Task Force
INTERNET–DRAFT
Intended status: Informational
Expires: September 2007

P. Sarolahti
Nokia Research Center
S. Floyd
ICIR
M. Kojo
University of Helsinki

5 March 2007

Transport-layer Considerations for Explicit Cross-layer Indications draft-sarolahti-tsvwg-crosslayer-01.ps

Status of this Memo

Abstract

Several types of explicit cross-layer communication schemes have been proposed to enhance the transport protocol performance. However, various challenges with such schemes have been identified, for example concerning the interactions with the middleboxes and tunnels in the network. This document discusses different types of explicit cross-layer notification mechanisms that have been proposed to enhance end-to-end transport performance. We analyze the different mechanisms using a taxonomy based on what kind of network interactions they require, and discuss the benefits and disadvantages different approaches have. The objective is to get a common understanding of the possibilities and challenges with these mechanisms, with pointers to past discussions on this topic, and to describe the possible next steps towards removing barriers from explicit cross-layer communication in future protocols.

Table of Contents

1. Introduction

Recent research argues that the traditional interface between the transport layer and the network layer may not be sufficient for the present day needs [EE06]. For example, the traditional TCP congestion control algorithms are slow to converge to sudden path changes where the throughput and round-trip times may change by orders of magnitude. Therefore, it has been proposed that in addition to the "implicit" observations about the path characteristics, such as measured round-trip times and the available bandwidth probed by usual congestion control mechanisms, enhancing the transport protocols by "explicit" information would be useful.

In the past there have been different proposals on enhancing transport protocol (usually TCP) performance by means of providing explicit information and notifications from different protocol layers above or below transport. The cross-layer notifications can be local notifications from the lower or upper protocol layers of the host device, or it can be explicit communication between the transport peers and the network between them. The mechanisms for cross-layer signaling inside a host implementation are largely dependent on the operating system architecture, and therefore not of interest for the IETF. However, explicit signaling between the network and the end-hosts involves several considerations on the network behavior that we try to capture in this document.

Cross-layer signaling could be used, for example, for delivering hints to a transport sender about the characteristics of the network path, to allow the sender to adjust its sending rate more efficiently than what would be possible using the traditional TCP probing mechanisms. While designing the possible uses of such signaling, a careful consideration needs to be made of what can be done within the limits of the congestion control principles [RFC2914, RFC2581], without endangering the network stability and fairness towards other flows. Often this determines whether end-hosts can negotiate directly without network support, or whether some or all of the routers along the network path need to support the signaling mechanism.

One of the guiding architectural principles of the Internet has been that the network should be stateless, with the transmission state and intelligence residing at the end hosts [Cla88]. Although today this principle has been ignored more than once by the different types of Network Address Translators (NATs) and stateful firewalls, it is an important consideration when evaluating the cross-layer notification methods. While many of the notification mechanisms discussed in this document conform to this principle, some mechanisms do require some additional state in the network. Adding new bits of state in the network is not necessarily a bad thing, but the design should be such that loss of the state would not cause serious fate-sharing problems that might prevent the network's packet forwarding function from working.

While the benefits of applying cross-layer notifications to improve the transport protocol performance has been evaluated in number of studies [SAF06, SEE+06, KSE+04], several problems have also been identified with regard to conformance to congestion control principles, interactions with middleboxes in the network, or interoperation with IP tunnels and lower layer bridges. An important design principle would also be to maintain the layer-abstraction that isolates the transport layer from any particular link technology, which is forgotten in some proposals on enhancing the transport performance by cross-layer interactions. This document casts an overview on different types of explicit cross-layer notifications, and discusses their possibilities and challenges.

The objective of the final document is to build a common understanding of the issues related to explicit communication between the transport layer and the network. This is intended to help the various proposals to enhance protocol performance using cross-layer information. Such enhancements have been discussed both in the TSV and INT areas. Because many IETF participants are focused on following only a selection of areas, it is possible that work conducted in one IETF area does not get a thorough review from participants focusing in other areas (before the IESG review). Because the cross-layer enhancements potentially touch different IETF areas and may be progressed in different IETF working groups, it could be helpful to have transport layer guidelines that would hopefully be useful in the design process of possible new cross-layer notification schemes. An additional goal for this document is to propose possible next steps towards solving the identified challenges related to explicit cross-layer communication.

1.1. Conventions and Terminology

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].

2. Definitions and Scope

This section defines some terms and concepts used in the rest of this document.

2.1. Definitions

router: Network node that forwards the IP packet to the next link towards its destination based on the information in the IP header. A router may modify the contents of the IP header, for example by decrementing the IPv4 TTL or IPv6 hop count fields.

middlebox: A network device along the transport path that performs operations that are beyond the normal IP packet forwarding done by the routers. Often this involves investigating the transport protocol header of the packet. Firewalls and Network Address Translators (NATs) are the most common types of middlebox.

bridge: A network node that forwards the data frames based on layer 2 information. A bridge does not process the IP header.

on-path: A message that has the same sender and receiver as the normal protrocol traffic, and that follows exactly the same sequence of routers as the normal traffic, is called an on-path message.

off-path: A message that has a different sender or receiver, or that is forwarded via a different sequence of routers than the normal protocol traffic, is called an off-path message.

in-band: A message carried in the same IP packet with the normal protocol traffic is called an in-band message. Implicitly, an in-band message is also an on-path message.

out-of-band: A message that is not carried in a same packet with the normal protocol traffic, but as a separate packet, is called out-of-band message.

notification: Although notification is a rather generic term, in this document notification is a message that carries explicit cross-layer information.

2.2. Roles of different protocol layers

It is difficult to find a course book on computer networking that would not begin with a description of the different protocol layers, usually according to the ISO reference model. For example, [Hal96, Figure 1.11] summarizes the different protocol layers as follows:

* Physical layer (1): Mechanical and electrical network interface definitions.

* Link layer (2): Data link control (framing, data transparency, error control).

* Network layer (3): Network routing, addressing, call set-up, and clearing.

* Transport layer (4): End-to-end message transfer (connection management, error control, fragmentation, flow control).

* Session layer (5): Dialog and synchronization control for application entities.

* Presentation layer (6): Transfer syntax negotiation, data representation transformations.

* Application layer (7): File transfer, access and management, document and message interchange, job transfer and manipulation.

  Although ISO reference model layering is not explicitly visible in many of the IETF protocols, and some protocols might do tasks of more than one layer, it is possible to find places of different IETF protocols in this model. Usually the proposed cross-layer enhancements concern interactions between the link layer, network layer, and transport layer.

2.3. Scope

This document discusses the cross-layer mechanisms that take place between the end-hosts and the network. Local triggers inside a protocol stack are out of the scope of the IETF, and this document does not discuss such schemes in detail. We focus on cross-layer notifications that are used by the transport layer to enhance the end-to-end communication. Therefore, for example the cross-layer information used for routing or packet forwarding inside specific network clouds is out-of-scope. We give somewhat less attention to off-path notification mechanisms, to make the discussion more focused. Also, we focus on unicast traffic and do not discuss multi-cast communication.

3. Possible Benefits of Explicit Signaling

The past proposals to add new explicit signaling mechanisms have been motivated in the following ways.

* Mobility: One of the key design principles of the IP mobility protocols has been to isolate the mobility from the upper protocol layers. While this makes sense architecturally, it has been observed that hiding the mobility event from upper layer protocols can lead to suboptimal performance. A significant amount of research has been conducted to investigate and optimize the performance of the upper layers after a mobile hand-off occurs (e.g., [SKDK06, SEE+06, DK06]). It has been observed that with better awareness of mobility, benefits on transport protocol performance can be achieved.

  There is ongoing work in the IEEE 802.21 group to develop Media Independent Handover services [IEEE21]. As part of this effort, the IEEE specifies link-layer triggers to optimize the hand-off performance of a mobile host. The IETF is also doing the related work in the MIPSHOP working group, and in the MOBOPTS IRTF group [TGM+06]. While this work is targeted to optimize mobility, the information from the specified link-layer triggers could also be useful to transport protocols.

* High delay-bandwidth networks: TCP slow-start is known to be inefficient when used over a very high-speed network path, or over a network path with large propagation delay. The TCP startup performance could be improved with explicit information about the current available capacity of the connection path [SAF06, KHR02].

* Non-congestion losses: a traditional research topic on TCP performance has been detection and response to packet reordering and to packet losses that are not caused by congestion. Possible performance benefits for being able to detect non-congestion losses have been evaluated in a number of documents [KSE+04].

* Quality of Service: In addition to DiffServ and RSVP QoS signaling, there have been other proposals for smaller enhancements regarding real-time transmission of packets. For example, Packet Lifetime Discard [GL03] assigns a lifetime for IP packets to allow routers to drop packets that have exceeded their lifetime, thereby saving network resources. An earlier work proposed adjusting the link-layer reliability mode in GSM networks based on the transport protocol in use [LR99], to allow more timely handling of UDP packets.

4. Classification of Explicit Notification Mechanisms

We classify the explicit notification mechanisms based on their different characteristics. First, the notifications can be transmitted in-band in the same packets with data, or out-of-band as separate packets from the data. Second, the notifications may need participation from some or all routers along the connection path, or they can be processed at the end-hosts only. Third, notifications may be transfered on the data path or off-path, following a different network route than the data traffic. Finally, notifications can be top-down (originating from the transport layer to lower layers), bottom-up (originating from lower layers and carrying information to the transport layer), and part of an extended conversation with a mixture of the two.

4.1. In-band and out-of-band notifications

In-band signaling goes with the normal protocol traffic, that is, with a protocol control message or data. The benefit is that in-band messages are known to share the same path as the normal protocol traffic, and generally use less header overhead than a separate message. In addition, with in-band messages the sender is often able to learn about a loss event via the transport protocol’s normal acknowledgment mechanisms. The disadvantage of in-band signaling is that if some routers or middleboxes drop a packet because of unknown protocol information (for example, a notification transfered in an IP option), the accompanying data also gets lost, causing a performance penalty for the data transfer. In-band notifications can also cause additional header and processing overhead for the data packets, especially if routers need to process additional notification information in the packets.

Out-of-band mechanisms require a dedicated protocol for signaling. While the out-of-band mechanisms save the normal protocol traffic from additional overhead, the transmission of separate messages may be prevented by middleboxes on the connection path. If the message is lost in the network for some reason, there may not be any way for either end of the connection path to know about it. If the out-of-band notification needs to be matched with a particular flow, the notification message would need to include the IP source and destination address, transport protocol, and source and destination transport protocol ports. Getting and using this information may not be possible in all cases, for example when the transport protocol header is encrypted by IPsec ESP [RFC2401]. If the notification needs to be in synchrony with the data flow, a separate out-of-band message may be problematic, because the message may be lost or delayed relative to the data traffic.

4.2. Involvement of routers on the path

The possible notification mechanisms can differ in how much they need assistance from the routers on the connection path. Some notification mechanisms can be useful even if they are supported by only a few of the routers on the path, whereas other notification mechanisms require that every router on the path supports the notification scheme. To help further discussion, we assign short names for each category.

* "NoRouters": Mechanisms that do not require support from routers on the connection path are easiest to deploy. For example, such an in-band notification scheme could use a TCP option to carry the required information. Because this class of mechanisms can use the normal unmodified IP header, there usually should not be additional problems with legacy routers, tunnels, or middleboxes processing these notifications. An additional benefit is that IPsec can be used to protect the notification content.

* "SomeRouters": If some of the routers are intended to process the notification, it needs to be placed in the packet so that the routers are able to see it. For example, the notification needs to be an IP option (or IPv6 hop-by-hop option), or there needs to a Router Alert option [RFC2113, RFC2711] that signals the router to process the packet more thoroughly. There may be some deployment problems with this class of mechanisms. For example, some misbehaving routers or middleboxes in the network are known to drop IP packets based in the ECN bits in the TCP header. It is also known that many routers or middleboxes drop packets containing unknown IP options [MAF04].

* "AllRouters": In some cases all routers on the connection path need to process a notification. This is a hard requirement, because the deployment of such mechanisms can take time. The above mentioned deployment problems with misbehaving routers and middleboxes applies also to this class of mechanisms. Requiring all routers to process a notification is challenging also because the packet forwarding logic in routers is often highly optimized and the fast path algorithms can not be expected to conduct very complicated additional processing on the IP packets. Verifying that all routers have indeed processed the notification

can sometimes be difficult. A common verification mechanism is to use a separate TTL counter that is adjusted at the same time with IP TTL or hop count fields. However, different types of IP tunnels can hide the notification data inside the outer IP header. In this case, if the tunnel encrypts the data inside the outer header, there is no way for routers to operate on the notification data. In some cases the TTL-based verification mechanisms may not be able to detect such tunnel. It is also possible that the TTL fields are manipulated by a malicious node on the connection path that has enough information to make educated guesses about the expected TTL values.

4.3. On-path and off-path mechanisms

As mentioned in the beginning, we focus on on-path mechanisms, but there is an important and common application of off-path signaling that deserves to be mentioned. Most ICMP messages [RFC792, RFC2463] are off-path notifications, because they are often sent by an intermediate router on the connection path in the reverse direction, towards the sender of the packet that triggered the ICMP condition; an example is the ICMP "host unreachable" error. Some ICMP messages are triggered by the connection receiver, such as the "Protocol unreachable" error. We consider such ICMP notification also as an off-path notification, because it traverses the reverse direction, and in case of asymmetric routing the ICMP message can traverse a different set of routers than the original message.

One of the problems with off-path mechanisms is that it may be difficult to authenticate an off-path notification that originates from the network. In many ICMP messages the beginning of the IP payload, i.e., the transport protocol header, is copied to the ICMP message. This information can be used to identify the flow that has triggered the ICMP message. There is a work-in-progress Internet-Draft on analyzing the security of ICMP messages [G06].

4.4. Top-down, bottom-up and mixed notifications

Some cross-layer notifications involve extended conversations between the transport layer and lower layers. For example, the ECN field in the IP header is used to carry ECN-capable information from the transport protocol to routers, and to carry Congestion Experienced information from routers back to the transport protocol.

However, other proposed cross-layer notifications are more clearly unidirectional, carrying information only from the transport layer to lower layers, or vice versa. As an example of a proposed top-down notification, a transport protocol such as UDP-Lite would communicate to lower layers about the desired checksum coverage for a packet [RFC3828]. An example of a proposed bottom-up notification would be a link-layer trigger designed to optimize the hand-off performance of a mobile host. Unidirectional top-down or bottom-up notifications are best designed to serve only as hints, to be used by the recipient only as is deemed appropriate.

5. Current, Proposed, and Past Explicit Cross-layer Mechanisms

In this section we discuss some of the explicit cross-layer signaling mechanisms that have been proposed in the past.

5.1. Determining the packet size

TCP uses an option to negotiate the Maximum Segment Size (MSS) during the TCP connection establishment, where each TCP end point may send a TCP MSS option to the other end point indicating the MSS it is able to receive. In general, the MSS information is local link information, the link MTU size, learned via the IP layer and translated to the transport layer MSS. This mechanism would be classified as an in-band "NoRouters" notification.

In order to determine the maximum segment size allowed on the whole connection path between the sender and receiver, Path MTU discovery needs to be applied. The traditional Path MTU discovery is not strictly an explicit on-path mechanism, because it is based on the use of an ICMP "packet too big" error message that a router sends when an incoming packet is too large to be sent on the router's next hop. However, an in-band IP option has been proposed as an alternative Path MTU mechanism [Wel03]. All routers would be required to process the IP option, so this would be a rather challenging scheme to deploy.

5.2. Congestion and rate control

Because TCP congestion control adjustments are a popular application for explicit notifications, some general guidelines on using the above categories are required:

* A sender MAY reduce the sending rate in response to a "NoRouters" or "SomeRouters" notification.

* In order to increase the sending rate more than would be allowed by the normal congestion control principles, a sender MUST use an "AllRouters" notification to verify that the rate increase does not cause congestion in the network.

We call a mechanism that does not conform to these principles an "invalid mechanism". It can be debated whether "AllRouters" mechanisms are truly valid because of problems the "AllRouters" mechanisms have with IP tunnels, that may cause false positives with the "AllRouters" mechanisms. We discuss this issue more thoroughly in Section 7.2.

Some proposals use information about the change of last-hop link characteristics, for example in adjusting the congestion control state [DK06, SEE+06]. This can be an attractive application for mobile terminals that are able to detect mobility, and the change of the wireless last-hop link, and make appropriate changes in the congestion control state under the assumption that in many cases the wireless last-hop link is the bottleneck on the connection path. In some cases the indication of the last-hop link change can be sufficient information for reducing the transmission rate, or restarting the TCP slow-start to evaluate the capacity of new path. However, following the principle given above, such a link indication MUST NOT be used alone to rapidly increase the data transmission rate. The only way to increase the transmission rate is through the normal congestion control mechanisms, or by using an "AllRouters" notification mechanism.

The above principle should also apply to non-congestion-controlled protocols, for example transmission of audio/video streams over RTP and UDP. For example, using an explicit notification about changing from a low-bandwidth first-hop link to a high-bandwidth first-hop link as a trigger to suddenly increase the transmission rate is against the congestion control fairness principles [RFC2914] and therefore would be an invalid mechanism.

A notification has been suggested to allow faster adaptation to changes in the end-to-end path properties. TCP's response to connectivity change indications such as mobility have been discussed in an Internet-Draft [SEE+06]. The draft describes a "connectivity-change indication" TCP option, and the response to a connectivity-change event, when detected either from the TCP option, or from the local stack. The TCP option could be used by a mobile host to indicate to the other end that it has moved and path characteristics may have changed. Depending on the current state of a connection, a host receiving a connectivity-change indication may decide to re-evaluate congestion control parameters of a path and/or make a quick retransmission to resume data transmission earlier after a temporary connectivity disruption instead of waiting for the retransmission timer to expire again. This is an in-band "NoRouters" notification.

Explicit Congestion Notification (ECN) [RFC3168] uses a two-bit ECN field in the IP header to allow routers to indicate congestion in the network before they have to start dropping packets due to buffer overflow. ECN can be useful even if only a subset of routers implement it on the connection path. There were initial deployment problems with ECN because some routers in the network dropped packets with a non-zero ECN field in the TCP header, but we believe that today most of these routers have been fixed. ECN is an in-band "SomeRouters" mechanism.

The use of the ECN field is taken further in an alternative protocol to use the field, called Re-ECN [BJSK06]. The protocol aims "to provide sufficient information in each IP datagram to be able to hold senders and whole networks accountable for the congestion they cause downstream, before they cause it."

In Quick-Start [RFC4782], the sender uses an IP option to request permission from routers to send at a higher rate than the normal congestion control would allow. [RFC4782] specifies the use of Quick-Start for TCP and discusses the challenges such a mechanism needs to address. Quick-Start router algorithms and their configuration are analyzed further in [SAF06], and [SKDK06] gives an initial analysis of Quick-Start in wireless environments with vertical hand-offs between different wireless link technologies. Quick-Start is an in-band "AllRouters" mechanism.

Variable-structure congestion Control Protocol (VCP) is another proposed congestion control proposal using explicit feedback from routers. VCP leverages the ECN field to let routers indicate their load information [XSSK05]. Based on the VCP bits, a TCP sender could apply either Multiplicative Increase, Additive Increase, or Multiplicative Decrease of the congestion window. VCP is an in-band mechanism, and it is intended to be a "AllRouters" mechanism, but it does not provide a mechanism for checking that all routers have understood and processed the notification. It is possible than VCP allows Multiplicative Increase even if there are fairly loaded routers on the connection path that do not support the mechanism. Therefore VCP is an invalid mechanism to be deployed in the Internet.

Explicit Control Protocol (XCP) [KHR02, FPK06] is a proposal for a full-fledged congestion control protocol involving the interaction of routers and the end-hosts. Although XCP can be considered to be more than just a cross-layer signaling mechanism, it also needs to consider the above-mentioned challenges. XCP uses a separate congestion header between IP and the transport protocols, i.e., it is an in-band protocol. XCP is an "AllRouters" scheme, but it is not currently specified how it is checked that all routers have processed the congestion control header.

Different forms of in-band signaling have also been proposed for dealing with corruption-based packet loss in wireless and satellite networks [KSE+04]. The paper on Explicit Transport Error Notification (ETEN) gives a taxonomy of different notification types, depending on the granularity of the notification, the direction of notification, the location of notification, and so on. The efficiency of cumulative error notification is investigated by simulation experiments. However, no specific packet format is proposed in the paper.

5.3. Quality of Service

The Resource ReSerVation Protocol (RSVP) uses separate out-of-band messages on top of IPv4 or IPv6 to make Quality-of-Service signaling [RFC2205]. The data sender sends a RSVP "Path" message to the data receiver that includes a Router Alert IP option telling the routers on the path to investigate the RSVP message contents closer. Each router adds its IP address to the message to enable routing of the Reservation (Resv) messages sent in the reverse direction to visit exactly the same routers on the reverse path to the data sender. The Resv message does not use the Router Alert option, but is rather explicitly routed on a hop-by-hop basis between the network routers using the state established earlier. In addition to the Path and Resv messages, RSVP has a few other message types delivered on a hop-by-hop basis. RSVP is an out-of-band "AllRouters" mechanism. We also call it an on-path mechanism because it takes measures to ensure that the resource reservation signaling follows the forward path from sender to receiver.

Recently the IETF has specified a NSIS (Next Steps in Signaling) framework to handle signaling in the Internet. The Generic Internet Signaling Transport (GIST) protocol has been specified to transport the application-specific signaling messages over the Internet [SH06]. GIST messages are transfered using TCP or UDP as the transport protocol, depending on whether a reliable connection-oriented service or a connectionless service is desired. The use of SCTP to carry GIST messages is also under investigation. GIST has some common characteristics with RSVP: it uses a Router Alert option to wake up the GIST-aware routers along the path, and for further signaling, explicit hop-by-hop routing can be applied using the state established at routers. Like RSVP, also GIST is an out-of-band "AllRouters" scheme.

6. Past IETF Activities

This section discusses the past history of the IETF in considering link-layer triggers and other types of cross-layer communication.

The IAB has an internet-draft on "Architectural Implications of Link Indications" that summarizes current proposals, describes the architectural issues and provides examples of appropriate and inappropriate uses of link layer indications [Abo07]. The document also gives a history of the integration of link indications within the Internet architecture.

The "Performance Implications of Link Characteristics (pilc)" working group produced seven RFCs concerning different types of links and their effects on transport protocols. The PILC working group did not explicitly consider cross-layer interactions; however, the Performance Enhancing Proxies document [RFC3135] gives guidelines for designing proxies that could also be useful considerations for network devices with cross-layer functionality.

The Triggers for Transport BOF in November 2002 [TrigTranBof] discussed triggers such as "Link Up", "Link Down", and "Packet Discarded". The necessity of a focused and narrow problem statement was discussed, with a need to define the semantics and uses of triggers in an exact way. It was questioned whether different wireless link technologies would be able to reliably produce the required information for the trigger, and what kind of responses would be appropriate at the transport protocol. The consensus was that the "Link Up" trigger might be viable, but that a "Link Down" trigger would be more difficult to be implement in a way that would be useful to the transport protocol. The BOF did not result in the creation of a working group.

The Transport Service at the Intermediary BOF (intersec) [IntersecBof] in March 2003 proposed to work on an architecture that helps performance enhancing middleboxes interoperate better with end-to-end transport protocols, especially with end-to-end security. No working group was established.

BOFs on "Access Link Intermediaries Assisting Services (alias)" [AliasBof1, AliasBof2] were held in two consecutive IETF meetings in 2003, continuing from the trigtran and intersec BOFs. ALIAS extended the discussion to middle-boxes that explicitly signal their existence and capabilities to the transport end-points (and vice-versa). ALIAS included an extensive discussion of security issues, along with a discussion of whether the possible benefits of such intermediaries would be clear enough to make the work worthwhile. No working group was created.

Recently there was a BOF proposal for IETF-66 in July 2006 called "Transport-Enhancing Refinements to the Network Layer Interface (ternli)". This didn't get as far as the above mentioned related BOFs, because the BOF was canceled before the IETF meeting. Instead, a group of people were gathered in an ad-hoc meeting in Montreal discussing the problem space. TERNLI was motivated in part by the research conducted in the MOBOPTS IRTF group about the effects of mobility to transport protocols. While there was an agreement that an explicit signaling mechanism between the transport and network layers should not be limited to mobility, there was a discussion of how the responsibilities should be divided between the Transport and Internet Areas. It was discussed that the work in these two areas is rather disconnected, and it is not always known what related work is being done in the other area. It was agreed that it is worthwhile to continue the discussion on the related research at least informally on a mailing list established under the IETF servers. The Jabber log of the meeting can be found at [http://www.ietf.org/meetings/ietf-logs/tsvwg/2006-07-11.html].

7. Challenges with Explicit Cross-layer Mechanisms

Today the Internet contains a wide variety of different types of middleboxes, tunnels, and advanced packet handling technologies that could cause problems for protocols that assume a simple architecture of interconnected routers with simple packet forwarding algorithms. In addition, the layer-two technologies have become more complex and difficult to model and understand correctly. In this section we list some common challenges to cross-layer mechanisms.

7.1. Security Issues

A cross-layer signaling protocol needs protective measures that are strong enough to make attacks on the protocol difficult and reasonably unprofitable. At the same time, if an otherwise light-weight protocol has heavy-weight security mechanisms, the cost of the security procedures may outweigh the possible benefits of the protocol. It may be possible also to mitigate the potential attacks from misleading hints by designing robust response mechanisms, and considering the offered data as advisory information, while still monitoring that other sources do not provide conflicting information [EE06]. For example, if the sender has increased the transmission rate based on a recent notification, followed by an increased number of congestion-based packet losses, there is a clear conflict in the received information.

For in-band mechanisms that use reserved header bits or IP options, the receiver of the packet can be expected to check that the IP addresses and transport ports match the existing connection, and that the sequence numbers in the packet belong to the currently valid window. Therefore, blind attacks generated outside the packet transmission path have a reasonably low probability of succeeding. For example, most TCP connections survive comfortably in the Internet, although security of the basic TCP has been discovered to be insufficient in certain mission-critical long-term connections [SD06,Tou06]. However, an attacker on a connection path that is able to read the transport and IP headers has a good chance of causing harm to a connection, particularly if the packet contains additional explicit information about the connection, for example in an IP option. IPsec can protect the transport header, but does not protect a mutable IP option that can be modified by routers along the path.

Out-of-band messages do not necessarily include the additional context from the transport protocol, so they can be an easier target for blind attackers. If a transport protocol context exists, for example when the message is triggered by a data packet, the sender of the out-of-band signaling message can include the transport header from a recent data packet with the message to authorize the message based on the "proof" that the message has come from the right source. In principle it cannot be assured that an out-of-band message uses the same path as the data traffic, although it can be assumed to be a common case.

For off-path signaling, for example sent by an intermediate router, including transport protocol context is not necessarily possible when IPsec is used to encrypt the data traffic. To more securely authenticate the sender of a signaling message a more elaborate security framework is needed. It is possible that the complexity of such a security framework causes the costs of the mechanism to defeat the possible benefits.

The routers on the connection path can also try to cheat a cross-layer signaling mechanism. A first-hop router that is located in the same administrative domain with the transport end-host may have an incentive to game the protocol to the end-host's benefit. For example, in the case of Explicit Congestion Notification, a router could try to erase the Congestion Experienced bit on the packet, or a Quick-Start-aware router could try to game a better transmission rate for the transport sender. ECN and Quick-Start both use random content in the header fields called Nonces to make it more difficult for routers and receivers to misuse the protocol. Nonces usually do not provide full protection against misuse, but rather make cheating difficult enough to be unprofitable.

7.2. IP Tunnels

IP tunnels are a challenge for an explicit cross-layer notification protocol that requires participation of the routers, because the tunnel isolates the original IP header inside an outer header. A tunnel protocol could copy the important cross-layer notification data to the outer header at the tunnel ingress so that the routers along the tunnel path can process the information, and then at the tunnel egress copy the possibly changed cross-layer data back to the inner header. For IPsec tunnels there is a special consideration whether exposing the cross-layer data in the outer header is a violation of the security policy. It is possible that some additional cross-layer information on the outer header makes it possible for an intruder to make additional conclusions about the nature of the data that is being transfered inside the IPsec tunnel.

Because the interaction of congestion control and mobility has been one of the key motivations for advanced cross-layer interactions, it is worth noting that one of the most common mobility mechanisms, Mobile IPv4, is based on the use of IP tunneling [RFC3344]. When a mobile host is not at its home location, the Mobile IPv4 home agent receives the packets on behalf of the mobile host, and forwards them to the care-of-address of the mobile host in an IP tunnel. There can also be deployments with several layers of tunneling, for example when IPsec is used together with Mobile IPv4.

IP tunnels are a particular challenge for "AllRouters" mechanisms, because currently there is no known guaranteed way to check that an "AllRouters" notification has indeed been processed by all routers when there is an IP tunnel on the connection path. The Quick-Start specification includes a thorough discussion of problems with IP tunnels [RFC4782]. The key points of that discussion are summarized below.

As described in Section 4.2, a typical way for an "AllRouters" mechanism to check that all of the routers have processed the notification mechanism is to use a special TTL or hop-count field with the notification data. Assuming that all routers decrease the IP TTL field as specified, the difference between the IP TTL and the special TTL field should tell if all routers have processed the notification. If the difference does not match, the end-host knows that there were routers along the path that did not support the notification. However, a problem arises because some tunnels do not necessarily decrease the IP TTL at the tunnel ingress. Therefore the presence of the tunnel and all the routers along the tunnel path may go undetected. This is harmful for the cross-layer notification mechanism that may take actions based on the false assumption that all routers processed the notification.

The Quick-Start specification defines two main categorizes for tunnels: "simple tunnels" simply discard the outer header at the tunnel egress, and "non-simple tunnels" that save and use information from the outer header before discarding it. The specification further divides tunnels into (i) tunnels that support Quick-Start, (ii) tunnels that do not support Quick-Start, but are compatible with Quick-Start, and (iii) tunnels that are not compatible with Quick-Start. A tunnel that supports Quick-Start processes the IP TTL and the special TTL fields appropriately and copies the Quick-Start Request to the outer header. A tunnel that does not support Quick-Start does not copy the Quick-Start Request to outer header, but decreases the IP TTL appropriately so that the end-hosts are able to detect that the whole network path did not support Quick-Start. A tunnel that is not compatible may allow false positives, i.e., false approvals of Quick-Start request in situations where all routers did not process the Quick-Start Request. Because an approved Quick-Start Request allows the sender to transmit at a higher rate than the congestion control rules would usually allow, in the worst case this could cause severe congestion in the network. Although the above classification was given in the context of Quick-Start, the same principles hold in general for an "AllRouters" cross-layer notification mechanism.

Multiprotocol Label Switching can be considered a special case of an IP tunnel, where the IP header can be encapsulated in a small MPLS shim header [RFC3031]. When a packet is transmitted through an MPLS region, the IP header is not processed, but the MPLS specification strongly recommends that the IP TTL field is decremented appropriately at the edge of an MPLS region according to the number of hops is traversed inside the MPLS region. If this recommendation is followed, the above-described problem of false positives due to unadjusted IP TTL cannot occur. However, because it seems unlikely that a cross-layer notification mechanism is supported by MPLS, the "AllRouters" schemes are not likely to work over MPLS regions, depending on the purpose of the cross-layer mechanism.

7.3. Non-conformant routers and middleboxes

[MAF04] observes that for 70% of the destinations tested, TCP SYN packets with unknown IP options were either lost in the network or ignored by the receiving web server. ([MAF04] was not able to determine further why these connections failed when unknown IP options were added to the TCP SYN packets.) The presence of routers or middleboxes that drop packets containing unknown IP options would be a major obstacle to any cross-layer mechanisms that depended on the use of IP options. With in-band mechanisms this would also prevent delivery of the data in the packets, while with out-of-band mechanisms the data transfer would not be directly affected. This is particularly a problem in "SomeRouters" and "AllRouters" schemes, that typically need to modify the IP header.

Employing the Destination Options or Hop-by-hop Options header in IPv6 would avoid this problem. The IPv6 Destination Options header is not subject to intermediary router inspection and would be suitable in delivering signaling information when in-band signaling is used without network involvement. The Hop-by-hop Options header with IPv6 can be used when in-band signaling with support from some routers is needed. The two highest-order bits of the Option Type specifies the action that must be taken if the processing IPv6 node does not recognize the option type, including the possibility to skip over the option.

Traffic normalizers are one type of middleboxes that can be used together with the Intrusion Detection Systems [HKP01]. Because traffic normalizers can modify the contents of an IP header, particularly the IP TTL field, they may interfere with the operation of "AllRouters" mechanisms that typically use the IP TTL to check that all routers have processed the notification. In the worst case such traffic normalizers might result in false positives by causing the IP TTL and special TTL to match even if some routers did not process the notification.

7.4. Processing efficiency

Packets with IP options are assumed to take the slow-path processing path in most routers, as opposed to the optimized fast-path. If the use of IP options or other mechanisms requiring router attention gained in popularity, the impact on the processing efficiency of routers would have to be considered. This problem concerns the "SomeRouters" and "AllRouters" mechanisms. In the Quick-Start proposal, it is assumed that Quick-Start-capable routers would rate-limit the number of Quick-Start requests that are processed, to preserve router efficiency and to protect against possible attacks on the routers themselves.

8. Proposals for Future Actions

We have described different possibilities for utilizing cross-layer indications on transport layer, as well as several challenges there might be in deployment and use of such mechanisms, depending on the level of network support a mechanism requires. The "AllRouters" notifications are the most challenging class of cross-layer mechanisms, because they not only require support from every router along the connection path, but also need a reliable mechanism to verify that each router has indeed processed the notification.

If there is interest in developing cross-layer indications further to improve transport protocol performance, it would be useful to solve the problems below, depending on the required level of network support.

* "AllRouters" notifications: There should be a common, well-specified mechanism to ensure that all routers have indeed processed an explicit notification that is required to be processed by every router, so that false positives would not be possible. To help solving this problem, there would need to be a common, well-specified way for tunnel ingress and egress nodes to process explicit indications that require some level of support from routers along the path. A possible approach could be to aim for a common framework for transmitting light-weight explicit notifications.

* "SomeRouters" and "AllRouters" notifications: There should be a way to discourage routers and middleboxes from dropping packets with unknown IP option header content. These nodes should rather forward the packet without processing the

unsupported option. As the first step, there should be a better understanding of which nodes drop the unknown options, and what is the reason for dropping the packet.

* It would be useful for a designer of an cross-layer mechanism to get input from the router designers to better understand the performance limitations of a modern router, to help in designing realistic cross-layer schemes. The router requirements can vary much depending on the exact usage of the router: a local WLAN access point may be able to employ more complex algorithms than a high-performance backbone router.

The above listed challenges may be technically difficult to solve in the current Internet. However, the above discussion hopefully sheds some light on the amount of work required to design a cross-layer mechanism that is usable in the Internet.

A. List of Changes

Changes from draft-sarolahti-tsvwg-crosslayer-00:

* Added a paragraph about MSS and its relation to link MTU.

* Added a paragraph about possibilities of IPv6 to Section 6.3.

* Description of terminology and the scope of this document added, from the proposal from Scott Brim. Also the document title and introduction were updated slightly.

* Re-organized the taxonomy and description of different proposed schemes, after proposal from Gregory Woodhouse.

* Some updates to introduction and security issues referring to a related paper on rethinking the transport layer interfaces [EE06].

* Changed the document title

* Added more discussion on IP tunnels and MPLS (from a recommendation by Wesley Eddy)

* Moved discussion about path vs. link indications to the congestion control subsection.

* Added a paragraph about traffic normalizers to middleboxes section.

* Added a paragraph about Mobile IPv4 to the IP tunnels section, from suggestion by Wesley Eddy.

* Added text to the "Proposals for Future Actions" section at the end of the document

* Added a short paragraph about IEEE 802.21 and MIPSHOP & MOBOPTS work at the

IETF, from proposal of Qiaobing Xie.

* Added a section on top-down and bottom-up indications.

* Modified the Connectivity-change option description based on the feedback from Simon Schuetz.

Normative References

[RFC793] J. Postel. Transmission Control Protocol. RFC 793, September 1981.

[RFC2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. BCP 14, RFC 2119, March 1997.

[RFC2460] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[RFC2581] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581. April 1999.

[RFC2914] S. Floyd. Congestion Control Principles. RFC 1914, September 2000.

[RFC3168] K.K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, Proposed Standard, September 2001.

Informative References

[AliasBof1] Access Link Intermediaries Assisting Services (alias) Bof minutes from IETF-57, Vienna, Austria, July 2003. Available at http://www3.ietf.org/proceedings/03jul/250.htm

[AliasBof2] Access Link Intermediaries Assisting Services (alias) Bof minutes from IETF-58, Minneapolis, MN, USA, November 2003. Available at http://www3.ietf.org/proceedings/03nov/248.htm

[RFC792] J. Postel. Internet Control Message Protocol. RFC 792, September 1981.

[RFC1191] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, November 1990.

[RFC2113] D. Katz. IP Router Alert Option. RFC 2113, February 1997.

[RFC2205] R. Braden (ed.). Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. RFC 2205, September 1997.

[RFC2207] R. Berger and T. O'Malley. RSVP Extensions for IPSEC Data Flows. RFC 2207, September 1997.

[RFC2401] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, November 1998.

[RFC2463] A. Conta and S. Deering. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 2463, December 1998.

[RFC2711] C. Partridge and A. Jackson. IPv6 Router Alert Option. RFC 2711, October 1999.

[RFC3031] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, January 2001.

[RFC3135] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135, June 2001.

[RFC3234] B. Carpenter and S. Brim, Middleboxes: Taxonomy and Issues, RFC 3234, February 2002.

[RFC3344] C. Perkins (ed.). IP Mobility Support for IPv4. RFC 3344, August 2002.

[RFC3828] L-A. Larzon, M. Degermark, S. Pink, L-E. Jonsonn, and G. Fairhurst, The Lightweight User Datagram Protocol (UDP-Lite), RFC 3828, July 2004.

[RFC4782] S. Floyd, M. Allman, A. Jain, and P. Sarolahti. Quick-Start for TCP and IP. RFC 4782, January 2007.

[Abo07] B. Aboba (ed.). Architectural Implications of Link Indications, Internet-Draft "draft-iab-link-indications-07.txt", February 2007. Work in progress.

[BJSK06] B. Briscoe, A. Jacquet, A. Salvatori, and M. Koyabe. Re-ECN: Adding Accountability for Causing Congestion to TCP/IP. Internet-Draft "draft-briscoe-tsvwg-re-ecn-tcp-02", June 2006. Work in progress.

[Cla88] D. D. Clark. The Design Philosophy of the DARPA Internet Protocols. In Proceedings of ACM SIGCOMM ’88, pages 106--114, Stanford, CA, USA.

[DK06] L. Daniel and M. Kojo. Adapting TCP for Vertical Handoffs in Wireless Networks. In Proc. 31st IEEE Conference on Local Computer Networks (LCN), Tampa, FL, USA, November 2006.

[EE06] L. Eggert and W. Eddy. Towards More Expressive Transport-Layer Interfaces. In Proceedings of ACM MOBIARCH ’06, San Francisco, CA, USA, November 2006.

[FPK06] A. Falk, Y. Pryadkin, and D. Katabi. Specification for the Explicit Control Protocol (XCP). Internet-Draft "draft-falk-xcp-spec-02.txt", November 2006. Work in progress.

[G06] F. Gont. ICMP attacks against TCP. Internet-Draft "draft-ietf-tcpm-icmp-attacks-01", October 2006. Work in progress.

[GL03] A. Gurtov and R. Ludwig. Lifetime Packet Discard for Efficient Real-Time Transport over Cellular Links. ACM Mobile Computing and Communications Review, 7(4):32-45, October 2003

[Hal96] F. Halsall. Data Communications, Computer Networks and Open Systems, Fourth edition. Addison-Wesley, 1996.

[HKP01] M. Handley, C. Kreibich and V. Paxson, Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics, Proc. USENIX Security Symposium 2001.

[IEEE21] IEEE 802.21: Media Independent Handover Services. Available at: http://www.ieee802.org/21/

[IntersecBof] Transport Service at the Intermediary BOF (intersec) minutes from IETF-56, San Francisco, CA, USA, March 2003. Available at http://www3.ietf.org/proceedings/03mar/248.htm

[KHR02] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In Proceedings of ACM SIGCOMM 2002, Pittsburgh, PA, USA, August 2002.

[KSE+04] R. Krishnan, J. Sterbenz, W. Eddy, C. Partridge, and M. Allman. Explicit Transport Error Notification (ETEN) for Error-Prone Wireless and Satellite Networks. Computer Networks, 46(3), October 2004

[LR99] R. Ludwig and B. Rathonyi. Link Layer Enhancements for TCP/IP over GSM. In Proceedings of the Conference on Computer Communications (IEEE Infocom), New York, USA, March 1999.

[MAF04] A. Medina, M. Allman, and S. Floyd. Measuring Interactions Between Transport Protocols and Middleboxes. Internet Measurement Conference 2004, August 2004. URL "http://www.icir.org/tbit/".

[RW03] M. Rossi and M. Welzl. On the Impact of IP Option Processing, Preprint-Reihe des Fachbereichs Mathematik - Informatik, No. 15, Institute of Computer Science, University of Innsbruck, Austria, October 2003.

[RW04] M. Rossi and M. Welzl. On the Impact of IP Option Processing - Part 2, Preprint-Reihe des Fachbereichs Mathematik - Informatik, No. 26, Institute of Computer Science, University of Innsbruck, Austria, July 2004.

[SAF06] P. Sarolahti, M. Allman, and S. Floyd. Determining an Appropriate Sending Rate Over an Underutilized Network. Computer Networks Journal, Elsevier, August 2006.

[SEE+06] S. Schuetz, L. Eggert, W. Eddy, Y. Swami, and K. Le. TCP Response to Lower-Layer Connectivity-Change Indications. Internet-Draft "draft-schuetz-tcpm-tcp-rlci-00", May 2006. Work in progress.

[SH06] H. Schulzrinne and R. Hancock. GIST: General Internet Signaling Transport.

Internet-Draft "draft-ietf-nsis-ntlp-10", July 2006. Work in progress.

[SKDK06] P. Sarolahti, J. Korhonen, L. Daniel, and M. Kojo. Using Quick-Start to Improve TCP Performance with Vertical Hand-offs. In Proc. IEEE Workshop on Wireless Local Networks (WLN) 2006, Tampa, FL, USA, November 2006.

[SD06] R. Stewart, M. Dalal (ed.). Improving TCP's Robustness to Blind In-Window Attacks. Internet-Draft "draft-ietf-tcpm-tcpsecure-05.txt", June 2006. Work in progress.

[TGM+06] F. Teraoka, K. Gogo, K. Mitsuya, R. Shibui, and K. Mitani. Unified L2 Abstractions for L3-Driven Fast Handover. Internet-Draft "draft-irtf-mobopts-l2-abstractions-01.txt", September 2006. Work in progress.

[Tou06] J. Touch. Defending TCP Against Spoofing Attacks. Internet-Draft "draft-ietf-tcpm-tcp-antispoof-05.txt", October 2006. Work in progress.

[TrigTranBof] Triggers for Transport (trigtran) Bof minutes from IETF-56, San Francisco, CA, USA, March 2003. Available at http://www3.ietf.org/proceedings/03mar/251.htm

[Wel03] M. Welzl, PMTU-Options: Path MTU Discovery Using Options. Expired Internet-Draft "draft-welzl-pmtud-options-01.txt", February 2003. URL "http://www.welzl.at/research/publications/".

[XSSK05] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman. One More Bit Is Enough. In Proceedings of SIGCOMM 2005, August 2005.

[ZDPS01] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, On the Constancy of Internet Path Properties, Proc. ACM SIGCOMM Internet Measurement Workshop, November 2001.

Acknowledgements

The authors would like to thank Scott Brim, Bob Briscoe, Wesley Eddy, Fernando Gont, Simon Schuetz, Gregory Woodhouse, and Qiaobing Xie for useful comments that have helped to improve this document.

AUTHORS' ADDRESSES

Pasi Sarolahti
Nokia Research Center
P.O. Box 407
FI-00045 NOKIA GROUP
Finland
Phone: +358 50 4876607
Email: pasi.sarolahti@nokia.com

Sally Floyd
Phone: +1 (510) 666-2989
ICIR (ICSI Center for Internet Research)
Email: floyd@icir.org
URL: http://www.icir.org/floyd/

Markku Kojo
University of Helsinki
Department of Computer Science
P.O. Box 68
FIN-00014 UNIVERSITY OF HELSINKI
Finland
Phone: +358 9 191 51305
EMail: kojo@cs.helsinki.fi

Full Copyright Statement

Intellectual Property