Internet Engineering Task Force (IETF)
Request for Comments: 7068
Category: Informational
ISSN: 2070-1721

E. McMurry
B. Campbell
Oracle

November 2013

Diameter Overload Control Requirements

Abstract

When a Diameter server or agent becomes overloaded, it needs to be able to gracefully reduce its load, typically by advising clients to reduce traffic for some period of time. Otherwise, it must continue to expend resources parsing and responding to Diameter messages, possibly resulting in a progressively severe overload condition. The existing Diameter mechanisms are not sufficient for managing overload conditions. This document describes the limitations of the existing mechanisms. Requirements for new overload management mechanisms are also provided.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7068.

Table of Contents

1. Introduction

A Diameter [RFC6733] node is said to be overloaded when it has insufficient resources to successfully process all of the Diameter requests that it receives. When a node becomes overloaded, it needs to be able to gracefully reduce its load, typically by advising clients to reduce traffic for some period of time. Otherwise, it must continue to expend resources parsing and responding to Diameter messages, possibly resulting in a progressively severe overload condition. The existing mechanisms provided by Diameter are not sufficient for managing overload conditions. This document describes the limitations of the existing mechanisms and provides requirements for new overload management mechanisms.

This document draws on the work done on SIP overload control ([RFC5390], [RFC6357]) as well as on experience gained via overload handling in Signaling System No. 7 (SS7) networks and studies done by the Third Generation Partnership Project (3GPP) (Section 3).

Diameter is not typically an end-user protocol; rather, it is generally used as one component in support of some end-user activity.

For example, a SIP server might use Diameter to authenticate and authorize user access. Overload in the Diameter backend infrastructure will likely impact the experience observed by the end user in the SIP application.

The impact of Diameter overload on the client application (a client application may use the Diameter protocol and other protocols to do its job) is beyond the scope of this document.

This document presents non-normative descriptions of causes of overload, along with related scenarios and studies. Finally, it offers a set of normative requirements for an improved overload indication mechanism.

1.1. Documentation Conventions

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 defined in [RFC2119], with the exception that they are not intended for interoperability of implementations. Rather, they are used to describe requirements towards future specifications where the interoperability requirements will be defined.

The terms "client", "server", "agent", "node", "peer", "upstream", and "downstream" are used as defined in [RFC6733].

1.2. Causes of Overload

Overload occurs when an element, such as a Diameter server or agent, has insufficient resources to successfully process all of the traffic it is receiving. Resources include all of the capabilities of the element used to process a request, including CPU processing, memory, I/O, and disk resources. It can also include external resources such as a database or DNS server, in which case the CPU, processing, memory, I/O, and disk resources of those elements are effectively part of the logical element processing the request.

External resources can include upstream Diameter nodes; for example, a Diameter agent can become effectively overloaded if one or more upstream nodes are overloaded.

A Diameter node can become overloaded due to request levels that exceed its capacity, a reduction of available resources (for example, a local or upstream hardware failure), or a combination of the two.

Overload can occur for many reasons, including:

Inadequate capacity: When designing Diameter networks, that is, application-layer multi-node Diameter deployments, it can be very difficult to predict all scenarios that may cause elevated traffic. It may also be more costly to implement support for some scenarios than a network operator may deem worthwhile. This results in the likelihood that a Diameter network will not have adequate capacity to handle all situations.

Dependency failures: A Diameter node can become overloaded because a resource on which it depends has failed or become overloaded, greatly reducing the logical capacity of the node. In these cases, even minimal traffic might cause the node to go into overload. Examples of such dependency overloads include DNS servers, databases, disks, and network interfaces that have failed or become overloaded.

Component failures: A Diameter node can become overloaded when it is a member of a cluster of servers that each share the load of traffic and one or more of the other members in the cluster fail. In this case, the remaining nodes take over the work of the failed nodes. Normally, capacity planning takes such failures into account, and servers are typically run with enough spare capacity to handle failure of another node. However, unusual failure conditions can cause many nodes to fail at once. This is often the case with software failures, where a bad packet or bad database entry hits the same bug in a set of nodes in a cluster.

Network-initiated traffic flood: Certain access network events can precipitate floods of Diameter signaling traffic. For example, operational changes can trigger avalanche restarts, or frequent radio overlay handovers can generate excessive authorization requests. Failure of a Diameter proxy may also result in a large amount of signaling as connections and sessions are reestablished.

Subscriber-initiated traffic flood: Large gatherings of subscribers or events that result in many subscribers interacting with the network in close time proximity can result in Diameter signaling traffic floods. For example, the finale of a large fireworks show could be immediately followed by many subscribers posting messages, pictures, and videos concentrated on one portion of a network. Subscriber devices such as smartphones may use aggressive registration strategies that generate unusually high Diameter traffic loads.

DoS attacks: An attacker wishing to disrupt service in the network can cause a large amount of traffic to be launched at a target element. This can be done from a central source of traffic or through a distributed DoS attack. In all cases, the volume of traffic well exceeds the capacity of the element, sending the system into overload.

1.3. Effects of Overload

Modern Diameter networks, composed of application-layer multi-node deployments of Diameter elements, may operate at very large transaction volumes. If a Diameter node becomes overloaded or, even worse, fails completely, a large number of messages may be lost very quickly. Even with redundant servers, many messages can be lost in the time it takes for failover to complete. While a Diameter client or agent should be able to retry such requests, an overloaded peer may cause a sudden large increase in the number of transactions needing to be retried, rapidly filling local queues or otherwise contributing to local overload. Therefore, Diameter devices need to be able to shed load before critical failures can occur.

1.4. Overload vs. Network Congestion

This document uses the term "overload" to refer to application-layer overload at Diameter nodes. This is distinct from "network congestion", that is, congestion that occurs at the lower networking layers that may impact the delivery of Diameter messages between nodes. This document recognizes that element overload and network congestion are interrelated, and that overload can contribute to network congestion and vice versa.

Network congestion issues are better handled by the transport protocols. Diameter uses TCP and the Stream Control Transmission Protocol (SCTP), both of which include congestion management features. Analysis of whether those features are sufficient for transport-level congestion between Diameter nodes and of any work to further mitigate network congestion is out of scope for both this document and the work proposed by it.

1.5. Diameter Applications in a Broader Network

Most elements using Diameter applications do not use Diameter exclusively. It is important to realize that overload of an element can be caused by a number of factors that may be unrelated to the processing of Diameter or Diameter applications.

An element that doesn't use Diameter exclusively needs to be able to signal to Diameter peers that it is experiencing overload regardless of the cause of the overload, since the overload will affect that element's ability to process Diameter transactions. If the element communicates with protocols other than Diameter, it may also need to signal the overload situation on these protocols, depending on its function and the architecture of the network and application for which it is providing services. Whether that is necessary can only be decided within the context of that architecture and use cases. This specification details the requirements for a mechanism for signaling overload with Diameter; this mechanism provides Diameter nodes the ability to inform their Diameter peers of overload, mitigating that part of the issue. Diameter nodes may need to use this, as well as other mechanisms, to solve their broader overload issues. Indicating overload on protocols other than Diameter is out of scope for this document and for the work proposed by it.

2. Overload Control Scenarios

Several Diameter deployment scenarios exist that may impact overload management. The following scenarios help motivate the requirements for an overload management mechanism.

These scenarios are by no means exhaustive and are in general simplified for the sake of clarity. In particular, this document assumes for the sake of clarity that the client sends Diameter requests to the server, and the server sends responses to the client, even though Diameter supports bidirectional applications. Each direction in such an application can be modeled separately.

In a large-scale deployment, many of the nodes represented in these scenarios would be deployed as clusters of servers. This document assumes that such a cluster is responsible for managing its own

internal load-balancing and overload management so that it appears as a single Diameter node. That is, other Diameter nodes can treat it as a single, monolithic node for the purposes of overload management.

These scenarios do not illustrate the client application. As mentioned in Section 1, Diameter is not typically an end-user protocol; rather, it is generally used in support of some other client application. These scenarios do not consider the impact of Diameter overload on the client application.

2.1. Peer-to-Peer Scenarios

This section describes Diameter peer-to-peer scenarios, that is, scenarios where a Diameter client talks directly with a Diameter server, without the use of a Diameter agent.

Figure 1 illustrates the simplest possible Diameter relationship. The client and server share a one-to-one peer-to-peer relationship. If the server becomes overloaded, either because the client exceeds the server's capacity or because the server's capacity is reduced due to some resource dependency, the client needs to reduce the amount of Diameter traffic it sends to the server. Since the client cannot forward requests to another server, it must either queue requests until the server recovers or itself become overloaded in the context of the client application and other protocols it may also use.

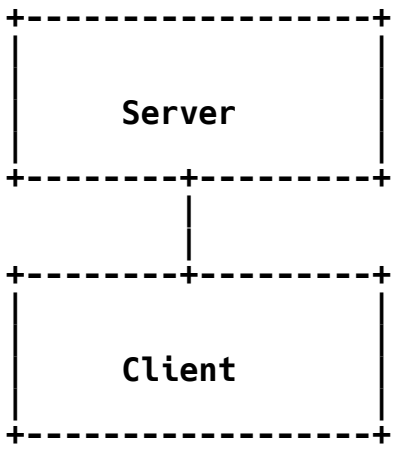

Figure 1: Basic Peer-to-Peer Scenario

Figure 2 shows a similar scenario, except in this case the client has multiple servers that can handle work for a specific realm and application. If Server 1 becomes overloaded, the client can forward traffic to Server 2. Assuming that Server 2 has sufficient reserve capacity to handle the forwarded traffic, the client should be able to continue serving client application protocol users. If Server 1 is approaching overload, but can still handle some number of new requests, it needs to be able to instruct the client to forward a subset of its traffic to Server 2.

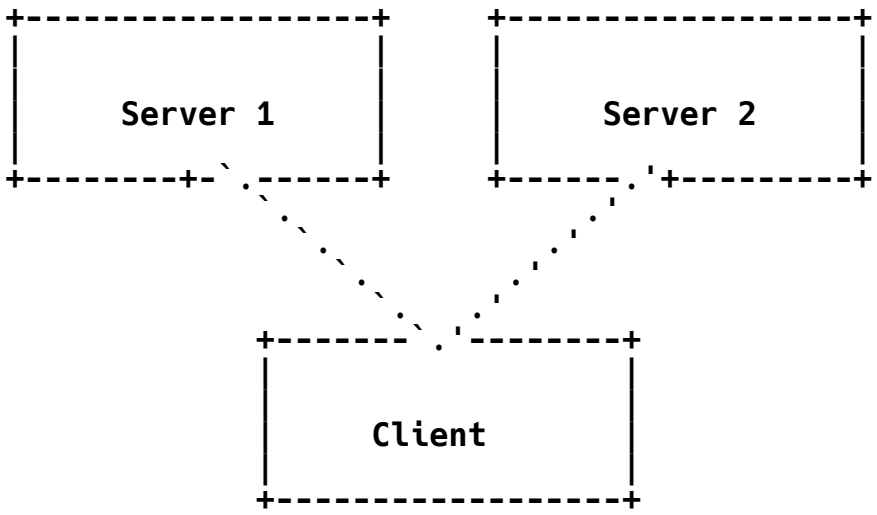

Figure 2: Multiple-Server Peer-to-Peer Scenario

Figure 3 illustrates a peer-to-peer scenario with multiple Diameter realm and application combinations. In this example, Server 2 can handle work for both applications. Each application might have different resource dependencies. For example, a server might need to access one database for Application A and another for Application B. This creates a possibility that Server 2 could become overloaded for Application A but not for Application B, in which case the client would need to divert some part of its Application A requests to Server 1, but the client should not divert any Application B requests. This requires that Server 2 be able to distinguish between applications when it indicates an overload condition to the client.

On the other hand, it's possible that the servers host many applications. If Server 2 becomes overloaded for all applications, it would be undesirable for it to have to notify the client separately for each application. Therefore, it also needs a way to indicate that it is overloaded for all possible applications.

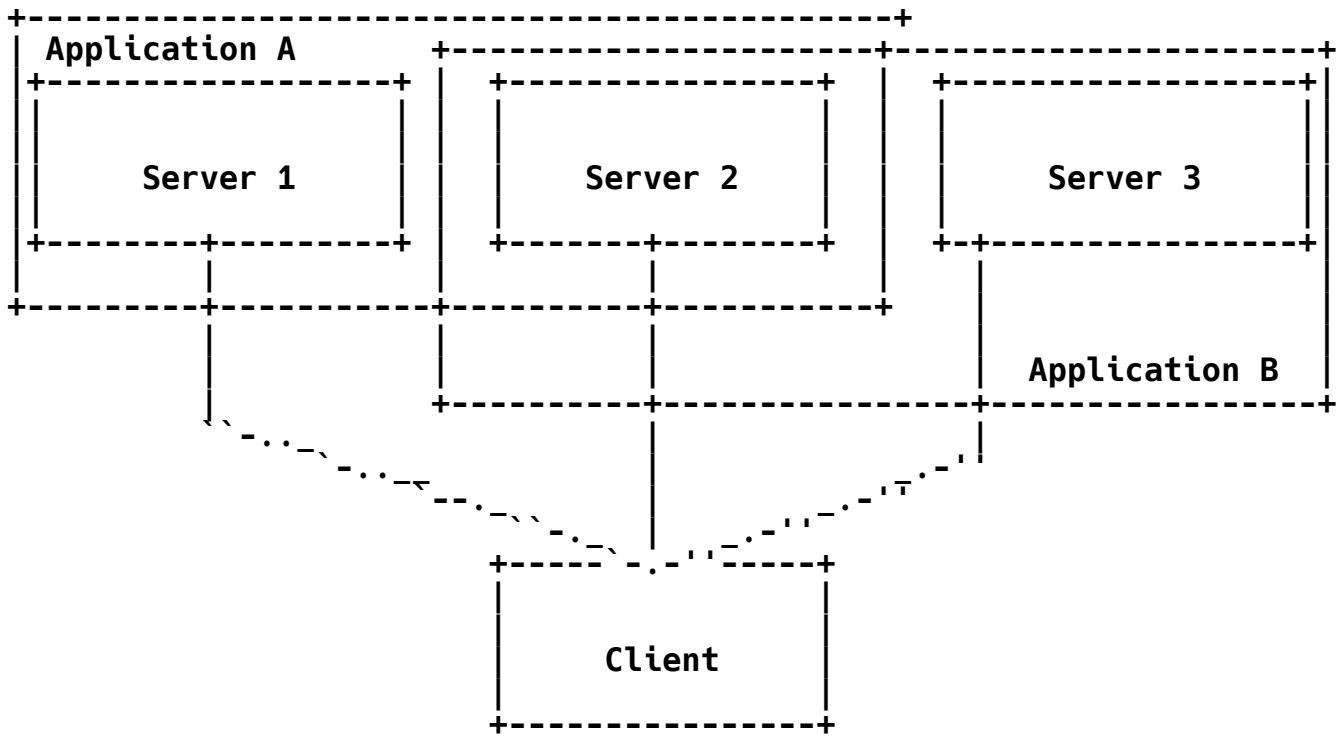

Figure 3: Multiple-Application Peer-to-Peer Scenario

2.2. Agent Scenarios

This section describes scenarios that include a Diameter agent, in the form of either a Diameter relay or Diameter proxy. These scenarios do not consider Diameter redirect agents, since they are more readily modeled as end servers. The examples have been kept simple deliberately, to illustrate basic concepts. Significantly more complicated topologies are possible with Diameter, including multiple intermediate agents in a path connected in a variety of ways.

Figure 4 illustrates a simple Diameter agent scenario with a single client, agent, and server. In this case, overload can occur at the server, at the agent, or both. But in most cases, client behavior is the same whether overload occurs at the server or at the agent. From the client's perspective, server overload and agent overload are the same thing.

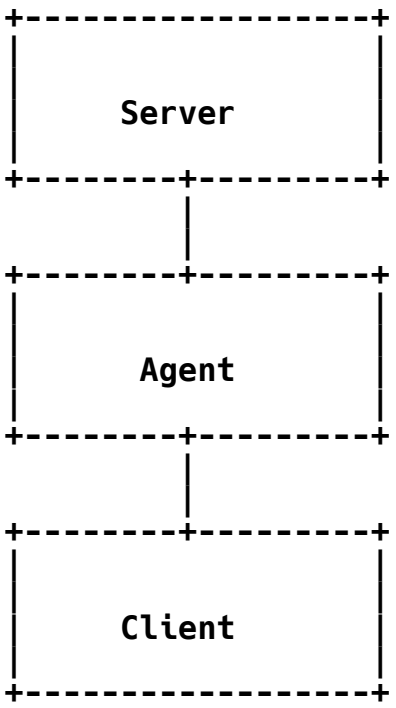

Figure 4: Basic Agent Scenario

Figure 5 shows an agent scenario with multiple servers. If Server 1 becomes overloaded but Server 2 has sufficient reserve capacity, the agent may be able to transparently divert some or all Diameter requests originally bound for Server 1 to Server 2.

In most cases, the client does not have detailed knowledge of the Diameter topology upstream of the agent. If the agent uses dynamic discovery to find eligible servers, the set of eligible servers may not be enumerable from the perspective of the client. Therefore, in most cases the agent needs to deal with any upstream overload issues in a way that is transparent to the client. If one server notifies the agent that it has become overloaded, the notification should not be passed back to the client in a way that the client could mistakenly perceive the agent itself as being overloaded. If the set

of all possible destinations upstream of the agent no longer has sufficient capacity for incoming load, the agent itself becomes effectively overloaded.

On the other hand, there are cases where the client needs to be able to select a particular server from behind an agent. For example, if a Diameter request is part of a multiple-round-trip authentication, or is otherwise part of a Diameter "session", it may have a Destination-Host Attribute-Value Pair (AVP) that requires that the request be served by Server 1. Therefore, the agent may need to inform a client that a particular upstream server is overloaded or otherwise unavailable. Note that there can be many ways a server can be specified, which may have different implications (e.g., by IP address, by host name, etc).

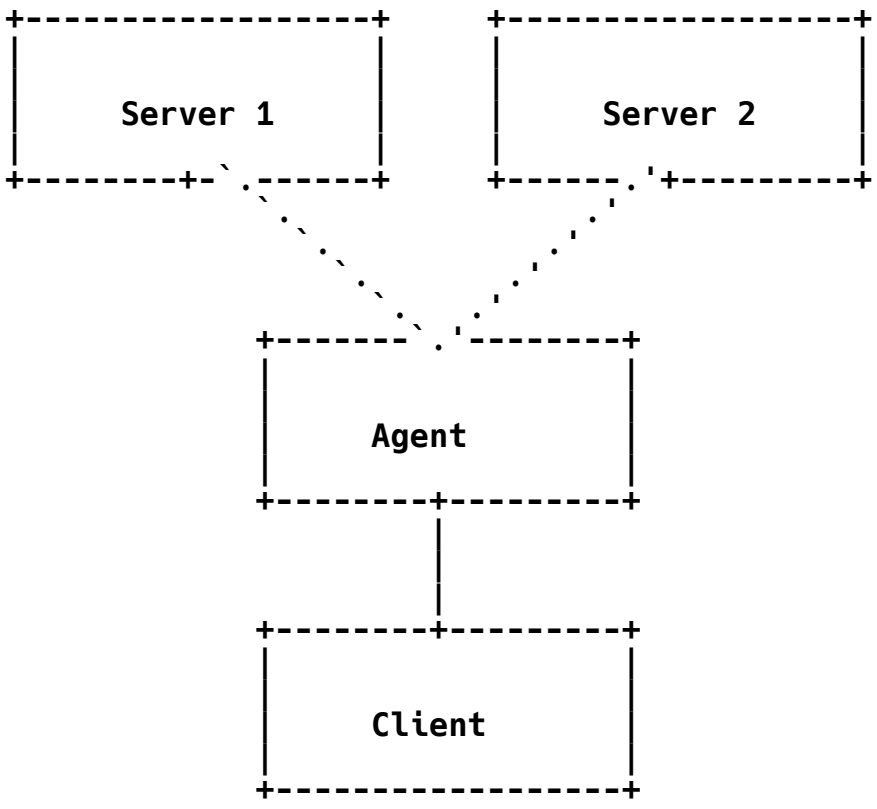

Figure 5: Multiple-Server Agent Scenario

Figure 6 shows a scenario where an agent routes requests to a set of servers for more than one Diameter realm and application. In this scenario, if Server 1 becomes overloaded or unavailable while Server 2 still has available capacity, the agent may effectively operate at reduced capacity for Application A but at full capacity for Application B. Therefore, the agent needs to be able to report that it is overloaded for one application but not for another.

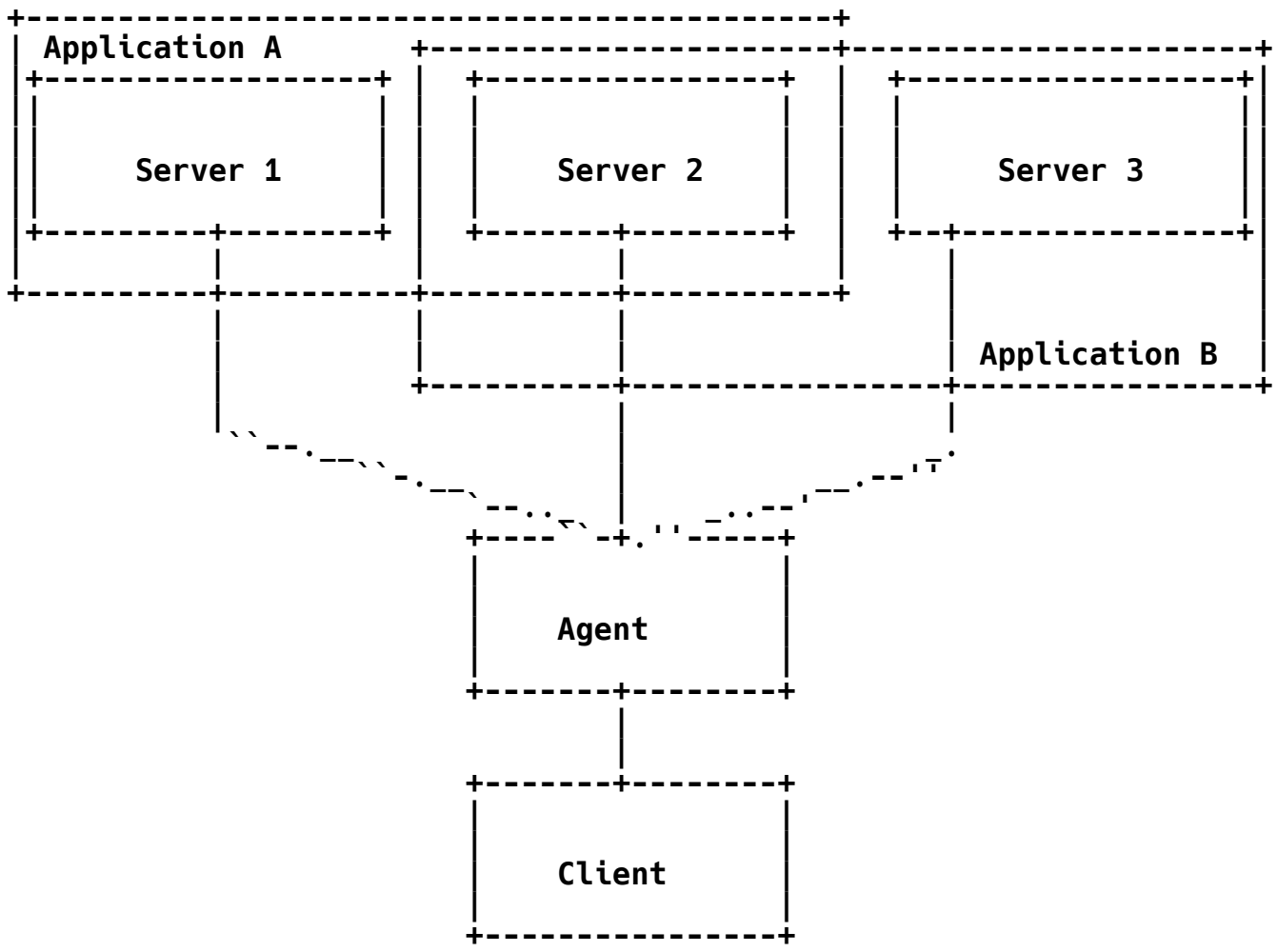

Figure 6: Multiple-Application Agent Scenario

2.3. Interconnect Scenario

Another scenario to consider when looking at Diameter overload is that of multiple network operators using Diameter components connected through an interconnect service, e.g., using IPX (IP Packet eXchange). IPX [IR.34] is an Inter-Operator IP Backbone that provides a roaming interconnection network between mobile operators and service providers. IPX is also used to transport Diameter signaling between operators [IR.88]. Figure 7 shows two network operators with an interconnect network between them. There could be any number of these networks between any two network operators' networks.

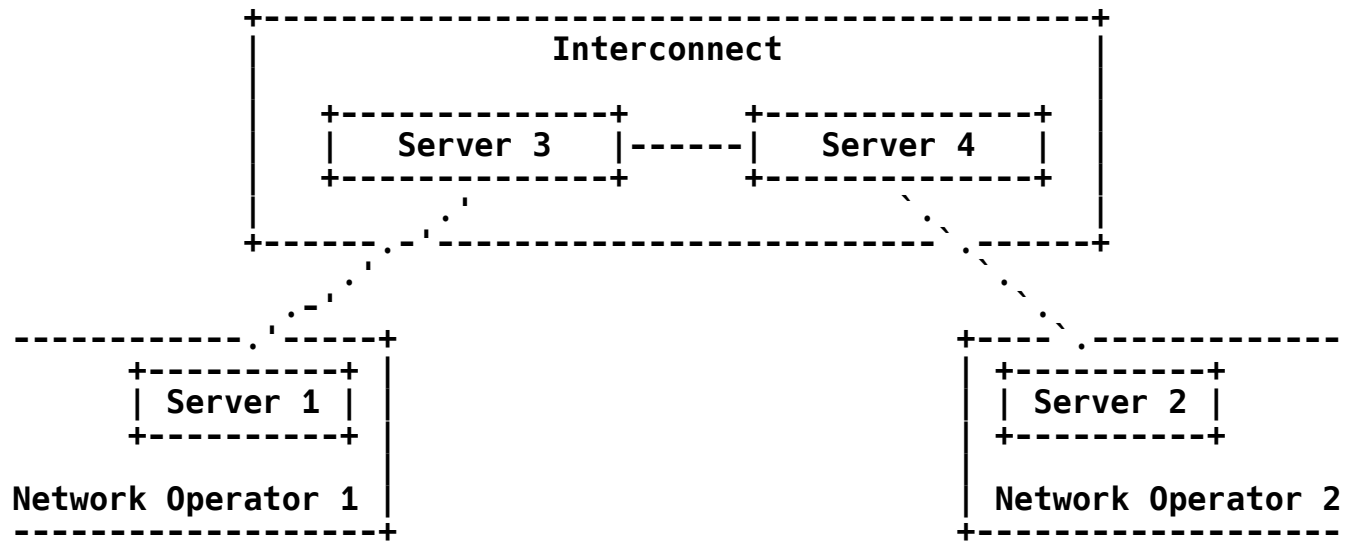

Figure 7: Two-Network Interconnect Scenario

The characteristics of the information that an operator would want to share over such a connection are different from the information shared between components within a network operator's network. For example, network operators may not want to convey topology or operational information; this would in turn limit how much overload and loading information can be sent. For the interconnect scenario shown in Figure 7, Server 2 may want to signal overload to Server 1, to affect traffic coming from Network Operator 1.

This case is distinct from those internal to a network operator's network, where there may be many more elements in a more complicated topology. Also, the elements in the interconnect network may not support Diameter overload control, and the network operators may not want the interconnect network to use overload or loading information. They may only want the information to pass through the interconnect

network without further processing or action by the interconnect network, even if the elements in the interconnect network do support Diameter overload control.

3. Diameter Overload Case Studies

3.1. Overload in Mobile Data Networks

As the number of smartphone devices that are Third Generation (3G) and Long Term Evolution (LTE) enabled continues to expand in mobile networks, there have been situations where high signaling traffic load led to overload events at the Diameter-based Home Location Registers (HLRs) and/or Home Subscriber Servers (HSS) [TR23.843]. The root causes of the HLR overload events were manifold but included hardware failure and procedural errors. The result was high signaling traffic load on the HLR and HSS.

The 3GPP architecture [TS23.002] makes extensive use of Diameter. It is used for mobility management [TS29.272], the IP Multimedia Subsystem (IMS) [TS29.228], and policy and charging control [TS29.212], as well as other functions. The details of the architecture are out of scope for this document, but it is worth noting that there are quite a few Diameter applications, some with quite large amounts of Diameter signaling in deployed networks.

The 3GPP specifications do not currently address overload for Diameter applications or provide a load control mechanism equivalent to those provided in the more traditional SS7 elements in the Global System for Mobile Communications (GSM); see [TS29.002]. The capabilities specified in the 3GPP standards do not adequately address the abnormal condition where excessively high signaling traffic load situations are experienced.

Smartphones, which comprise an increasingly large percentage of mobile devices, contribute much more heavily, relative to non-smartphones, to the continuation of a registration surge, due to their very aggressive registration algorithms. Smartphone behavior contributes to network loading and can contribute to overload conditions. The aggressive smartphone logic is designed to:

a. always have voice and data registration, and

b. constantly try to be on 3G or LTE data (and thus on 3G voice or Voice over LTE (VoLTE) [IR.92]) for their added benefits.

Non-smartphones typically have logic to wait for a time period after registering successfully on voice and data.

The aggressive smartphone registration is problematic in two ways:

- first, by generating excessive signaling load towards the HSS that is ten times the load from a non-smartphone, and
- second, by causing continual registration attempts when a network failure affects registrations through the 3G data network.

3.2. 3GPP Study on Core Network Overload

A study in the 3GPP System Aspects working group 2 (SA2) on core network overload has produced the technical report [TR23.843]. This enumerates several causes of overload in mobile core networks, including portions that are signaled using Diameter. [TR23.843] is a work in progress and is not complete. However, it is useful for pointing out scenarios and the general need for an overload control mechanism for Diameter.

It is common for mobile networks to employ more than one radio technology and to do so in an overlay fashion with multiple technologies present in the same location (such as 2nd or 3rd generation mobile technologies, along with LTE). This presents opportunities for traffic storms when issues occur on one overlay and not another as all devices that had been on the overlay with issues switch. This causes a large amount of Diameter traffic as locations and policies are updated.

Another scenario called out by this study is a flood of registration and mobility management events caused by some element in the core network failing. This flood of traffic from end nodes falls under the network-initiated traffic flood category. There is likely to also be traffic resulting directly from the component failure in this case. A similar flood can occur when elements or components recover as well.

Subscriber-initiated traffic floods are also indicated in this study as an overload mechanism where a large number of mobile devices are attempting to access services at the same time, such as in response to an entertainment event or a catastrophic event.

While this 3GPP study is concerned with the broader effects of these scenarios on wireless networks and their elements, they have implications specifically for Diameter signaling. One of the goals of this document is to provide guidance for a core mechanism that can be used to mitigate the scenarios called out by this study.

4. Existing Mechanisms

Diameter offers both implicit and explicit mechanisms for a Diameter node to learn that a peer is overloaded or unreachable. The implicit mechanism is simply the lack of responses to requests. If a client fails to receive a response in a certain time period, it assumes that the upstream peer is unavailable or is overloaded to the point of effective unavailability. The watchdog mechanism [RFC3539] ensures that transaction responses occur at a certain rate even when there is otherwise little or no other Diameter traffic.

The explicit mechanism can involve specific protocol error responses, where an agent or server tells a downstream peer that it is either too busy to handle a request (DIAMETER_TOO_BUSY) or unable to route a request to an upstream destination (DIAMETER_UNABLE_TO_DELIVER) perhaps because that destination itself is overloaded to the point of unavailability.

Another explicit mechanism, a DPR (Disconnect-Peer-Request) message, can be sent with a Disconnect-Cause of BUSY. This signals the sender's intent to close the transport connection and requests that the client not reconnect.

Once a Diameter node learns via one of these mechanisms that an upstream peer has become overloaded, it can then attempt to take action to reduce the load. This usually means forwarding traffic to an alternate destination, if available. If no alternate destination is available, the node must either reduce the number of messages it originates (in the case of a client) or inform the client to reduce traffic (in the case of an agent).

Diameter requires the use of a congestion-managed transport layer, currently TCP or SCTP, to mitigate network congestion. It is expected that these transports manage network congestion and that issues with transport (e.g., congestion propagation and window management) are managed at that level. But even with a congestion-managed transport, a Diameter node can become overloaded at the Diameter protocol or application layers due to the causes described in Section 1.2, and congestion-managed transports do not provide facilities (and are at the wrong level) to handle server overload. Transport-level congestion management is also not sufficient to address overload in cases of multi-hop and multi-destination signaling.

5. Issues with the Current Mechanisms

The currently available Diameter mechanisms for indicating an overload condition are not adequate to avoid service outages due to overload. This inadequacy may, in turn, contribute to broader impacts resulting from overload due to unresponsive Diameter nodes causing application-layer or transport-layer retransmissions. In particular, they do not allow a Diameter agent or server to shed load as it approaches overload. At best, a node can only indicate that it needs to entirely stop receiving requests, i.e., that it has effectively failed. Even that is problematic due to the inability to indicate durational validity on the transient errors available in the base Diameter protocol. Diameter offers no mechanism to allow a node to indicate different overload states for different categories of messages, for example, if it is overloaded for one Diameter application but not another.

5.1. Problems with Implicit Mechanism

The implicit mechanism doesn't allow an agent or server to inform the client of a problem until it is effectively too late to do anything about it. The client does not know that it needs to take action until the upstream node has effectively failed. A Diameter node has no opportunity to shed load early to avoid collapse in the first place.

Additionally, the implicit mechanism cannot distinguish between overload of a Diameter node and network congestion. Diameter treats the failure to receive an answer as a transport failure.

5.2. Problems with Explicit Mechanisms

The Diameter specification is ambiguous on how a client should handle receipt of a DIAMETER_TOO_BUSY response. The base specification [RFC6733] indicates that the sending client should attempt to send the request to a different peer. It makes no suggestion that the receipt of a DIAMETER_TOO_BUSY response should affect future Diameter messages in any way.

The Authentication, Authorization, and Accounting (AAA) Transport Profile [RFC3539] recommends that a AAA node that receives a "Busy" response failover all remaining requests to a different agent or server. But while the Diameter base specification explicitly depends on [RFC3539] to define transport behavior, it does not refer to [RFC3539] in the description of behavior on receipt of a DIAMETER_TOO_BUSY error. There's a strong likelihood that at least some implementations will continue to send Diameter requests to an upstream peer even after receiving a DIAMETER_TOO_BUSY error.

BCP 41 [RFC2914] describes, among other things, how end-to-end application behavior can help avoid congestion collapse. In particular, an application should avoid sending messages that will never be delivered or processed. The DIAMETER_TOO_BUSY behavior as described in the Diameter base specification fails at this, since if an upstream node becomes overloaded, a client attempts each request and does not discover the need to failover the request until the initial attempt fails.

The situation is improved if implementations follow the [RFC3539] recommendation to keep state about upstream peer overload. But even then, the Diameter specification offers no guidance on how long a client should wait before retrying the overloaded destination. If an agent or server supports multiple realms and/or applications, DIAMETER_TOO_BUSY offers no way to indicate that it is overloaded for one application but not another. A DIAMETER_TOO_BUSY error can only indicate overload at a "whole server" scope.

Agent processing of a DIAMETER_TOO_BUSY response is also problematic as described in the base specification. DIAMETER_TOO_BUSY is defined as a protocol error. If an agent receives a protocol error, it may either handle it locally or forward the response back towards the downstream peer. If a downstream peer receives the DIAMETER_TOO_BUSY response, it may stop sending all requests to the agent for some period of time, even though the agent may still be able to deliver requests to other upstream peers.

DIAMETER_UNABLE_TO_DELIVER errors, or using DPR with cause code BUSY, also have no mechanisms for specifying the scope or cause of the failure, or the durational validity.

The issues with error responses described in [RFC6733] extend beyond the particular issues for overload control and have been addressed in an ad hoc fashion by various implementations. Addressing these in a standard way would be a useful exercise, but it is beyond the scope of this document.

6. Extensibility and Application Independence

Given the variety of scenarios in which Diameter elements can be deployed and the variety of roles they can fulfill with Diameter and other technologies, a single algorithm for handling overload may not be sufficient. For purposes of this discussion, an algorithm is inclusive of behavior for control of overload but does not encompass the general mechanism for transporting control information. This effort cannot anticipate all possible future scenarios and roles. Extensibility, particularly of algorithms used to deal with overload, will be important to cover these cases.

Similarly, the scopes to which overload information may apply may include cases that have not yet been considered. Extensibility in this area will also be important.

The basic mechanism is intended to be application independent, that is, a Diameter node can use it across any existing and future Diameter applications and expect reasonable results. Certain Diameter applications might, however, benefit from application-specific behavior over and above the mechanism's defaults. For example, an application specification might specify relative priorities of messages or selection of a specific overload control algorithm.

7. Solution Requirements

This section proposes requirements for an improved mechanism to control Diameter overload, with the goals of addressing the issues described in Section 5 and supporting the scenarios described in Section 2. These requirements are stated primarily in terms of individual node behavior to inform the design of the improved mechanism; solution designers should keep in mind that the overall goal is improved overall system behavior across all the nodes involved, not just improved behavior from specific individual nodes.

7.1. General

REQ 1: The solution MUST provide a communication method for Diameter nodes to exchange load and overload information.

REQ 2: The solution MUST allow Diameter nodes to support overload control regardless of which Diameter applications they support. Diameter clients and agents must be able to use the received load and overload information to support graceful behavior during an overload condition. Graceful behavior under overload conditions is best described by REQ 3.

REQ 3: The solution MUST limit the impact of overload on the overall useful throughput of a Diameter server, even when the incoming load on the network is far in excess of its capacity. The overall useful throughput under load is the ultimate measure of the value of a solution.

REQ 4: Diameter allows requests to be sent from either side of a connection, and either side of a connection may have need to provide its overload status. The solution MUST allow each side of a connection to independently inform the other of its overload status.

REQ 5: Diameter allows nodes to determine their peers via dynamic discovery or manual configuration. The solution MUST work consistently without regard to how peers are determined.

REQ 6: The solution designers SHOULD seek to minimize the amount of new configuration required in order to work. For example, it is better to allow peers to advertise or negotiate support for the solution, rather than to require that this knowledge be configured at each node.

7.2. Performance

REQ 7: The solution and any associated default algorithm(s) MUST ensure that the system remains stable. At some point after an overload condition has ended, the solution MUST enable capacity to stabilize and become equal to what it would be in the absence of an overload condition. Note that this also requires that the solution MUST allow nodes to shed load without introducing non-converging oscillations during or after an overload condition.

REQ 8: Supporting nodes MUST be able to distinguish current overload information from stale information.

REQ 9: The solution MUST function across fully loaded as well as quiescent transport connections. This is partially derived from the requirement for stability in REQ 7.

REQ 10: Consumers of overload information MUST be able to determine when the overload condition improves or ends.

REQ 11: The solution MUST be able to operate in networks of different sizes.

REQ 12: When a single network node fails, goes into overload, or suffers from reduced processing capacity, the solution MUST make it possible to limit the impact of the affected node on other nodes in the network. This helps to prevent a small-scale failure from becoming a widespread outage.

REQ 13: The solution MUST NOT introduce substantial additional work for a node in an overloaded state. For example, a requirement for an overloaded node to send overload information every time it received a new request would introduce substantial work.

REQ 14: Some scenarios that result in overload involve a rapid increase of traffic with little time between normal levels and levels that induce overload. The solution SHOULD provide for rapid feedback when traffic levels increase.

REQ 15: The solution MUST NOT interfere with the congestion control mechanisms of underlying transport protocols. For example, a solution that opened additional TCP connections when the network is congested would reduce the effectiveness of the underlying congestion control mechanisms.

7.3. Heterogeneous Support for Solution

REQ 16: The solution is likely to be deployed incrementally. The solution MUST support a mixed environment where some, but not all, nodes implement it.

REQ 17: In a mixed environment with nodes that support the solution and nodes that do not, the solution MUST NOT result in materially less useful throughput during overload as would have resulted if the solution were not present. It SHOULD result in less severe overload in this environment.

REQ 18: In a mixed environment of nodes that support the solution and nodes that do not, the solution MUST NOT preclude elements that support overload control from treating elements that do not support overload control in an equitable fashion relative to those that do. Users and operators of nodes that do not support the solution MUST NOT unfairly benefit from the solution. The solution specification SHOULD provide guidance to implementors for dealing with elements not supporting overload control.

REQ 19: It MUST be possible to use the solution between nodes in different realms and in different administrative domains.

REQ 20: Any explicit overload indication MUST be clearly distinguishable from other errors reported via Diameter.

REQ 21: In cases where a network node fails, is so overloaded that it cannot process messages, or cannot communicate due to a network failure, it may not be able to provide explicit indications of the nature of the failure or its levels of overload. The solution MUST result in at least as much useful throughput as would have resulted if the solution were not in place.

7.4. Granular Control

REQ 22: The solution MUST provide a way for a node to throttle the amount of traffic it receives from a peer node. This throttling SHOULD be graded so that it can be applied gradually as offered load increases. Overload is not a binary state; there may be degrees of overload.

REQ 23: The solution MUST provide sufficient information to enable a load-balancing node to divert messages that are rejected or otherwise throttled by an overloaded upstream node to other upstream nodes that are the most likely to have sufficient capacity to process them.

REQ 24: The solution MUST provide a mechanism for indicating load levels, even when not in an overload condition, to assist nodes in making decisions to prevent overload conditions from occurring.

7.5. Priority and Policy

REQ 25: The base specification for the solution SHOULD offer general guidance on which message types might be desirable to send or process over others during times of overload, based on application-specific considerations. For example, it may be more beneficial to process messages for existing sessions ahead of new sessions. Some networks may have a requirement to give priority to requests associated with emergency sessions. Any normative or otherwise detailed definition of the relative priorities of message types during an overload condition will be the responsibility of the application specification.

REQ 26: The solution MUST NOT prevent a node from prioritizing requests based on any local policy, so that certain requests are given preferential treatment, given additional retransmission, not throttled, or processed ahead of others.

7.6. Security

REQ 27: The solution MUST NOT provide new vulnerabilities to malicious attack or increase the severity of any existing vulnerabilities. This includes vulnerabilities to DoS and DDoS attacks as well as replay and man-in-the-middle attacks. Note that the Diameter base specification [RFC6733] lacks end-to-end security, and this must be considered (see Security Considerations in this document (Section 8)). Note

that this requirement was expressed at a high level so as to not preclude any particular solution. Is is expected that the solution will address this in more detail.

REQ 28: The solution MUST NOT depend on being deployed in environments where all Diameter nodes are completely trusted. It SHOULD operate as effectively as possible in environments where other nodes are malicious; this includes preventing malicious nodes from obtaining more than a fair share of service. Note that this does not imply any responsibility on the solution to detect, or take countermeasures against, malicious nodes.

REQ 29: It MUST be possible for a supporting node to make authorization decisions about what information will be sent to peer nodes based on the identity of those nodes. This allows a domain administrator who considers the load of their nodes to be sensitive information to restrict access to that information. Of course, in such cases, there is no expectation that the solution itself will help prevent overload from that peer node.

REQ 30: The solution MUST NOT interfere with any Diameter-compliant method that a node may use to protect itself from overload from non-supporting nodes or from denial-of-service attacks.

7.7. Flexibility and Extensibility

REQ 31: There are multiple situations where a Diameter node may be overloaded for some purposes but not others. For example, this can happen to an agent or server that supports multiple applications, or when a server depends on multiple external resources, some of which may become overloaded while others are fully available. The solution MUST allow Diameter nodes to indicate overload with sufficient granularity to allow clients to take action based on the overloaded resources without unreasonably forcing available capacity to go unused. The solution MUST support specification of overload information with granularities of at least "Diameter node", "realm", and "Diameter application" and MUST allow extensibility for others to be added in the future.

REQ 32: The solution MUST provide a method for extending the information communicated and the algorithms used for overload control.

REQ 33: The solution MUST provide a default algorithm that is mandatory to implement.

REQ 34: The solution SHOULD provide a method for exchanging overload and load information between elements that are connected by intermediaries that do not support the solution.

8. Security Considerations

A Diameter overload control mechanism is primarily concerned with the load-related and overload-related behavior of nodes in a Diameter network, and the information used to affect that behavior. Load and overload information is shared between nodes and directly affects the behavior, and thus the information is potentially vulnerable to a number of methods of attack.

Load and overload information may also be sensitive from both business and network protection viewpoints. Operators of Diameter equipment want to control the visibility of load and overload information to keep it from being used for competitive intelligence or for targeting attacks. It is also important that the Diameter overload control mechanism not introduce any way in which any other information carried by Diameter is sent inappropriately.

Note that the Diameter base specification [RFC6733] lacks end-to-end security, making it difficult for non-adjacent nodes to verify the authenticity and ownership of load and overload information. Authentication of load and overload information helps to alleviate several of the security issues listed in this section.

This document includes requirements intended to mitigate the effects of attacks and to protect the information used by the mechanism. This section discusses potential security considerations for overload control solutions. This discussion provides the motivation for several normative requirements described in Section 7. The discussion includes specific references to the normative requirements that apply for each issue.

8.1. Access Control

To control the visibility of load and overload information, sending should be subject to some form of authentication and authorization of the receiver. It is also important to the receivers that they are confident the load and overload information they receive is from a legitimate source. REQ 28 requires that the solution work without assuming that all Diameter nodes in a network are trusted for the purposes of exchanging overload and load information. REQ 29 requires that the solution let nodes restrict unauthorized parties

from seeing overload information. Note that this implies a certain amount of configurability on the nodes supporting the Diameter overload control mechanism.

8.2. Denial-of-Service Attacks

An overload control mechanism provides a very attractive target for denial-of-service attacks. A small number of messages may effect a large service disruption by falsely reporting overload conditions. Alternately, attacking servers nearing, or in, overload may also be facilitated by disrupting their overload indications, potentially preventing them from mitigating their overload condition.

A design goal for the Diameter overload control mechanism is to minimize or eliminate the possibility of using the mechanism for this type of attack. More strongly, REQ 27 forbids the solution from introducing new vulnerabilities to malicious attack. Additionally, REQ 30 stipulates that the solution not interfere with other mechanisms used for protection against denial-of-service attacks.

As the intent of some denial-of-service attacks is to induce overload conditions, an effective overload control mechanism should help to mitigate the effects of such an attack.

8.3. Replay Attacks

An attacker that has managed to obtain some messages from the overload control mechanism may attempt to affect the behavior of nodes supporting the mechanism by sending those messages at potentially inopportune times. In addition to time shifting, replay attacks may send messages to other nodes as well (target shifting).

A design goal for the Diameter overload control solution is to minimize or eliminate the possibility of causing disruption by using a replay attack on the Diameter overload control mechanism. (Allowing a replay attack using the overload control solution would violate REQ 27.)

8.4. Man-in-the-Middle Attacks

By inserting themselves between two nodes supporting the Diameter overload control mechanism, an attacker may potentially both access and alter the information sent between those nodes. This can be used for information gathering for business intelligence and attack targeting, as well as direct attacks.

REQs 27, 28, and 29 imply a need to prevent man-in-the-middle attacks on the overload control solution. A transport using Transport Layer Security (TLS) and/or IPsec may be desirable for this purpose.

8.5. Compromised Hosts

A compromised host that supports the Diameter overload control mechanism could be used for information gathering as well as for sending malicious information to any Diameter node that would normally accept information from it. While it is beyond the scope of the Diameter overload control mechanism to mitigate any operational interruption to the compromised host, REQs 28 and 29 imply a need to minimize the impact that a compromised host can have on other nodes through the use of the Diameter overload control mechanism. Of course, a compromised host could be used to cause damage in a number of other ways. This is out of scope for a Diameter overload control mechanism.

9. References

9.1. Normative References

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

[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.

9.2. Informative References

[RFC5390] Rosenberg, J., "Requirements for Management of Overload in the Session Initiation Protocol", RFC 5390, December 2008.

[RFC6357] Hilt, V., Noel, E., Shen, C., and A. Abdelal, "Design Considerations for Session Initiation Protocol (SIP) Overload Control", RFC 6357, August 2011.

[TR23.843] 3GPP, "Study on Core Network (CN) overload solutions", TR 23.843 1.2.0, Work in Progress, October 2013.

[IR.34] GSMA, "Inter-Service Provider IP Backbone Guidelines", IR 34 9.1, May 2013.

[IR.88] GSMA, "LTE Roaming Guidelines", IR 88 9.0, January 2013.

[IR.92] GSMA, "IMS Profile for Voice and SMS", IR 92 7.0, March 2013.

[TS23.002] 3GPP, "Network Architecture", TS 23.002 12.2.0, June 2013.

[TS29.272] 3GPP, "Evolved Packet System (EPS); Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter protocol", TS 29.272 12.2.0, September 2013.

[TS29.212] 3GPP, "Policy and Charging Control (PCC) over Gx/Sd reference point", TS 29.212 12.2.0, September 2013.

[TS29.228] 3GPP, "IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signalling flows and message contents", TS 29.228 12.0.0, September 2013.

[TS29.002] 3GPP, "Mobile Application Part (MAP) specification", TS 29.002 12.2.0, September 2013.

Appendix A. Contributors

Significant contributions to this document were made by Adam Roach and Eric Noel.

Appendix B. Acknowledgements

Review of, and contributions to, this specification by Martin Dolly, Carolyn Johnson, Jianrong Wang, Imtiaz Shaikh, Jouni Korhonen, Robert Sparks, Dieter Jacobsohn, Janet Gunn, Jean-Jacques Trottin, Laurent Thiebaut, Andrew Booth, and Lionel Morand were most appreciated. We would like to thank them for their time and expertise.

Authors' Addresses

Eric McMurry
Oracle
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
US

EMail: emcmurry@computer.org

Ben Campbell
Oracle
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
US

EMail: ben@nostrum.com