Diameter Agent Overload

IETF 88 - Vancouver

Goal

- Get consensus from the working group that Agent overload needs to be addressed
- If so, get guidance on the best path forward

Background

- DOC-DT decided to not address handling of overloaded agents in base DOIC specification.

Assertion

- A complete solution to handling Diameter overload requires addressing overload of all nodes in a Diameter network, including agents.

Requirements –
An agent is a Diameter node.

- REQ 1: The solution MUST provide a communication method for Diameter nodes to exchange load and overload information.
- 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 this on other nodes in the network. This helps to prevent a small-scale failure from becoming a widespread outage.
- Other requirements also apply.

Question 1

- Do we agree that we need to address the handling of Agent Overload?

Behavior

- Minimal new behavior required in clients
  - Behavior for a client is the same as the case where there is a direct connection between the client and multiple servers and the client gets a realm overload report from one of the servers.
- Overload abatement is handled by peer on a hop-by-hop basis.
  - Agents are required to inspect overload reports and act on those from peer agents.
- No change in loss abatement algorithm for throttled requests.

Question 2

- How should agent overload handling be specified?
  - Option 1 – As an extension
  - Option 2 – As part of the base DOIC specification

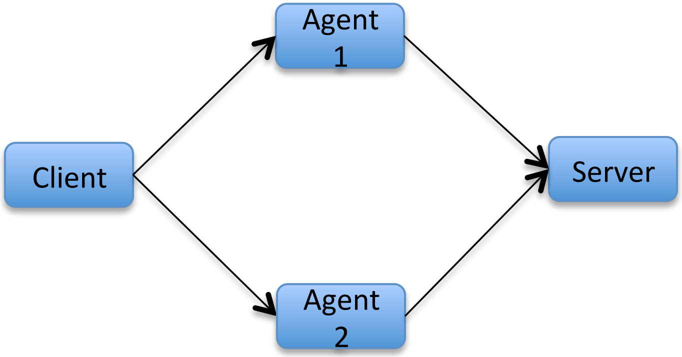
BACKUP SLIDES

Use Cases

- Single Agent
- Redundant Agents
- Agent Chains
- Interaction between agent overload and end-point overload

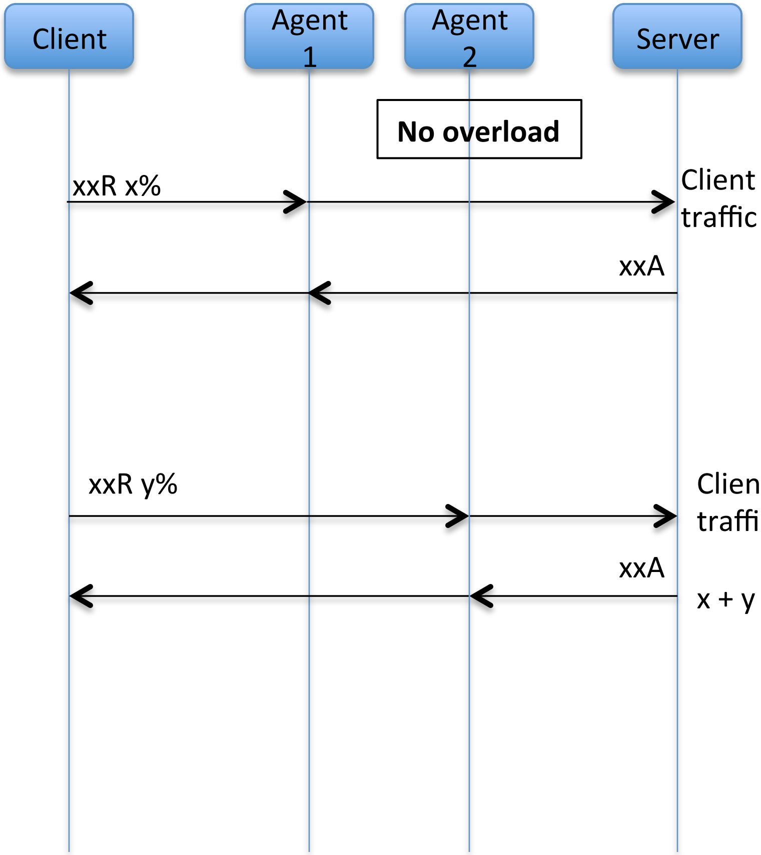
Agent Overload
Multiple Agents

Architecture



- Client has active connection to both agent 1 and agent 2
- Client shares the load between the two agents
- The load distribution mechanism is local policy to the client

Agent Overload
Multiple Agents

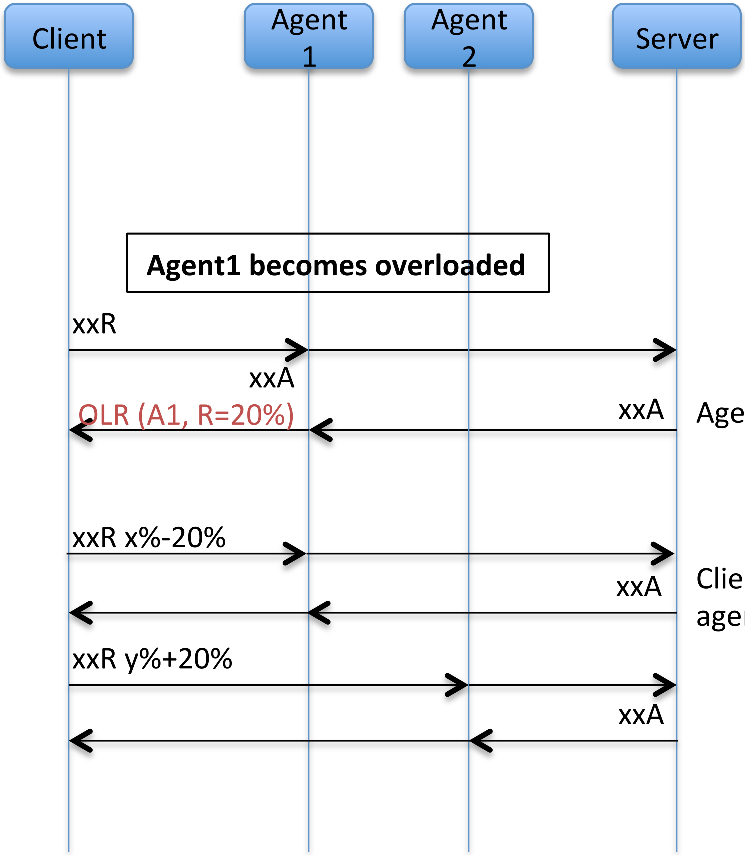


Client sends x percent of traffic through agent 1

Client sends y percent of traffic through agent 2

x + y = 100%

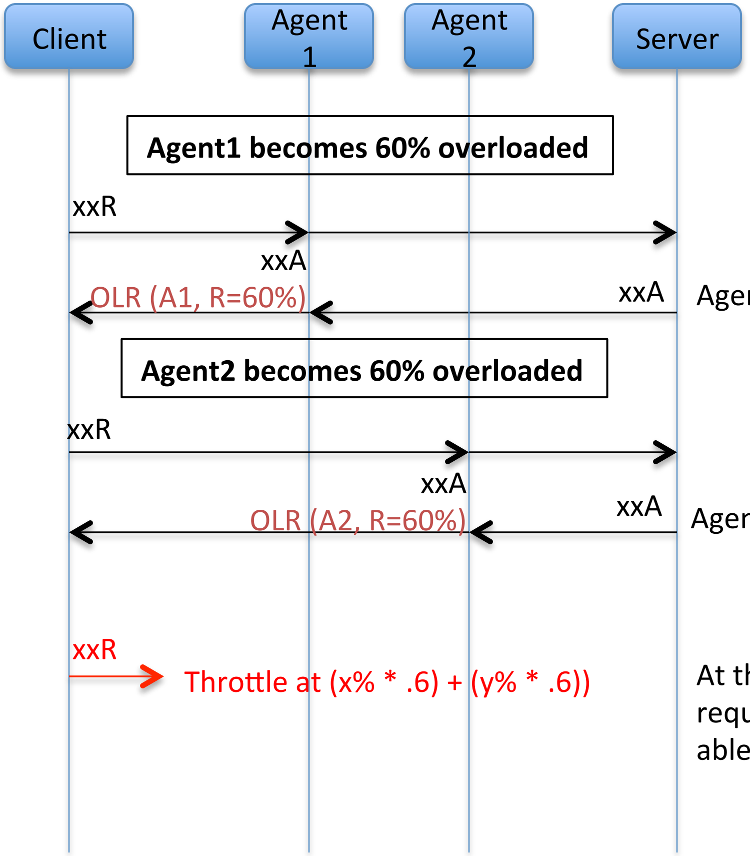
Agent Overload
Multiple Agents



Agent OVL report contains requested reduction

Client adjusts distribution of load between agents based on the agent overload report

Agent Overload
Multiple Agents


Client
Agent 1
Agent 2
Server
Agent1 becomes 60% overloaded
xxR
xxA
OLR (A1, R=60%)
xxA
Agent2 becomes 60% overloaded
xxR
xxA
OLR (A2, R=60%)
xxA
xxR
Throttle at (x% * .6) + (y% * .6))

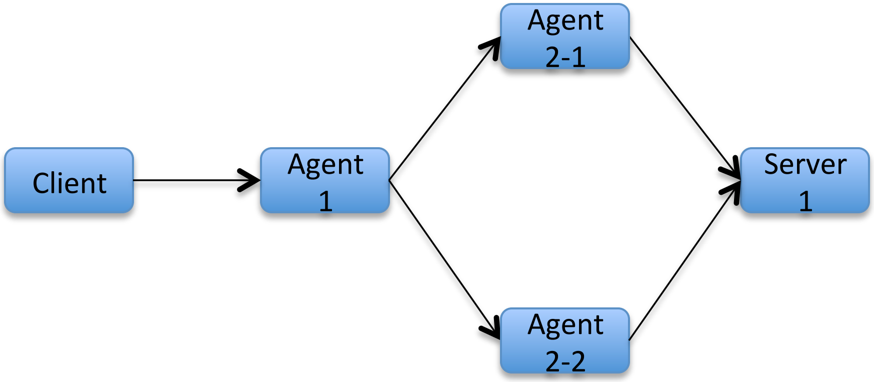
Agent OVL report contains requested reduction

Agent OVL report contains requested reduction

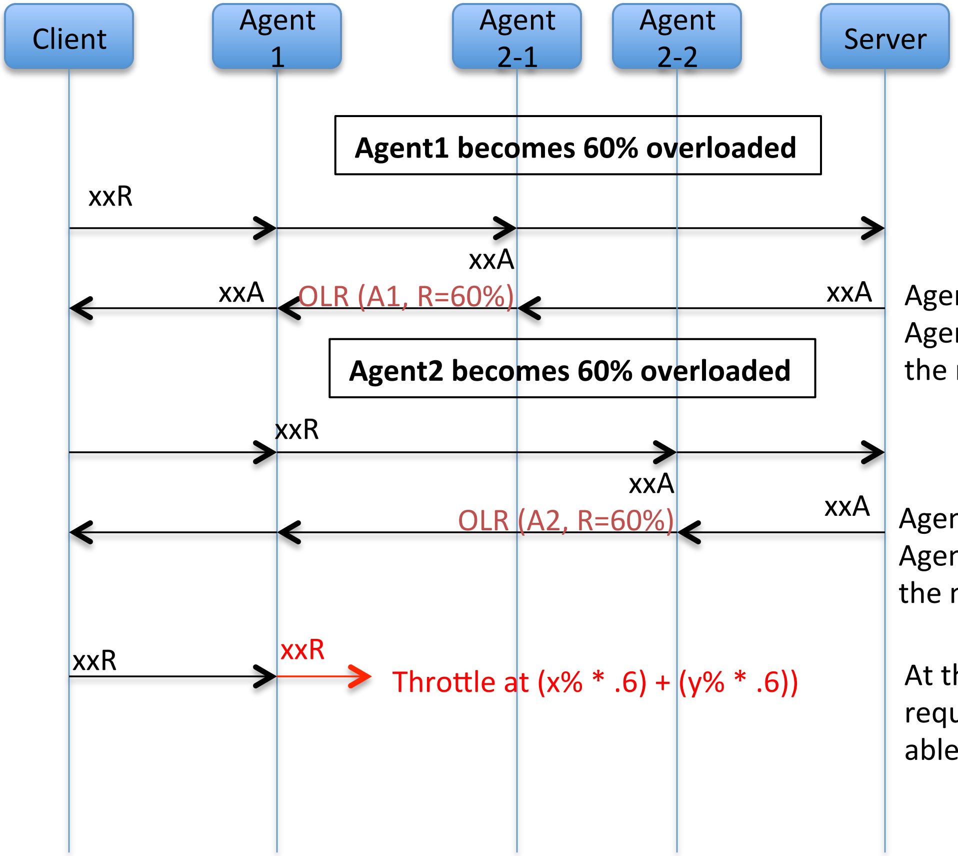
At this point the combined agent overload requires throttling to a level the agents are able to handle

Agent Overload UC4
Agent Chain

Architecture



Agent Overload
Agent Chain



Agent OVL report contains requested reduction
Agent 1 acts on the report and removes it from the message

Agent OVL report contains requested reduction
Agent 1 acts on the report and removes it from the message

At this point the combined agent overload requires throttling to a level the agents are able to handle