envoycon

NORTH AMERICA

Making Envoy Resilient to Sudden Increases in Load

envoycon

NORTH AMERICA

November 12, 2024 | Salt Lake City

Yan Avlasov

Staff Software Engineer

Google

Boteng Yao

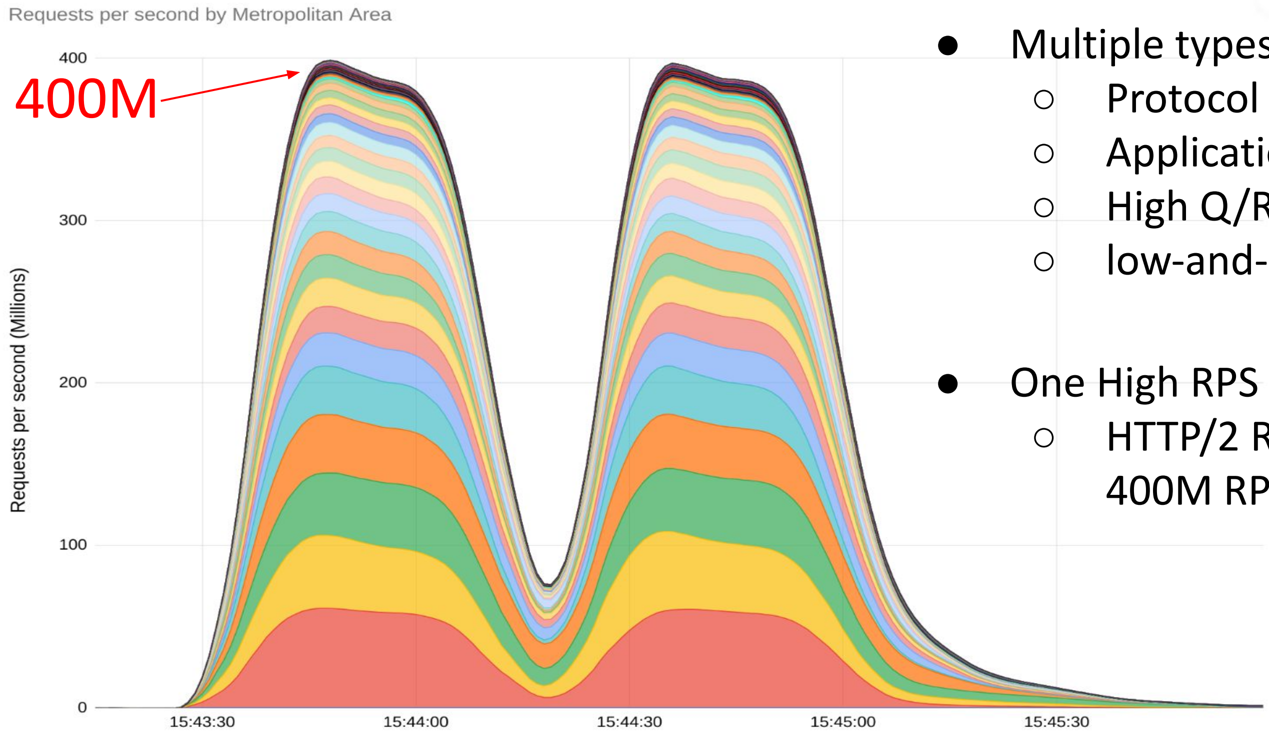
Software Engineer

Google

Agenda

- What is DoS attack
- HTTP2 Rapid Reset attack
- Effects on Envoy
- Mitigations
- Overload manager and the system

What is DoS Attack

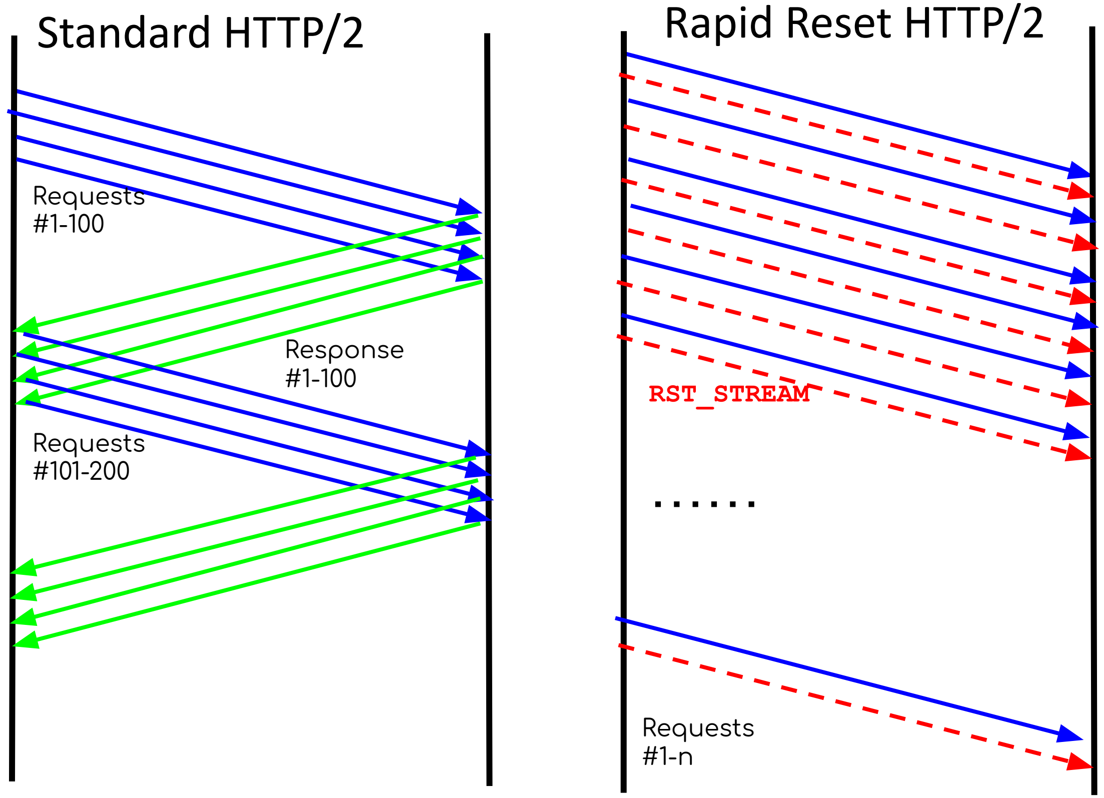


Source: https://cloud.google.com/blog/products/identity-security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps

- Multiple types of Denial of Service(DoS)
  - Protocol level
  - Application level
  - High Q/RPS
  - low-and-slow

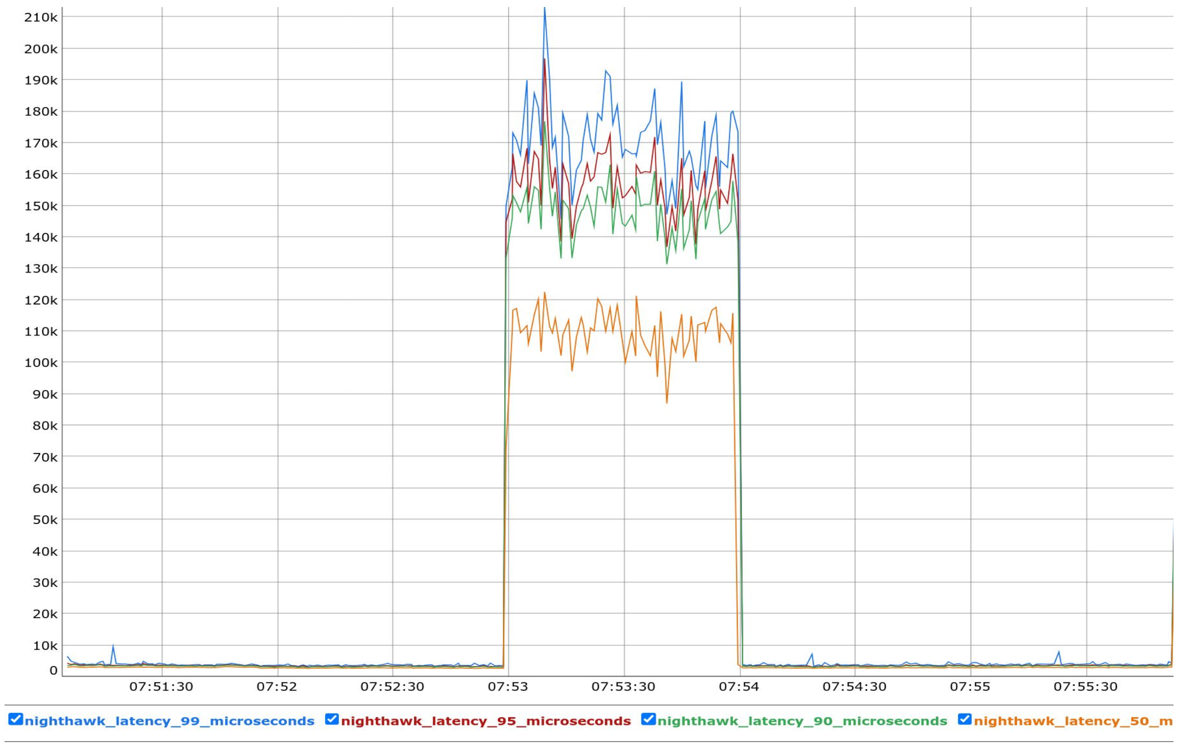
- One High RPS example:
  - HTTP/2 Rapid Reset reached up to 400M RPS globally.

Internals of HTTP/2 Rapid Reset Vulnerability



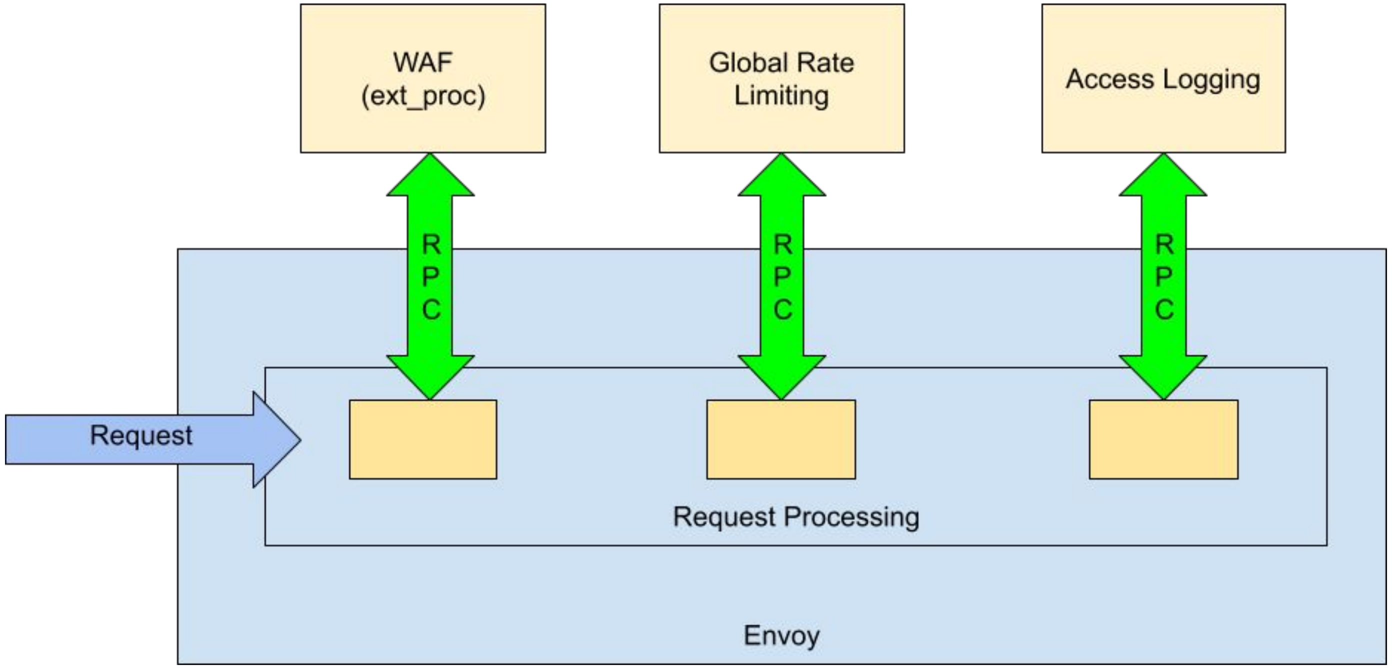
1. Quickly create a lot of requests.
2. Immediately delete them through `RST_STREAM`
3. Repeat step 1
4. `MAX_CONCURRENT_STEAM` will not work here
   a. Account open and half close stream status

Impact of CPU Exhaustion of Response Latency



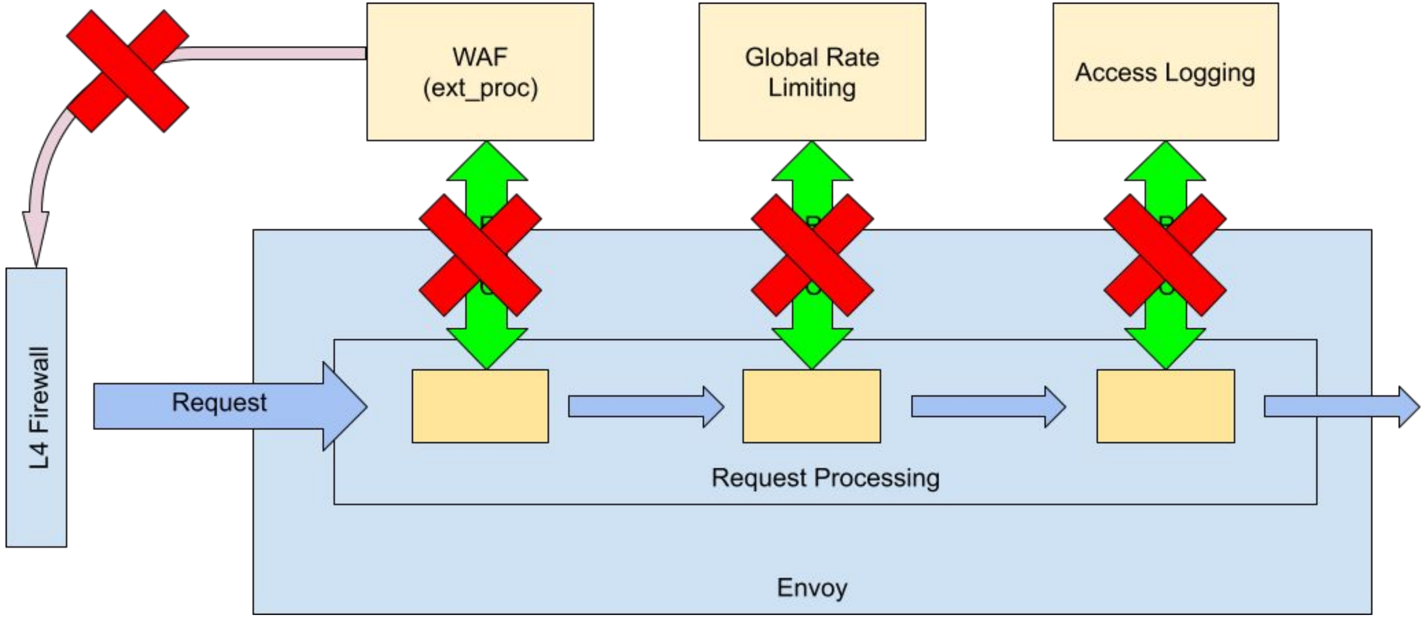
- 200ms of response latency from a single Rapid Reset connection for a single worker thread Envoy.
- Latency impact is almost linear with the number of rapid reset connections.

From Bad to Worse



- Response latency of sidecar services is also impacted.

From Bad to Worse



- Requests to sidecar services start to time out.
- Some side-car services are configured fail-open

Countermeasures

- Improve fairness of sharing CPU across client connections.
  - **http.max_requests_per_io_cycle**
    - - limit number of requests processed from single TCP segment.

- Specific to Rapid Reset - detection of abusive connections that frequently reset new requests.
  - **overload.premature_reset_total_stream_count**
  - **overload.premature_reset_min_stream_lifetime_seconds**

Future Improvements

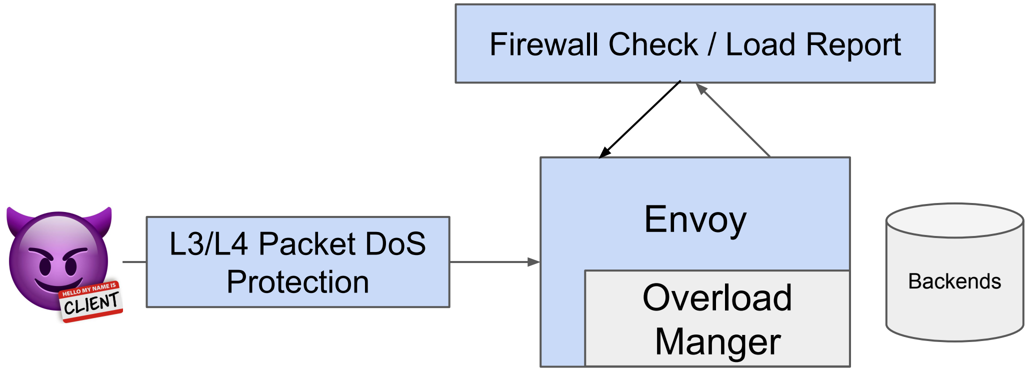
- CPU accounting per connection
  - Allow Overload Manager to throttle or close costly connections.
- Dynamically scaling worker count in response to load spikes.

Check that external authorization does not fail open.

Highlight the Role of Overload Manger

- A large scale system can be scaled **Horizontally** and **Vertically**
  - It could take **O(Mins)** to scale horizontally.
  - We need to make sure our system healthy to scale, especially at beginning.
    - Load report, DoS feedback, etc.
  - The time when overload manger will take effect in **O(ms/s)**

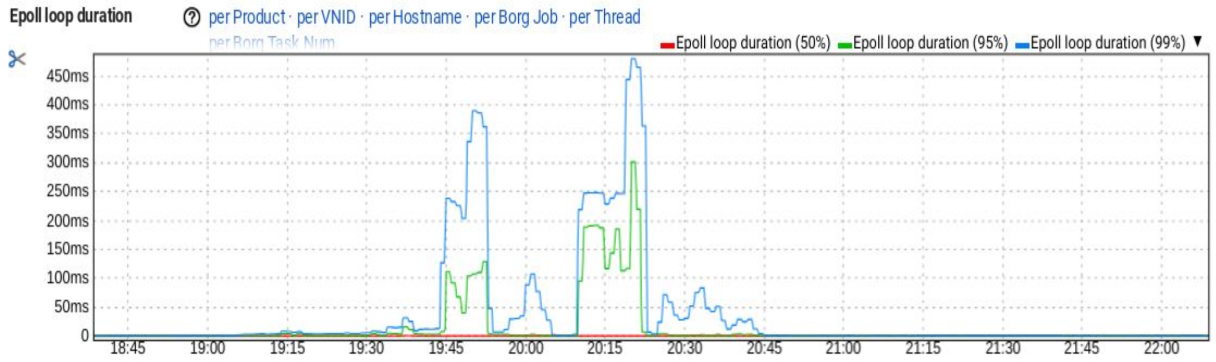
Priority	Action	Time
1	Autoscaling of Envoys	O(mins)
2	Overload Manager Actions Load Shedding	O(ms/s)

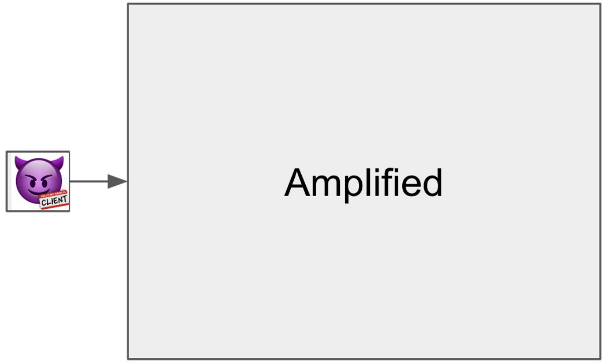


Sudden traffic spike can make LBs OOM

- Sudden increase of traffic can make CPU saturated
- And then it can continue to lead Out-of-Memory - **crashes**!
    - Event queue length increases due to saturated CPU
        - Delay overload manager actions
    - Small requests can amplify the memory effect in our LBs as well!
        - Cost is low for clients
        - Usually Envoy needs more work to process requests

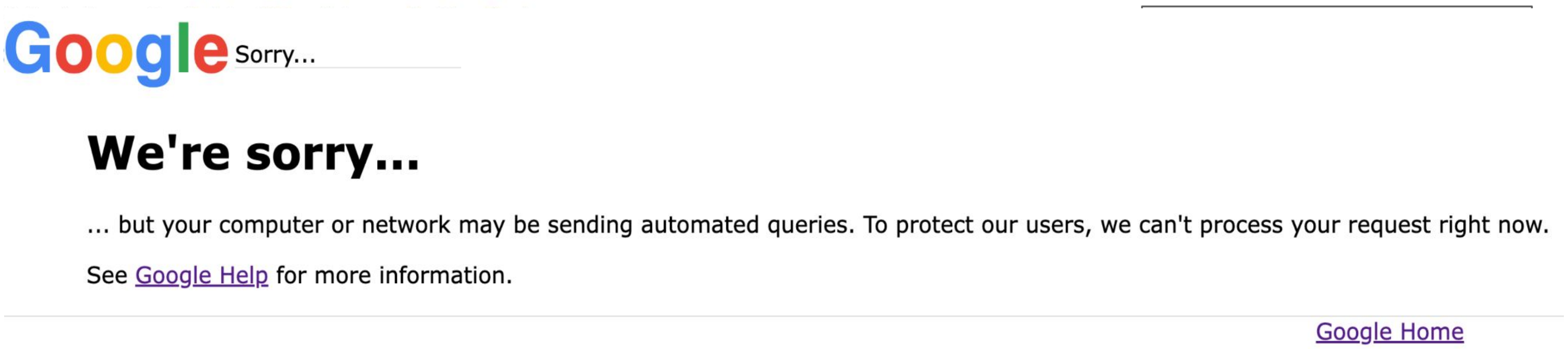
          Buffer / External RPC / Large Response





Sudden traffic spike can make LBs OOM

- Sudden increase of traffic can make CPU saturated
- And then it can continue to lead Out-of-Memory - **crashes**!
  - Event queue length increases due to saturated CPU
    - Delay overload manager actions
  - Small requests can amplify the memory effect in our LBs!
    - Cost is low for clients
    - Usually Envoy needs more work to process requests
      Buffer / External RPC / Large Response


Google Sorry...

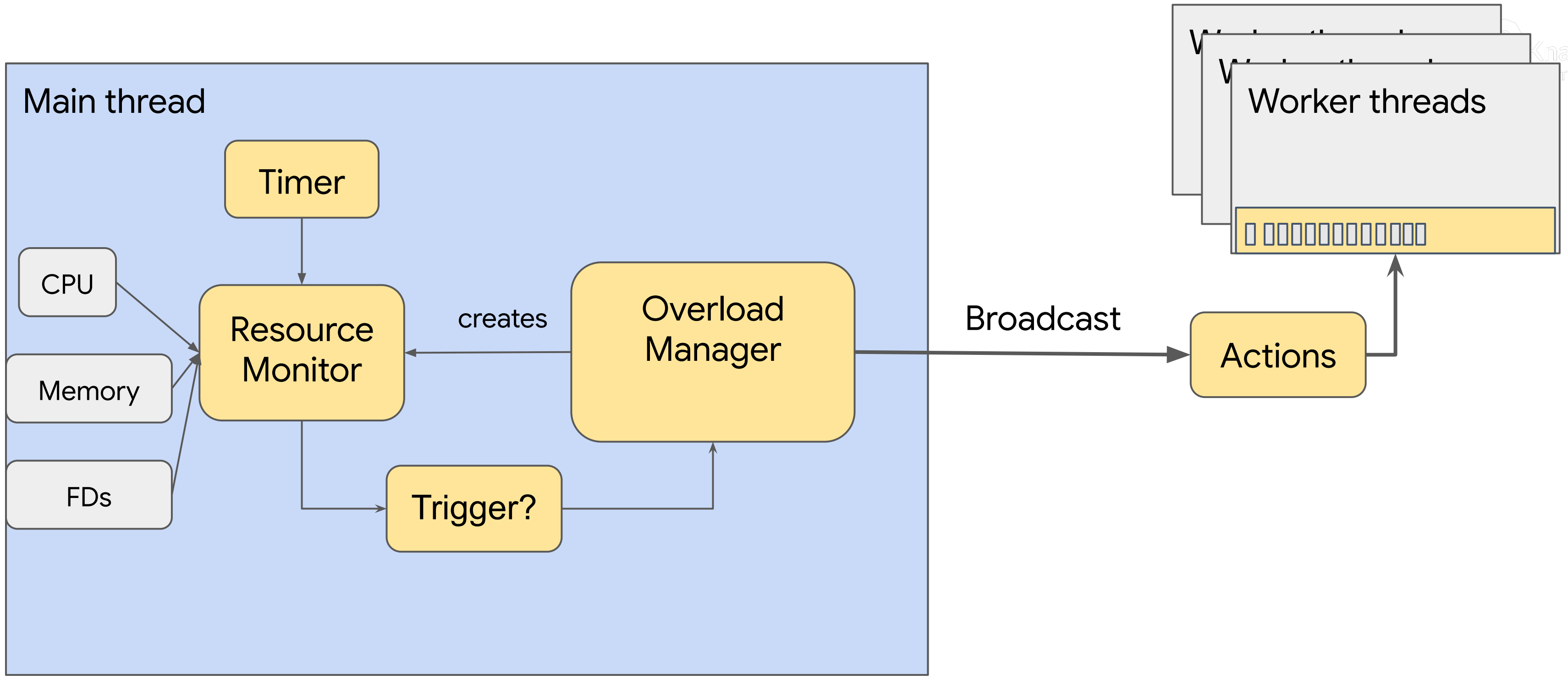
We're sorry...

... but your computer or network may be sending automated queries. To protect our users, we can't process your request right now.

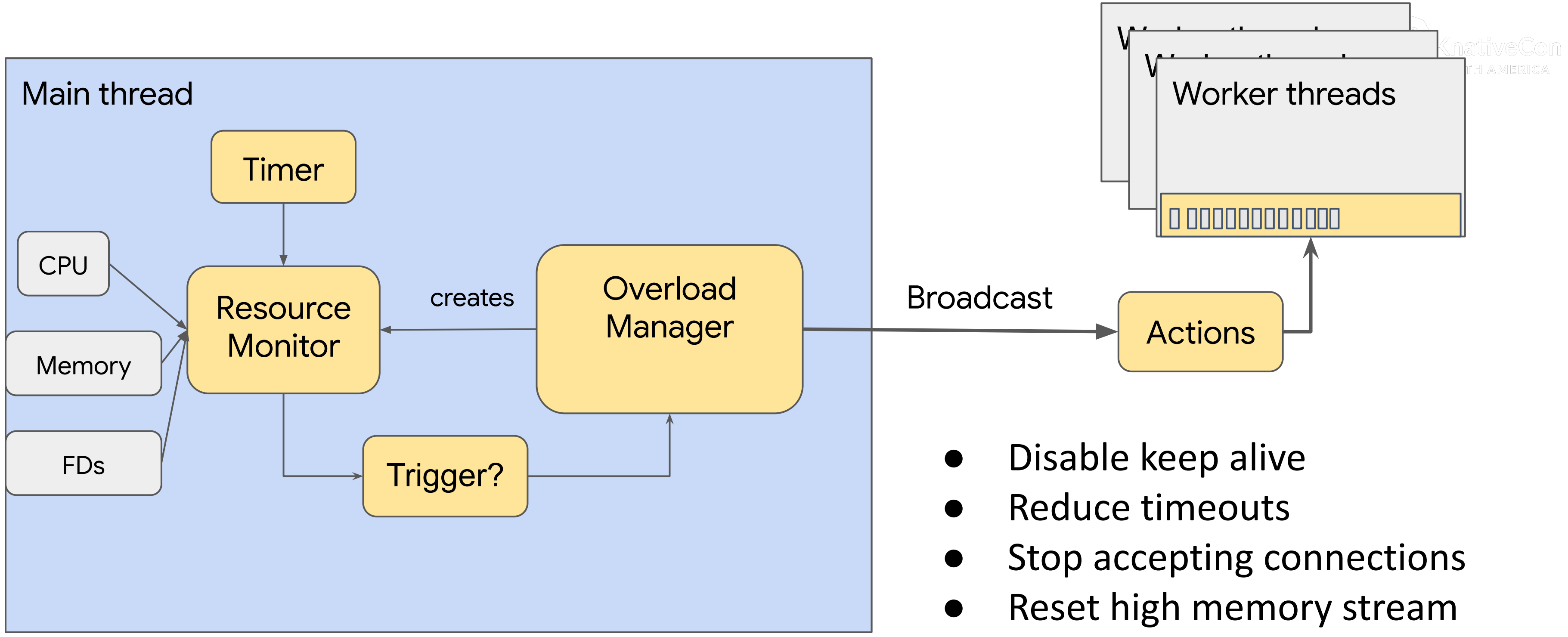
See Google Help for more information.

Google Home

“Traditional” Overload Manager Actions

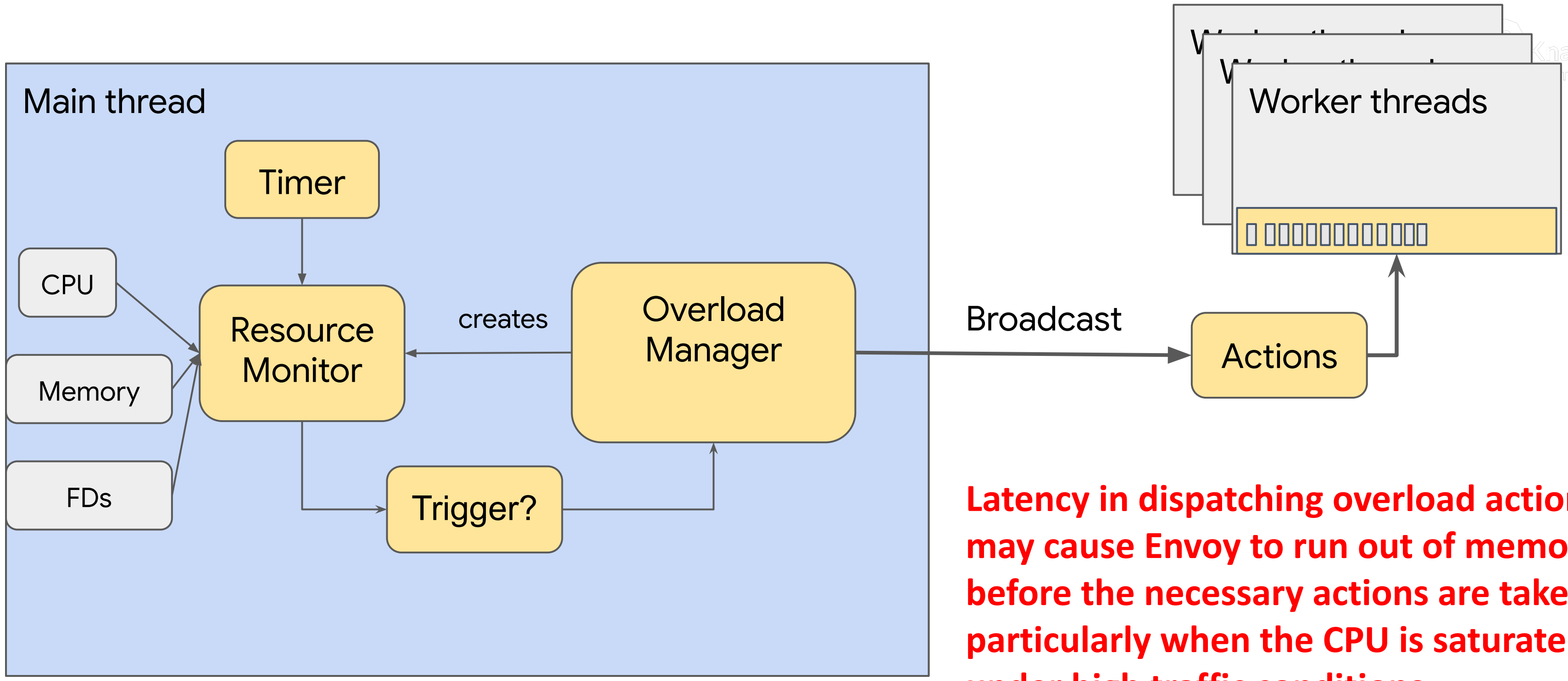


“Traditional” Overload Manager Actions



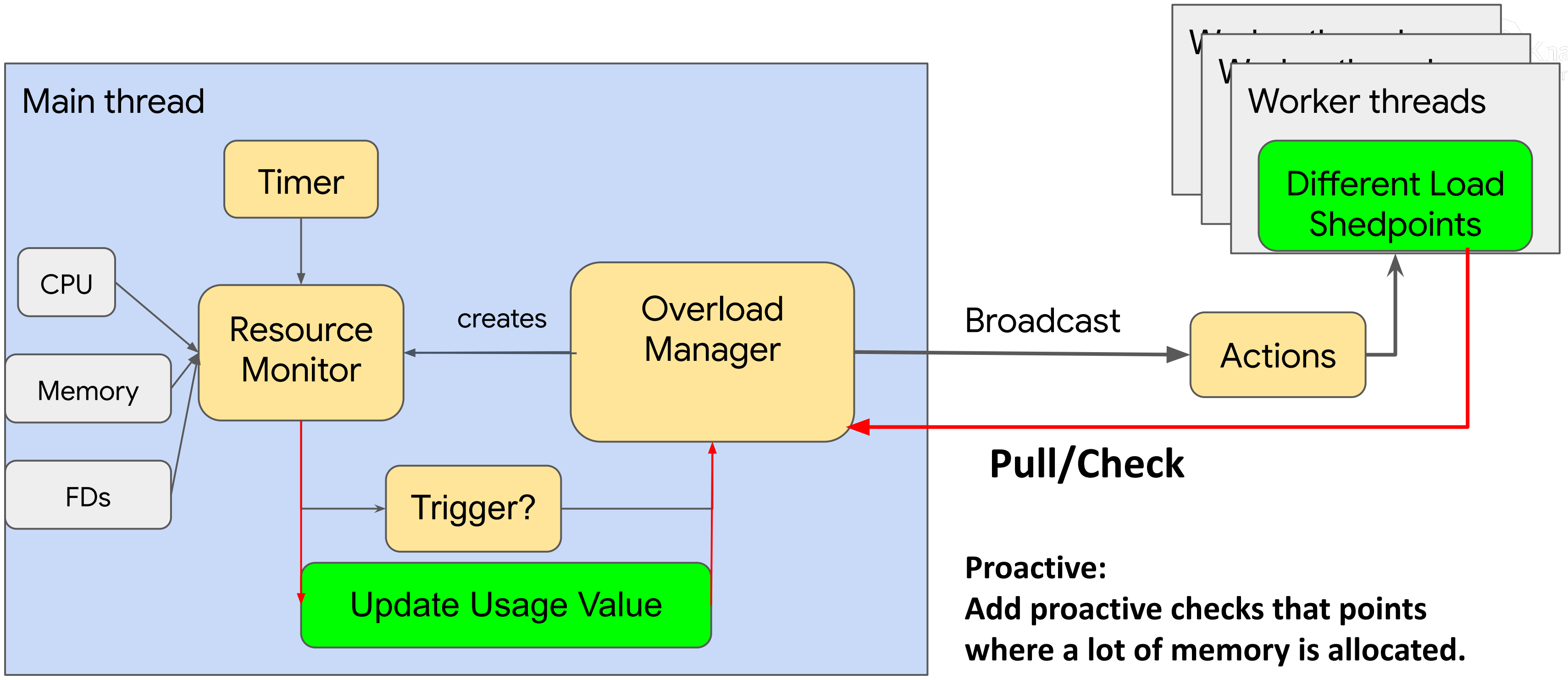
- Disable keep alive
- Reduce timeouts
- Stop accepting connections
- Reset high memory stream
- Shrink heap

“Traditional” Overload Manager Actions



Latency in dispatching overload actions may cause Envoy to run out of memory before the necessary actions are taken, particularly when the CPU is saturated under high traffic conditions.

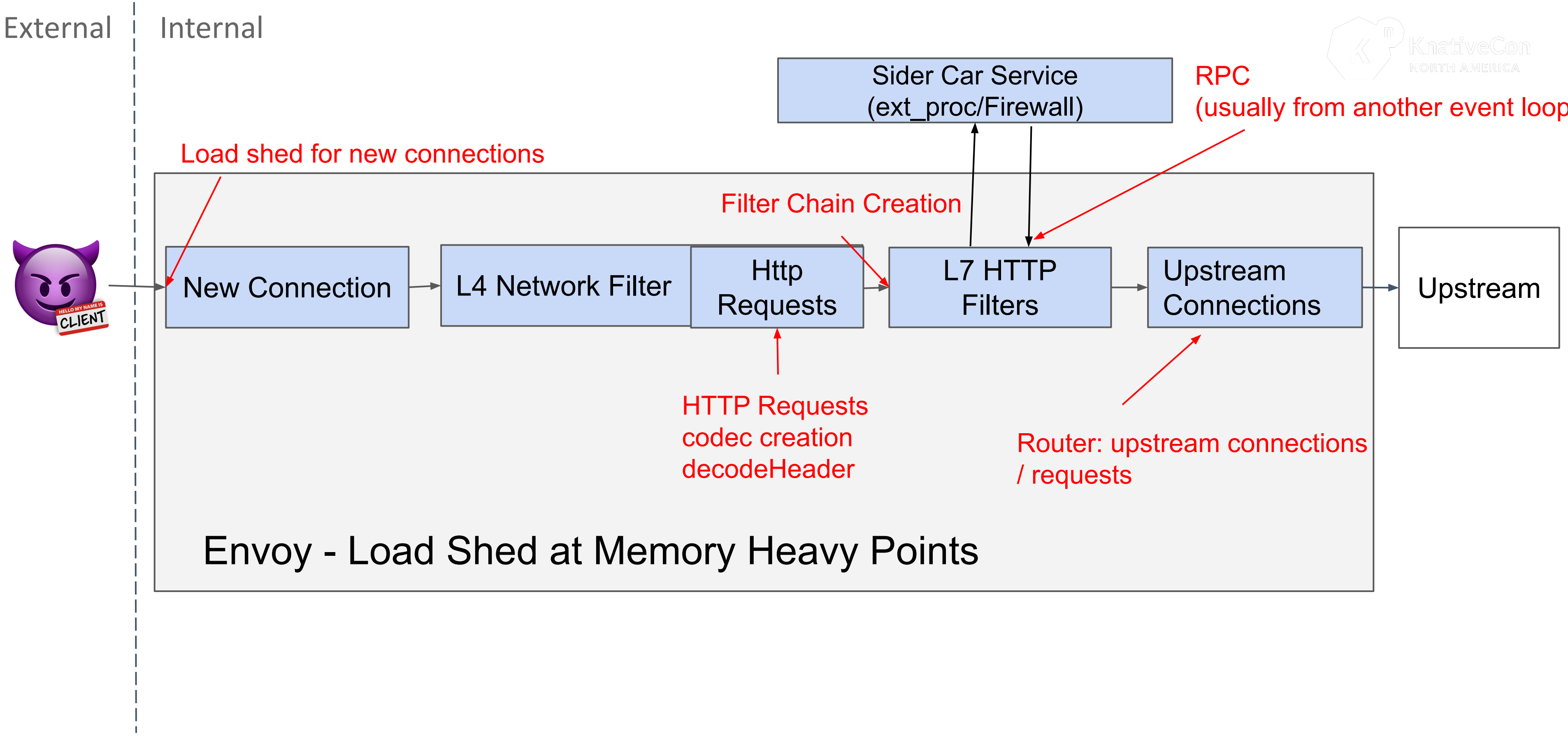
“Proactive” Overload Manager Thread Model



Pull/Check

Proactive:
Add proactive checks that points where a lot of memory is allocated.

Proactive load shed points



External
Internal
Sider Car Service (ext_proc/Firewall)
RPC
(usually from another event loop
Load shed for new connections
Filter Chain Creation
New Connection
L4 Network Filter
Http Requests
L7 HTTP Filters
Upstream Connections
Upstream
CLIENT
HTTP Requests codec creation decodeHeader
Router: upstream connections / requests
Envoy - Load Shed at Memory Heavy Points

Load shed points in Envoy

1. tcp_listener_accept
2. http_connection_manager_decode_headers
3. http1_server_abort_dispatch
4. http2_server_go_away_on_dispatch
5. hcm_ondata_creating_codec
6. Http_downstream_filter_check
   a. Http decoder filters
   b. Router

More info can be found here
https://www.envoyproxy.io/docs/envoy/latest/configuration/operations/overload_manager/overload_manager#overload-manager

@kbaicho, @boteng

Examples to config overload manager

```
refresh_interval:
  seconds: 0
  nanos: 250000000
```

```
resource_monitors:
  - name: "envoy.resource_monitors.container_memory"
    typed_config:
      "@type": type.googleapis.com/envoy.extensions.resource_monitors.container_memory
```

```
loadshed_points {
          name: "envoy.load_shed_points.tcp_listener_accept"
          triggers {
            name: "resource_monitors.container_memory"
            scaled {
              scaling_threshold: 0.8
              saturation_threshold: 0.9
            }
          }
        }
```

Additional notes to protect our system

- Consider configuring *`envoy.load_shed_points.tcp_listener_accept`*

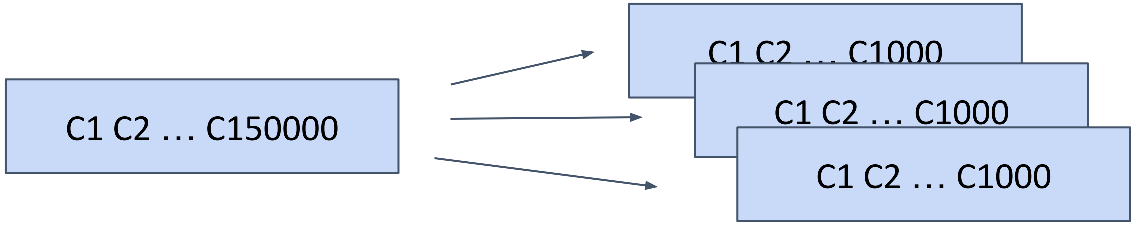
- Make sure GOAWAY can be triggered for HTTP/2 and HTTP/3
  - We'd like to see a good portion of GOAWAY for a better connection load balancing between LB containers rather than most of them are 503 local reply
    - *`http2_server_go_away_on_dispatch`*
    - *`http_connection_manager_decode_headers`*

- Consider dropping load around side streams
  - *`http_downstream_filter_check`*

Additional notes - Not Only Overload Manager - It is A System

- Set a default value for ***max_requests_per_io_cycle***
  - Mainly for HTTP/2 and HTTP/3 Multiplex Streams
  - To fairly use CPU cycles
  - Limit to **15 - 50** via Load test

- Set a value to ***max_connections_to_accept_per_socket_event***
  - Envoy listener will by default accept many new connections as possible
  - It can be up to **15K** or more in a single I/O event
    - Envoy can run out of memory before the overload manager has a chance to react.
  - Increasing frequency of resource monitor polling is not sufficient with more cost.
  - Recommend **50 - 1000** with ignorable impact on the tail latency via Load test!
    - Metrics: `connections_accepted_per_socket_event`



Additional notes - Not Only Overload Manager - It is A System

- Examine the buffer size configuration
  - tls_inspector buffer size
    - - default 64KB -> reduce the amount of memory tls_inspector pre-allocates

  - Flow control as always
      - Listener limits
      - Cluster limits
      - H/2 Stream limits

  - Examine the customized buffer usage in your data-plane

References and Acknowledgments

- How it works: The novel HTTP/2 ‘Rapid Reset’ DDoS attack
- Envoy Rapid Reset CVE-2023-44487
- Google mitigated the largest DDoS attack to date, peaking above 398 million rps
- Envoy Overload Manager Doc
- Lightning Talk: Protecting Envoy: Overload Manager by @kbaichoo

Thank you all for making Envoy more resilient

yanavlasov@, kbaichoo@, antoniovicente@, nezdolik@, alyssawilk@, htuch@, akonradi@, eziskind@, esmet@, and all others

Thank you & QA