

Game Server Networking with Envoy

Christopher M Luciano
Advisory Software Engineer at IBM
cmluciano@us.ibm.com

IBM Developer

\$ bash whoami

- Christopher M Luciano
- Advisory Software Engineer @ IBM
 - Envoy
 - Istio
 - Kubernetes
- Contact
 - Github: github.com/cmluciano
 - Slack: [cmluciano](#)
 - Twitter: [@cmluciano](#)
 - Speakerdeck: [speakerdeck.com/cmluciano](#)



/msg drag0n

- Tank Bonechewer realm
- ADC in HoTs and LoL
- Protoss for life

Agenda

- Game Server Networking
- Envoy Basics
- Envoy Uses in Gaming

IBM Developer

Game Server Networking Goals

Scale

LAN PARTY!!!

- Several clients connected over a Rube Goldberg like web of spare ethernet cables
- One player designated as host
- Players rely on host to have a good connection
- Host migration to mitigate host dropping

Dedicated Game Server

- 1 game engine server (may be multiple on one machine)
- 10s to 100s of players connecting to game engine
- 100-200 ms latency max
 - 30 – 50 ms optimal

Massive Multiplayer Worlds

- Could be shards of world spanning multiple servers
- Transitioning should be seamless
- Thousands spread throughout a virtual world

Performance

- Scale factor of message attributes
- Complexity of event types
 - Replicating data state
- Protocol choices
- Prioritizing data

Availability

Security

Fleet Upgrades

Speech Feature Rollouts



<https://blogs.unity3d.com/2018/02/20/bringing-the-power-of-ai-to-developers-with-the-ibm-watson-unity-sdk/>

Speech Feature Rollouts



<https://assetstore.unity.com/packages/tools/ai/ibm-watson-unity-sdk-108831>

Speech Feature Rollouts



Envoy

- Intelligent proxy deployed as a sidecar
- Intercept & manage network traffic
- Security/Identity
- TLS termination
- Low memory footprint
- Language Agnostic



Envoy Communication



Listener

- One to many number of listeners per Envoy process
- Proxies event when connection is made to listeners
- Configured with network or listener filters
- Listener binding
 - Freebind
 - Transparent
- TCP only (UDP soon)

Clusters

- Collection of similar hosts for Envoy connections
- Determined through various service discovery systems
- Can be configured either from a static list or dynamic API
- Warmed on initial boot

Filters

- Listener
 - Original Destination
 - TLS Inspector
- Network (Layer 3 & 4)
 - Rate limiting
 - Mongo
 - TCP Proxy
- HTTP
 - Fault injection
 - Router

Custom Network Filters

- Read filter
 - Triggered on data receive from downstream
- Write
 - Triggered before sending to downstream
- Read/Write
 - All the above

Rate Limiting

- Network rate limit filter
 - Called for every new connection

Stat	Description
total	Total requests to service
over_limit	Total connections > limit
cx_closed	Connections closed due to limits of rate limit service
failure_mode_allowed	Total requests allowed through from failure_mode_deny

Overload Manager

- Protects Envoy under distress
- Helpful for noisy neighbor prediction
- Built-in support for fixed heap
 - Stop processing packets
 - Disable protocol features
- Could be extended to prioritize packets?

Additional Built-in Features

- Websocket support
- gRPC with protocol buffers
 - Useful integrations with game engines
- Service discovery
 - Static
 - Strict DNS
 - Logical DNS
 - Original Destination
 - Endpoint Discovery Service (EDS)

Endpoint Discovery Service

```
{  
  "cluster_name": "...",  
  "endpoints": [],  
  "policy": "..."  
}
```

Envoy Fleet Management

- Draining connections
 - POST to /healthcheck/fail
 - Listener modification with Listener Manager Service
 - Health checkers available for HTTP
 - Outlier detections for passively removing unhealthy hosts from the load-balancer pool
- Hot restart
 - Envoy full reloads w/o dropped connections
 - Preserves gauge stats across the two processes
 - Utilize Unix domain sockets for RPC communication
 - New process fully initialized then starts asking old process for copies of listen sockets

General Cases for Game Platforms

General Platform Uses

- Access logging
 - Determine player position
 - Gather metadata that's useful
- Protocol level stats
 - Players connection speeds
 - Player's previous values
- Better Matchmaking
 - Match based on connection speeds
 - Locality
 - Separate for Game types and Events

QA testing

- Poor connections
 - [Fault injection](#)
- Resource failures
- Host migrations

Future Envoy Enhancements

- UDP
 - [Envoyproxy/envoy#492](#)
- QUIC
 - [Envoyproxy/envoy#1193](#)
 - [ValveSoftware/GameNetworkingSockets](#)
- Improved performance for large clusters
- WebAssembly support
- Mobile build targets & uses

Questions

- Christopher M Luciano
- Contact
 - cmluciano@us.ibm.com
 - Github: github.com/cmluciano
 - Slack: cmluciano
 - Twitter: @cmluciano
 - Speakerdeck: speakerdeck.com/cmluciano
- [Mesh expansion across IKS & ICP](#)
- [Connecting private and public clouds](#)
- [Basic envoy101 examples](#)