Computing-Aware Traffic Steering (CATS)

CATS Chairs
Adrian Farrel (adrian@olddog.co.uk)
Peng Liu (liupengyjy@chinamobile.com)

IETF-117 – San Francisco – July 2023
History

• Some drafts in 2020 on *Dynamic Anycast (Dyncast) for Compute First Networking*

• Renamed *Compute Aware Networking (CAN)*
  • BoF at IETF-113 online and in Vienna (March 2022)
  • Presented at RTGWG and TSVWG at IETF-114 in Philadelphia (July 2022)
  • Mailing list for CAN (created October 2022)
    • [https://www.ietf.org/mailman/listinfo/can](https://www.ietf.org/mailman/listinfo/can)
  • BoF at IETF-115 in London (November 2022)
    • Clear interest in the work, but very unfocussed meeting
  • A pile of Internet-Drafts got written
    • [https://datatracker.ietf.org/doc/search/?&name=computing-aware&sort=&activedrafts=on&olddrafts=on](https://datatracker.ietf.org/doc/search/?&name=computing-aware&sort=&activedrafts=on&olddrafts=on)
  • Draft charter for a CAN working group written by proponents
    • Reviewed by the IETF community

• Renamed to *Compute Aware Traffic Steering (CATS)*
  • Slightly more accurate name, avoids name conflict with Controller Area Networking
  • Area Directors decided to form a working group without further BoFs
    • First meeting at IETF-116 in Yokohama (March 2023)
  • Old drafts renamed, and many new drafts posted
Charter Scope

• There are often multiple service instances
  – Geographically distributed to multiple sites
  – A single site may support multiple instances of a service

• The services provided on computing platforms and are generically referred to as "compute services".

• The performance experienced by clients depends on:
  – Network metrics such as bandwidth and latency
  – Compute metrics such as processing, storage capabilities, and capacity

• How can the network edge steer traffic between clients of a service and sites offering the service?
But before we start - Groundwork

- CATS is not chartered to work on solutions
- Do the groundwork first
  - Problem statement
  - Use cases
  - Requirements
  - Framework and architecture
  - Metrics for Compute and requirements for distribution
  - Analyse usefulness of existing protocols and tools
- Only one RFC explicitly in charter
  - CATS Framework and Architecture
Current Status

• It’s early days
  – But some of the work has been around for a while
• Just adopted a draft for problem statement, use cases, requirements
• Meeting (tomorrow, Thursday, 9.30am)
  – Terminology
  – Use cases
  – Requirements
  – Metrics
So, what’s it all about?

• Traffic targets a service that uses computing resources
  • The local network edge selects a remote edge that provides access to one or more instances of that service
    • May select a specific service instance
    • Note that the application and host do not participate in this choice
  • The local network edge steers the traffic to the remote edge
    • Network might also be traffic engineered
• The choice of instance depends on:
  • Service requirements
  • Capabilities of server
  • Load on server
  • Capabilities of network
  • Load on network
What are the Use Cases?

- Still firming this up
- Applications that have been mentioned...
  - Real-time image capture and processing
  - Interconnected and event-aware “smart cars”
  - Multi-player game servers
  - Networked AR/VR
  - Holographic presence conferencing
  - Digital twin
  - SD-WAN
  - Further uses cases being discussed
- Objective is not a complete list
  - We want a few compelling use cases
- Differences and commonalities
  - The use cases all have different network and service requirements
  - All need to move data, have it processed, and get a response
What are the Requirements?

• This is also work in progress – just a summary overview

• Mark traffic for a “group of service instances”
  • Anycast addresses are suggested

• Collect information from the network (topology and metrics)
  • Already do this, but may need supplements for (e.g.) latency

• Collect information from service instances
  • Service locations (membership of Anycast address group)
  • Service capabilities
  • Service location loading (metrics)

• We may also need to know:
  • Service demand requirements
  • A way of batching packets into service requests
First Draft Functional Architecture

Traffic steered to service instance

Collect information about the network

NOT CONSENSUS