Requirements and Gap Analysis of CATS

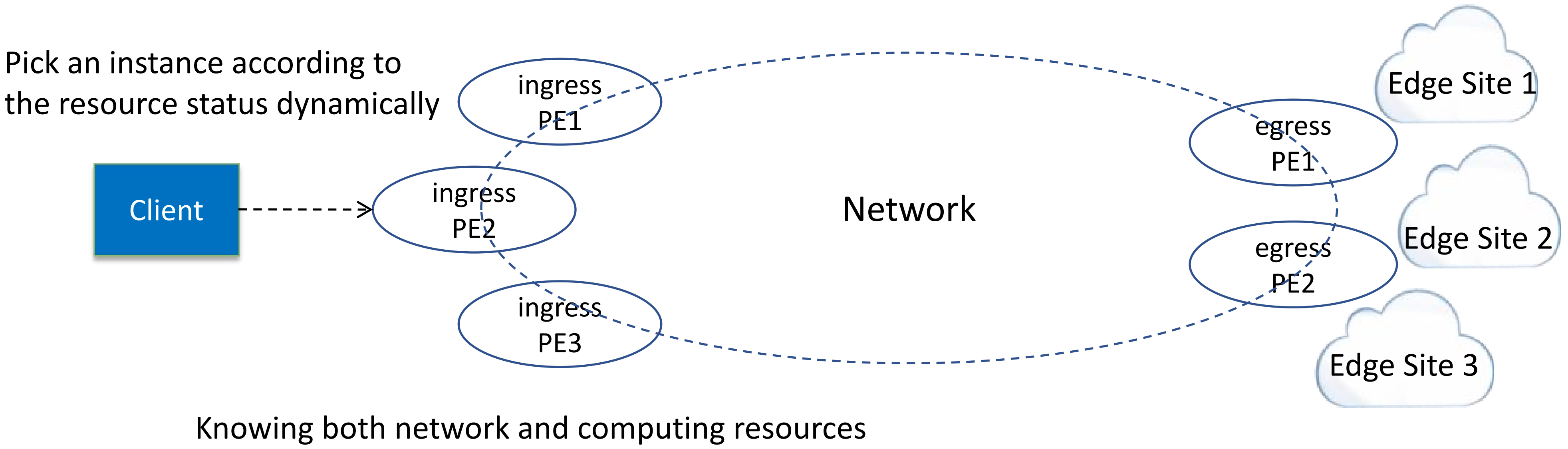
draft-yao-cats-gap-reqs-00

K.Yao, China Mobile
T. Jiang,China Mobile
P. Eardley
D. Trossen, Huawei
C. Li,Huawei
D. Huang,ZTE

CATS Requirements

Main Goals

- **Service equivalence/Multi instances: considering to access one of possibly many instances deployed across multiple sites**
- **Multi-metric: considering both the network and computing resource status**
- **Dynamic decision: select the appropriate computing resource dynamically**



Potential Main Requirements

- **Support access to the available edge sites dynamically**
  - Allow for making dynamic selections of service instances across multiple sites, based on suitable metrics
- **Provide both network and computing metrics for further use**
  - Computing resource related model for the metric
  - Control the rate of metrics distribution
- **Support effective computing resource representation and encapsulation**
  - Single index or multi-dimentional information for specific purpose
- **Support the session continuity and service continuity**
  - Functional equivalency
  - Preserve affinity in light of metric-based instance selection decisions
- **Preserve communication confidentiality**
  - Avoid unnecessary exposure of service as well as metric information

CATS Gap Analysis

Existing Solutions

There are existing solutions that utilize end point resource information for selecting server instances, such as DNS, ALTO, or Layer 4 & Layer 7 load balancers.

•Common to all those solutions is the **exposure and usage of network conditions or computing resource status** for decision making, …

•Together with the **use of an indirection approach** for determining the ‘best’ service instance, while...

•None of the existing solutions **integrate the computing resource conditions with the network conditions** for deciding the optimal paths.

This has implications upon supporting the **dynamicity** of service relations, the **exposure** of holistic compute/network metrics, as well as the **efficiency** and **complexity** of those solutions

Summary of Gap Analysis

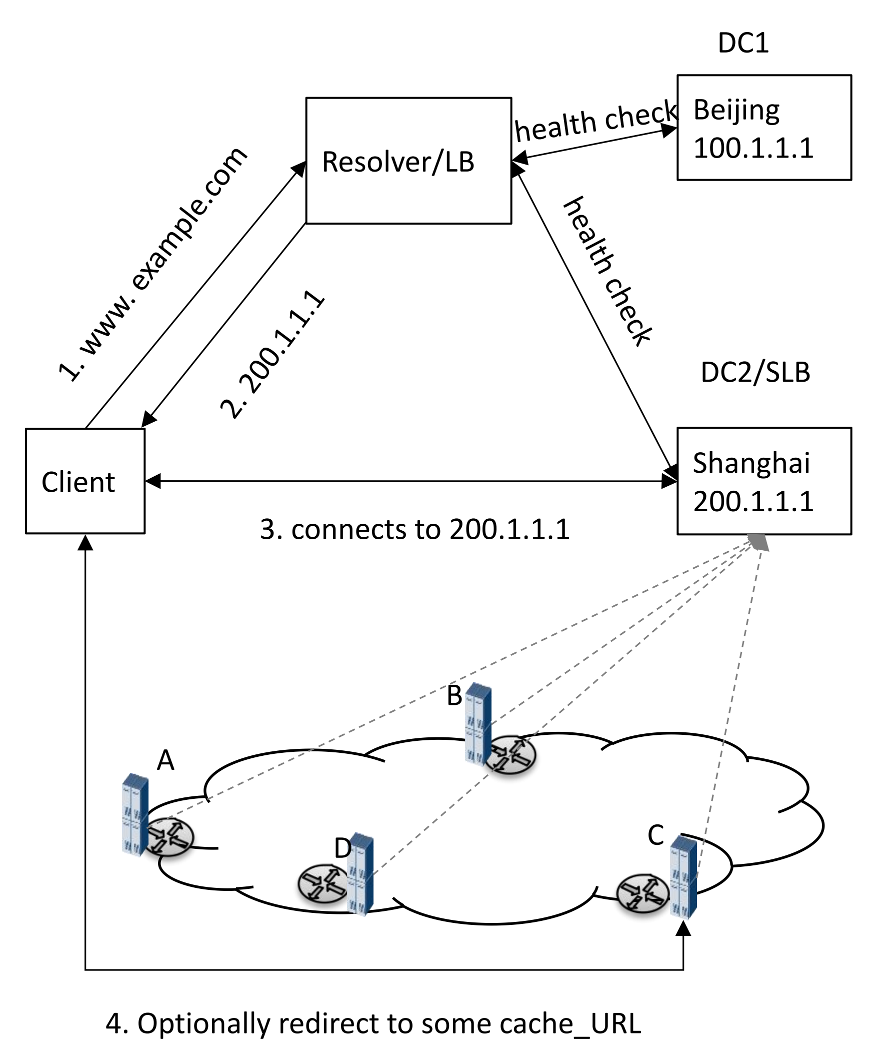
- **Dynamicity:** Existing solutions exhibit limitations in providing dynamic instance affinity
  - E.g., DNS is not designed for this level of dynamicity (i.e., minute level originally, client needs to flush the local DNS cache, frequent resolving may lead to overloading DNS)
- **Efficiency:** Existing solutions may introduce additional latencies and inefficiencies (e.g., more messages) in packet transmission, e.g., path stretch.
- **Complexity and Accuracy:** Existing solutions require careful planning for the placement of necessary control plane functions in relation to the resulting data plane traffic, which is difficult and may lead to the inaccuracy of the scheduling.
- **Insufficient metric exposure and use:** Existing solutions lack the necessary information to make the right decision on the selection of the suitable service instance due to the limited semantic or due to information not being exposed.

Next Step

- Refine the drafts upon comments and suggestions
- More comments, suggestions and contributions would be welcome.

Backup

Gap Analysis of Existing Solutions-DNS/GSLB



- **Early binding**: clients resolve IP address first and then send traffic.
  - If using the DNS entry cached at client, stale info may be used.
  - Often, resolver and LB are separate entities which incurs even more signaling overhead by needing to first resolve and then redirect to LB for final decision
  - Resolution is L7 or app-level decision making, i.e. DB lookup. Intended for control, NOT data plane speed!
- **Health check**: on an infrequent base, switch when fail-over
  - Limited computing resources at edge will change rapidly, while more frequent health check is prohibitive in cost
- **Load balance over DNS**: usually focused on edge server load first, then utilizing lowest latency routing to the selected server's IP address
  - Lacks combined consideration of load & latency for a better E2E experience
  - Problem of how to obtain necessary metrics for decision
  - DNS-SD[RFC6763] and Multi-cast DNS[RFC6762] could be used to discover the service, which might be extended to collect the computing information. However, in most cases, they are used in the LAN environment. Use in WAN faces the same DNS stale cache problem.

Gap Analysis of Existing Solutions - Load Balancer

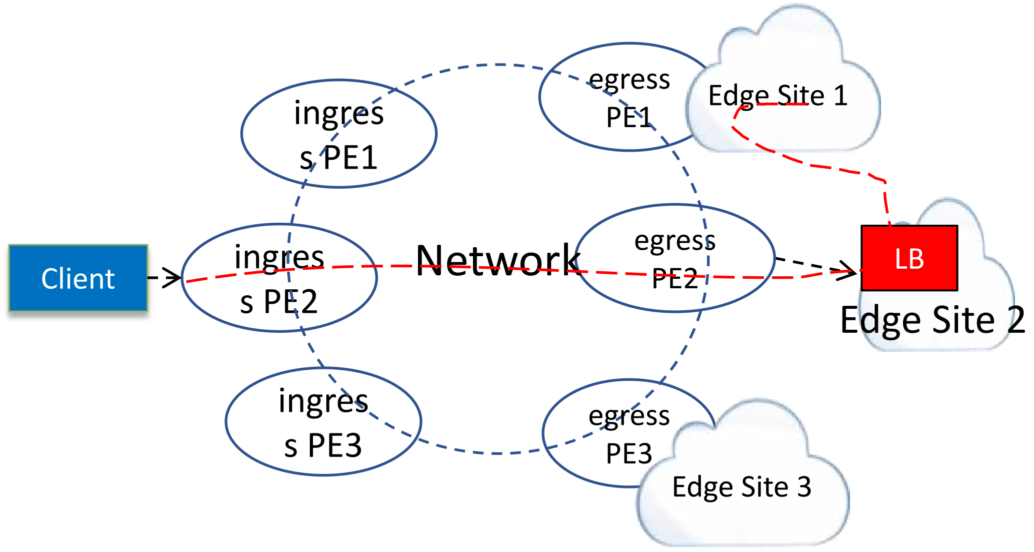
Single LB in a single site for all service instances:

Pros:

- easy deployment

Cons:

- Single point of failure at the LB
- *Path stretch*, i.e., the network path from the LB to server instances at other sites might not be optimal, e.g., the red dotted path



Each site has its own LB:

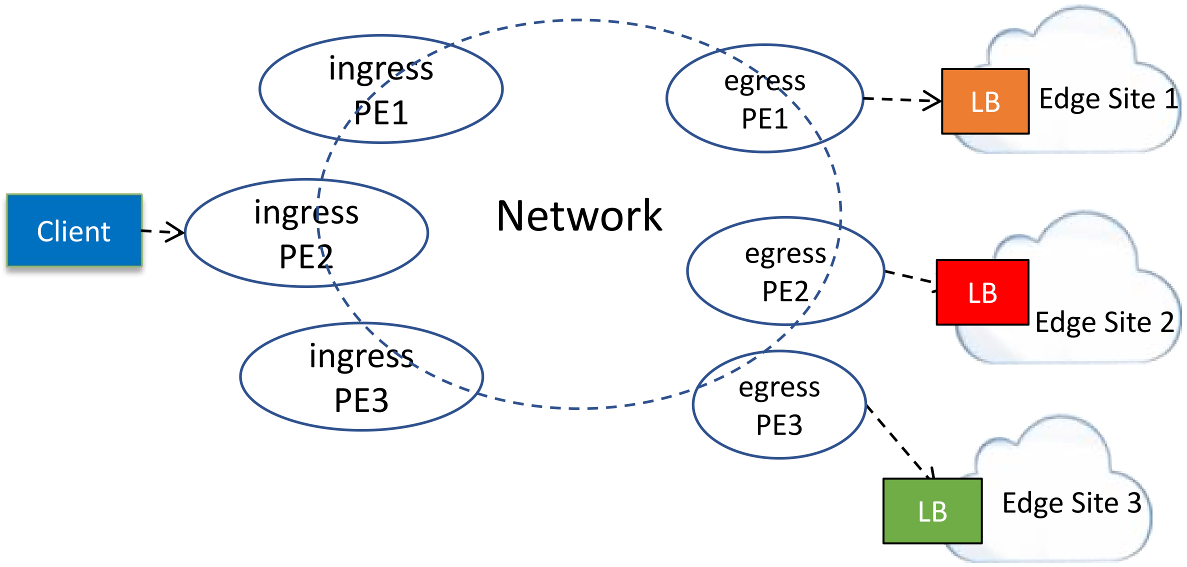
Pros:

-Easy deployment.

Cons:

-No load balance among multiple sites. Any site might be overloaded.

-Problem on selecting the 'best' site remains unsolved

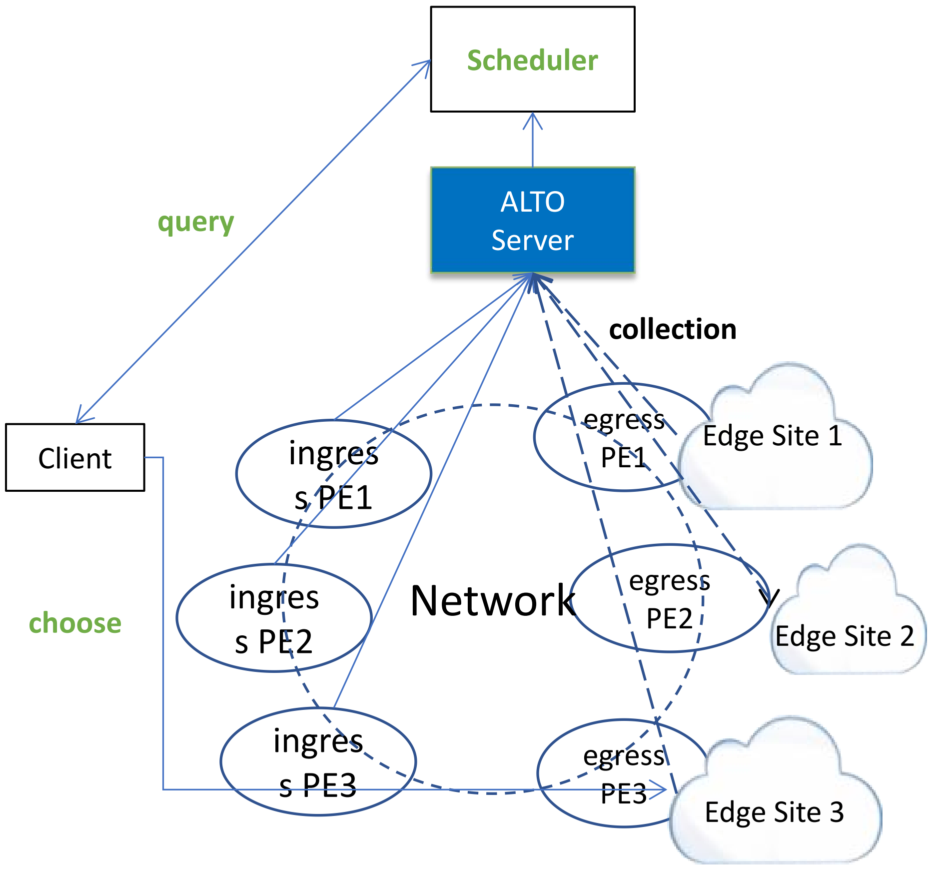


Gap Analysis of Existing Solutions - ALTO

ALTO [RFC7285] addresses the problem of selecting the 'optimal' service instance as an **off-path solution**, which can be seen as an alternative to DNS for tackling the problem space of CATS at the Application Layer.

The critical aspect is the signaling latency and the control plane load that a service-instance selection process may incur in both on-and off-path solutions. This in turn may impact the frequency with which applications will query ALTO server(s), especially in the mobile system.

As a result, off-path systems, e.g., ALTO, which are based on receiving replies for applications/services before traffic could be delivered, might not keep optimal or even valid after the handover.



Gap Analysis of Existing Solutions – Client-based solution

The individual destination, i.e., the network identifier for a service instance, must be known to the client a priori for direct service dispatching. While this may be viable for certain applications, it suffers from following issues:

•Unwanted service instance exposure: It may be undesirable for clients to learn all available service instance identifiers for reasons of Service Providers' being reluctant to expose their 'valuable' information to clients.

•Can not learn all network paths: Operators would not expose all available paths to clients, forcing clients to measure the network by itself.

•Scalability: The number of service instances and network paths may be very high.