Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements

draft-ietf-cats-usecases-requirements-07

K. Yao, China Mobile
LM. Contreras, Telefonica
H.Shi,Huawei
S. Zhang, China Unicom
Qing An, Alibaba Group

Updates Summary Since IETF122

- **Comments that have been addressed from -06 to -07 :**
  - Many thanks to Ines Robles's comprehensive review, comments have been addressed (mailing list thread June 10th)
  - Adrian's comments on R14(mailing list thread April 20th)
  - Comments from Paul (mailing list thread Feb 14th)
- **Other mailing list discussions**
  - Consolidation on merging new use cases and requirements(mailing list thread July 5th)
    - ISAC use cases from Carlos
    - Task Segmentation use case from Minh Ngoc Tran

Updates on Requirements

- **Requirements on Instance Selection(Sec 5.1)**
  - R1: MUST provide a dynamic discovery and resolution method for mapping a service identifier to one or more current service instance addresses, based on real-time system state.
  - R2: MUST provide a method to dynamically assess the availability of service instances, based on up-to-date status metrics (e.g., health, load, reachability).

➢ ***Add description on dynamicity***

- **Requirements on Metric Representation and Definition(Sec 5.2)**
  - R3: The implementations MUST agree on using metrics that are oriented towards compute capabilities and resources and their representation among service instances in the participating edges, at both design time and runtime.

Updates on Requirements -- Cont'd

- **Requirements on Metric Definition and Representation (Sec 5.2)**
  - R8: Beyond metrics definition, CATS solutions MUST contain the staleness handling of CATS metrics and indicate when to refresh the metrics, so that CATS components can know if a metric value is valid or not.
  - R9: All metric information used in CATS MUST be produced and encoded in a format that is understood by participating CATS components. For metrics that CATS components do not understand or support, CATS components will ignore them.
  - R10: CATS components SHOULD support a mechanism to advertise or negotiate supported metric types and encodings to ensure compatibility across implementations.
  - R11: The computation and use of metrics in CATS MUST be designed to avoid introducing routing loops or path oscillations when metrics are distributed and used for path selection.

➢ ***Refinement on the description of CATS metrics encoding, advertisement, and validation...***

Updates on Requirements -- Cont'd

- **Use of CATS Metrics (Sec 5.3)**

➢ ***Provide some complementary description on metric distribution and examples on the choice of signaling protocols, considering multiple solution choices...***

- R16: MUST NOT be sensitive to the update frequency of the metrics, and MUST NOT be dependent on or vulnerable to the mechanisms used to distribute the metrics.

➢ ***Refinement on the stability requirement with respect to diverse metric update frequencies.***

However, metrics that are oriented towards compute capabilities and resources in general can be highly dynamic, e.g., changing rapidly with the number of sessions, the CPU/GPU utilization and the memory consumption, etc. Service providers must determine at what interval or based on what events such information needs to be distributed. Overly frequent distribution with more accurate synchronization may result in unnecessary overhead in terms of signaling.

Moreover, depending on the service related decision logic, one or more metrics need to be conveyed in a CATS domain. The problem to be addressed here may be the frequency of such conveyance, and which CATS component is the decision maker for the service instance selection should also be considered. Thereby, choosing appropriate protocols for conveying CATS metrics is important. While existing routing protocols may serve as a baseline for signaling metrics, for example, BGP extensions[RFC4760] and GRASP[RFC8990]. These routing protocols may be more suitable for distributed systems. Considering about some centralized approaches to select CATS service instances, other means to convey the metrics can equally be chosen and even be realized, for example, ALTO protocol[RFC7285] which leverages restful API for publication of CATS metrics to a centralized decision maker. Specifically, a desirable system,

Updates on Requirements -- Cont'd

- **Requirements on Instance Affinity (Sec 5.4)**
  - R18: CATS systems MUST maintain instance affinity for stateful sessions and transactions.

➢ ***Refinement on instance affinity requirement for both transport sessions and application level transactions.***

  - R20: MUST avoid maintaining per-flow states for specific applications in network nodes for providing instance affinity. (Deleted "keeping fine runtime-state granularity")

  - R22: SHOULD support service continuity in the presence of UE or service instance mobility.

Security Considerations

- R25: The nature of a user activities SHOULD be protected to preserve user privacy, including minimizing the exposure of identifying or behavioral patterns.

- R26: Authenticated and Encrypted advertisements are REQUIRED to prevent rogue nodes from participating in the network.

➢ ***Add some detailed descriptions regarding security.***

Next Steps

- Regarding consolidation on use cases, personally I recommend adding a new sub-section referencing related work in other SDOs, like ETSI and 3GPP, but not in details.

- Service-specific requirements are not recommended to be added in the current document, considering the stability for both this document and the framework.

- For future extensions, still very welcome for more I-Ds and maybe ask for applicability of CATS into the milestone.

- Plan to ask for WGLC after this meeting.