CATS
Internet-Draft
Intended status: Informational
Expires: 4 January 2026

X. Liu
R. Yang
Y. Zhang
Pengcheng Laboratory
D. Ma
ZDNS

3 July 2025

DNS-based Service Discovery for Computing-Aware Traffic Steering (CATS)
draft-liu-cats-dns-service-discovery-00

Abstract

This document specifies how DNS-based Service Discovery (DNS-SD) can be used as a discovery and resolving method for mapping service identifiers to specific addresses within the CATS framework. It details extensions to DNS-SD to support CATS-specific service discovery requirements and describes how the discovery mechanism integrates with other components of the CATS architecture to enable compuating-aware traffic steering.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 4 January 2026.

Copyright Notice

Table of Contents

1. Introduction

The Computing-Aware Traffic Steering (CATS) framework [I-D.draft-ietf-cats-framework-07] is designed to enable traffic steering that takes into account both network conditions and computing resource availability. A key requirement of this framework is providing a discovery and resolving method for the mapping of a service identifier to a speficic address [I-D.draft-ietf-cats-usecases-requirements-06]. where computing resources are available.

This document specifies how DNS-based Service Discovery (DNS-SD) [RFC6763] can be extended and used to fulfull this requirement within the CATS framework. DNS-SD provides a standardized mechanism for service discovery using existing DNS infrastructure, making it well-suited for integration with the CATS architecture.

The approach outlined in this document enables:

* Publishing of computing service availability through DNS-SD

* Discovery of appropriate CATS service instances based on service and resource requirement

* Resolution of CATS service identifiers to specific network addresses.

* Advertisement of CATS service capabilities and parameters.

* Dynamic updates of service availability and characteristics.

This document describes the necessary extensions to DNS-SD to support CATS-specific parameters and how the discovery mechanism integrates with other components of the CATS framework.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

This document uses the terms defined in [RFC6763] and [I-D.draft-ietf-cats-framework-07].

4. Service Instance Names for CATS

CATS service instances MUST be identified using DNS-SD service instance name following the format defined in RFC 6763 [RFC6763]:

Service Instance Name = <Instance>.<Service>.<Domain>

where

* the <Instance> portion is a user-friendly name for the instance

* the <Service> portion indicates the name of a specific type of service

* the <Domain> portion indicates the domain name where the service registered

4.1. Service Names

As defined in RFC 6763 [RFC6763], the <Service> portion of a Service Instance Name consists of a pair of DNS labels

<Service-name>.<Proto>

where _<Service-name> is a symbolic name of the desired service, and _<Proto> is the symbolic name of the desired transport protocol.

For services using TCP, the second label is "_<tcp>", and for services using any transort protocol than TCP, the second label is _<udp>

This document defines the following primary types of service for CATS services:

* _cats-inference._tcp (for ML inference services)

* _cats-storage._tcp (for storage services)

* _cats-computing._tcp (for general computing services)

4.2. Service Instance Names

Service instance names in CATS follow the DNS-SD convention:

<instance-name>.service-name._tcp.<domain>

Where:

* <instance-name> is a user-friendly name for the service instance

* <domain> is the DNS domain in which the service is registered

For example:

edge-inference-1._cats-inference._tcp.example.com

The instance name SHOULD be unique within the domain to avoid conflicts.

4.3. Service Parameters and TXT Records

DNS TXT records are used to advertise CATS-specific parameters for each service instance. This section defines standard parameters that SHOULD be included in CATS service advertisements.

The following parameters are defined:

* "cpu": CPU capacity in normalized units (integer)

* "mem": Memory capacity in MB (integer)

* "lat": Expected processing latency in milliseconds (float)

* "load": Current load level (0-100) as a percentage (integer)

* "gpu": GPU availability and type (string)

* "accel": Other accelerator availability and type (string)

* "vers": Service version (string)

* "caps": Capabilities as a comma-separated list (string)

* "prio": Priority tier (integer, lower values indicate higher priority)

* "cost": Relative cost metric (integer)

* "avail": Availability status (0=offline, 1=online, 2=degraded)

For example, a TXT record for a CATS service might contain:

cpu=8 mem=16384 lat=15.5 load=35 gpu=nvidia-t4 vers=1.2 caps=inference,training prio=1 cost=2 avail=1

TXT record attributes MUST follow the format specified in RFC 6763, with attribute names and values separated by '=', and no spaces around the '=' sign.

4.4. SRV Records for Service Location

For each service instance, an SRV record MUST be published according to RFC 2782 to enable clients to locate the service. The SRV record format for CATS services instance is:

<instance-name>._cats._tcp.<domain> IN SRV <priority> <weight> <port> <target>

Where:

* <priority> represents the priority of the target host (lower values indicate higher priority)

* <weight> is used for load balancing among targets with the same priority

* <port> is the TCP port where the service is available

* <target> is the hostname of the machine providing the service

For example:

```
edge-inference-1._cats._tcp.example.com.  SRV 0 5 8080
compute1.example.com.
```

5. Integration with CATS Framework

5.1. Relationship with CATS Control Plane

The DNS-SD discovery mechanism integrates with the CATS control plane in the following ways:

* The CATS control plane MAY act as a discovery client, querying for available computing services and maintaining a database of available resources.

* The CATS control plane MAY facilitate service registration by providing interfaces and automation for DNS record management.

* The CATS control plane MAY implement advanced selection algorithms that consider both the parameters advertised via DNS-SD and additional network and computing metrics.

* For centralized deployments, the CATS control plane MAY provide a proxy service that mediates between clients and the DNS-SD infrastructure.

5.2. Service Parameter Advertising

Service parameters advertised through DNS-SD TXT records provide inputs to the CATS framework’s decision-making process for traffic steering.

The computing service MUST ensure that advertised parameters accurately reflect the current state and capabilities of the computing resource. Parameters SHOULD be updated when significant changes in resource availability or characteristics occur.

The CATS control plane MAY augment the DNS-SD parameters with additional information from other sources when making steering decisions.

5.3. Computing Resource Information

In addition to the basic parameters defined in Section 4.3, computing services MAY advertise more detailed information through additional TXT record attributes.

For application-specific capabilities, a naming convention using prefixes is RECOMMENDED:

* "app.X": Application-specific parameter X

For example:

app.model=resnet50 app.batch-size=16 app.precision=fp16

This extensible approach allows for advertising specialized capabilities while maintaining compatibility with the base specification.

5.4. Dynamic Updates

Computing services MUST update their DNS-SD records when significant changes in availability or capabilities occur. These updates can be performed through:

* Standard DNS Dynamic Update mechanisms [RFC2136]

* DNS Update Leases [RFC7553] for time-limited registrations

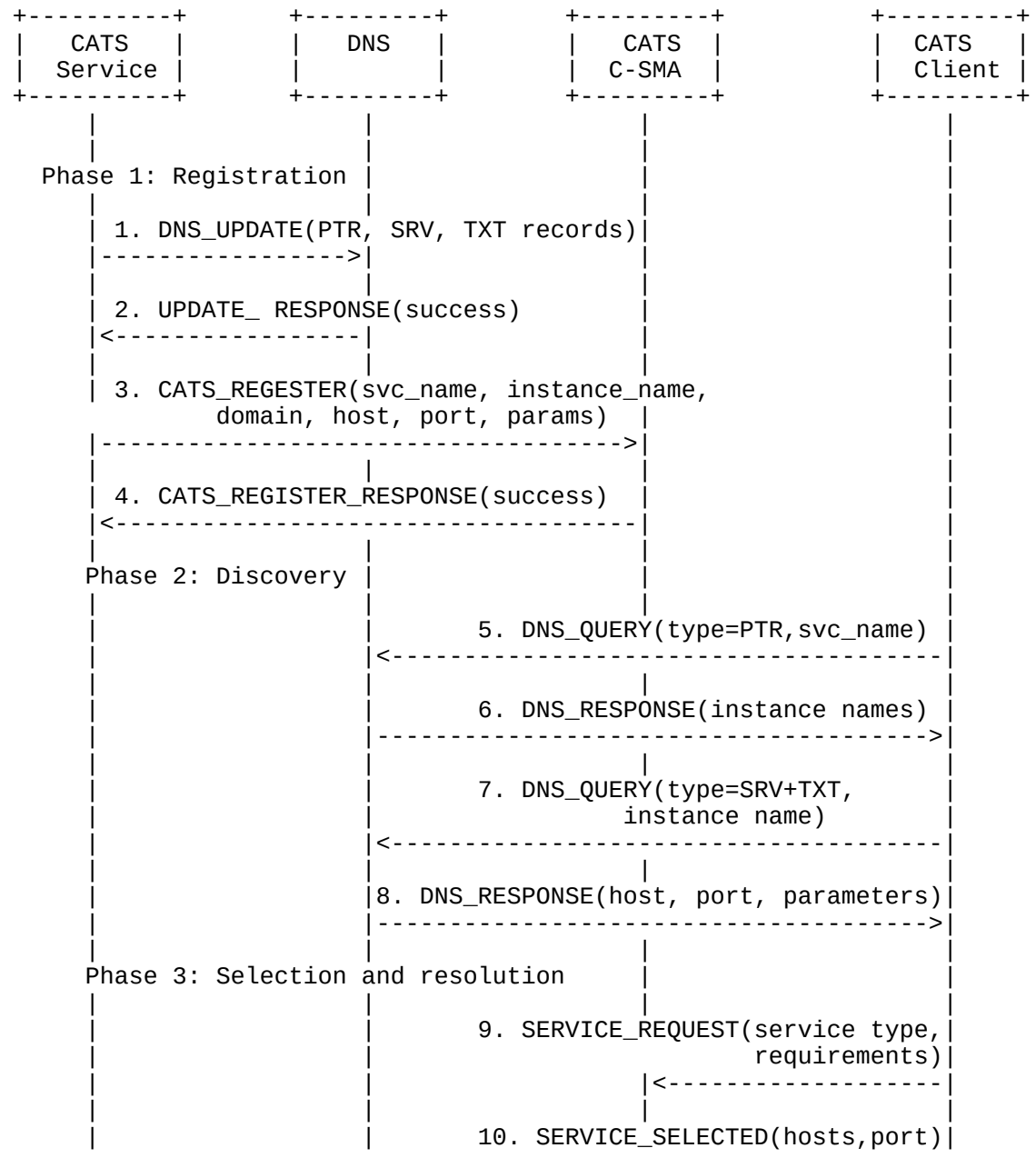
* Multicast DNS (mDNS) for local network scenarios

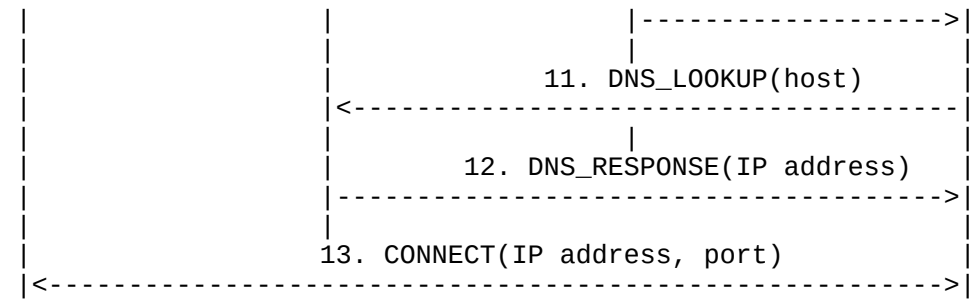
The frequency of updates SHOULD be balanced to reflect accurate information while avoiding excessive DNS traffic. Services SHOULD implement a dampening mechanism to avoid frequent updates for minor or transient changes.

For highly dynamic parameters like current load, services MAY implement a threshold-based update policy, only updating the DNS records when the parameter crosses predefined thresholds.

6. Service Discovery Process and Protocol Flow

The DNS-SD for CATS protocol flow can be shown in the following figure.





6.1. Registration Phase

Registration can be performed using standard DNS update mechanisms [RFC2136] or Dynamic DNS update protocols.

For each CATS service:

1. Create a PTR record pointing from the service type to the service instance name

2. Create an SRV record specifying the host and port for the service

3. Create a TXT record containing the CATS-specific parameters

4. Send a DNS_UPDATE message to the DNS server to add or update these records to the zone file.

5. Register to the C-SMA with service name, instance name and parameters.

The CATS control plane MAY facilitate this registration process through an appropriate management interface.

6.2. Discovery Phase

Clients requesting CATS services initiate the discovery process by querying for PTR records matching the appropriate service type. For example:

_cats-inference._tcp.<domain>

The response includes the list of matching service instance names. The client then queries for the SRV and TXT records associated with each service instance to obtain location information and service parameters.

6.3. Selection and Resolution Phase

After obtaining the list of available services and their parameters, the client or the CATS control plane performs service selection based on the application requirements and the advertised parameters.

The selection process MAY consider:

* resource capabilities (CPU, memory, GPU, etc.)
* Current load and availability
* Expected latency and performance metrics
* Priority and cost considerations
* Specific capabilities required by the application

Once a suitable service is selected, the client resolves the hostname from the SRV record to an IP address using standard DNS A or AAAA queries, and establishes a connection to the service using the specified port.

7. Implementation Considerations

7.1. Multicast DNS Considerations

In local network environments, Multicast DNS (mDNS) [RFC6762] MAY be used in conjunction with DNS-SD to provide service discovery without requiring a centralized DNS server.

When using mDNS, CATS services SHOULD:

* Respond to mDNS queries for their service type
* Advertise their presence periodically as specified in RFC 6762
* Implement proper conflict resolution mechanisms
* Consider the scope and scale of the deployment, as mDNS is primarily designed for local network use

7.2. DNS-SD/DNS Integration

For larger-scale deployments across multiple networks, traditional unicast DNS infrastructure is RECOMMENDED. In these scenarios:

* CATS services SHOULD be registered in appropriate DNS zones

* DNS infrastructure SHOULD support DNS Dynamic Updates

* DNS servers SHOULD be configured to allow updates from authorized CATS components

* Consider using DNS Update Leases for time-limited registrations

* Implement appropriate caching policies for DNS records

7.3. Performance Considerations

Implementers SHOULD consider the following performance aspects:

* DNS query volume: In large deployments with many clients, implement appropriate caching and consolidation of discovery requests.

* Update frequency: Balance the need for accurate information with the overhead of frequent DNS updates.

* Record size: TXT records have size limitations. For complex service descriptions, consider using a minimal set of parameters in DNS-SD and providing a URI for detailed metadata.

* Scalability: For very large deployments, consider hierarchical discovery approaches or specialized discovery proxies.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

The use of DNS-SD for CATS service discovery introduces several security considerations:

* Authentication: DNS updates for service registration SHOULD be authenticated to prevent unauthorized registration of services. DNS Security Extensions (DNSSEC) [RFC4033] SHOULD be implemented to provide authentication of DNS data.

* Information disclosure: Service parameters may reveal sensitive information about computing capabilities and deployment details. Consider the privacy implications of the parameters being advertised.

* Denial of Service: Large-scale DNS-SD queries could potentially be used for denial-of-service attacks. Implement rate limiting and monitoring for unusual query patterns.

* Spoofing: Without DNSSEC, DNS responses could potentially be spoofed, leading to service misdirection. DNSSEC validation SHOULD be enabled for DNS-SD queries.

* Data integrity: Ensure that computing parameter updates go through proper validation to prevent advertising incorrect capabilities, which could lead to suboptimal traffic steering decisions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", March 2005.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", February 2013.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", February 2013.

[RFC7553] Faltstrom, P. and O. Kolkman, "The Uniform Resource Identifier (URI) DNS Resource Record", June 2015.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", April 1997.

[I-D.draft-ietf-cats-framework-07] Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", April 2025.

[I-D.draft-ietf-cats-usecases-requirements-06]
Yao, K., Contreras, L. M., Shi, H., Zhang, S., and Q. An, "Computing-Aware Traffic Steering (CATS) Problem Statement, Use Cases, and Requirements", February 2025.

Authors' Addresses

Xiang Liu
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: liux15@pcl.ac.cn

Rongwei Yang
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: yangrw@pcl.ac.cn

Yu Zhang
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: zhangy08@pcl.ac.cn

Di Ma
ZDNS
Email: madi@zdns.cn