

Address Translation Service

Version 1

DAT Collaborative

Abstract: This specification defines the Address Translation Service (ATS), which provides ULPs with an ability to utilize IP addresses on InfiniBand networks.

1 Introduction

InfiniBand (IB) network addresses do not directly support IP addresses. Many existing applications that use IP addresses in order to support them over InfiniBand require IP address support over InfiniBand addresses. These applications locally pass an IP address and port identifier for a remote destination for connection establishment, analogous to socket connection establishment. On remote side an application identify a connection requestor by its IP address and port identifier. APIs[\[8\]](#)[\[9\]](#) have the same requirements since they provide abstraction - RDMA transport-independent APIs. To satisfy IP addressing support for ULPs for InfiniBand DAT Collaborative defines facility called Address Translation Service (ATS)[\[4\]](#). This service can be used by any ULP or API.

The problem of translating a GID to/from an IP address is specific to the InfiniBand transport. It is completely abstracted away from the APIs exposed to ULPs.

IB uses a GID when making a connection from one node to another. GIDs may either be formed from the port GUID, or may be dynamically assigned by the Subnet Manager (SM); see InfiniBand Architecture Specification Volume 1 Release 1.2 [\[5\]](#), section 4.1.1. Hence, ATS must support GID regardless how GIDs are administered.

ATS addresses the two key requirements:

- Translate an IP address to an IB GID
- Translate an IB GID to an IP address

The ATS does not define a new InfiniBand protocol. Instead it relies on the existing InfiniBand Subnet Administration Service Record Facility [\[5\]](#) (15.2.5.14).

Both current uDAPL and kDAPL implementations depend on ATS for IP address support on InfiniBand (IB) network [6], [7].

2 ATS OVERVIEW

ATS uses the facilities provided by the InfiniBand Subnet Administration (SA) [5] to resolve IP to GID and GID to IP mappings. There are two components to the resolution mechanisms, a service provider and a client. The service provider is responsible for registering and deregistering a service record with the SA. Each service record defines an IP address for a GID. A GID can support more than one IP address. A collection of such Service Records de-facto creates a database of IP to GID mappings that can be queried by clients in order to satisfy the requirements for ATS. The client is responsible for performing SA queries to retrieve IPs to GID or GIDs to IP mappings.

ATS uses the InfiniBand SA and logically is a part of an InfiniBand Subnet Manager (SM). IB provides all of the necessary mechanisms to implement ATS transparently to the system and to an ATS user like DAPL Providers. Service Records used by ATS must comply with the InfiniBand rules.

The SA is used as a central database in the following ways:

- 1). An IB driver¹ registers a ServiceRecord with the SA containing a GID and the local IP address. This function can be done by an ATS user like DAPL IB Provider, or by the lower level IB driver.
- 2). Connection requests obtain a remote GID from the SA by querying for Service Records containing the provided IP address.
- 3). The passive side of a connection may obtain an IP address from the SA by querying for Service Records containing the provided GID.
- 4). An IB driver deletes a ServiceRecord containing a GID and its local IP address. This function can be done by an ATS user like DAPL IB Provider, or by the lower level IB driver.

The IB MADs and methods necessary to use the SA in this way are defined in the IB specification[5], and ATS servers use it for Service Record registration and ATS clients use it for query.

1. IB driver is deliberately vague. It can be portion of IB driver, like SA, some other module of IB stack, or ULPs and APIs, like DAPL or IPoIB.

The rest of this proposal goes into detail on the above four items. To support ATS a fabric topology must include an SM supporting an SA.

This proposal is necessary for interoperability between DAPL Providers on InfiniBand. Any ULP can use ATS for interoperability and an ULP that uses DAPL API on one end of a connection must use ATS or its equivalent for interoperability.

A host may already have rules that map IP addresses to specific physical ports. ATS addresses address translation after a port is selected. It is assumed that network and system administrators have avoided conflicts between these two address assignment layers. This happens outside the scope of ATS.

2.1 ServiceRecords creation and deletion

Initial registration of ATS Service Records should be done once per available IP address for each GID used by ULP on a machine at the earliest possible time. The ATS Service Records are best registered when the IB driver is configured before upper layers, like DAPL, is invoked. Specifically it can be part of SA client initialization.

An ATS registers its IP Address for its local GID. Every GID that can be used for connection that has ATS requirements must register at least one Service Record with SID 0x1000CE100415453. This will be a primary IP address of the GID. Additional IP addresses for the same GID can also be registered but require different Service IDs.

A service record must be deleted when a record creator no longer owns the registered IP address. A service record should be deleted before the service goes offline or becomes unavailable. In general the publisher and maintainer of ATS Service Records should take appropriate steps to ensure that everyone else has accurate up-to-date information regarding the services offered.

When an IP address is disassociated from a GID, its record should be removed by ATS. How ATS becomes aware that IP address no longer associated with a GID is outside the scope of this specification.

When a GID is assigned a new IP address ATS registers a new record with SA for the GID and IP address combination. The Service Record PKEY is the partition key of the IB port of the

GID. Thus, the partition Key should match the underlying subnet administration had set to. The Service Record SERVICE-KEY is reserved and by default must be 0. For a new ATS service record for the GID the next SID from the ATS SID range is used. How ATS becomes aware that new IP address is associated with a GID is outside the scope of this specification.

When a GID is assigned a new *primary* IP address ATS queries the SA for the currently registered primary IP address, or uses locally cached information for it. ATS then reregister a Service Record with the primary SID 0x10000CE100415453 with a new IP address for the GID. Previously used primary IP address of the GID is registered as a new ATS service record. How ATS becomes aware that new *primary* IP address is associated with a GID is outside the scope of this specification.

When a GID's *primary* IP address is replaced, ATS reregisters a Service Record with the primary SID 0x10000CE100415453 with a new IP address for the GID. The previous primary IP address for a GID disappears and its IP address is no longer registered by ATS for the GID. How ATS becomes aware that new *primary* IP address is associated with a GID is outside the scope of this specification.

If multiple ULPs use the same IB ports and use the same partition then they will use the same ATS ServiceRecords, and share the same IP addresses. Clearly, they rely on the same primary IP address for a GID. ULPs may use different SLs and VLs but still share ATS records. It must be ensured that service records are not destroyed when one ULP terminates.

There are multiple ways that this can be achieved. One is for ULP themselves cooperate when maintaining ATS records. Another is that ULPs use APIs that perform that service, like DAPL. When each ULP opens an IA ATS records for it can be registered. For example, DAPL maintains the refcount of open IA instances and ensures that ATS records remain until the last ULP exits by closing IA.

If multiple ULPs use the same IB port but use its private partition, there is no issues since each partition will have its own ATS records. This scenario has added benefits, that each partition (hence ULP), can have its own primary IP address and other IP addresses specific to ULP.

ATS supports IP addresses in IPv4 and IPv6 formats. Which IP address that is primary for a GID is left to configuration management.

ATS and its users can cache some information locally. It is safe for ATS to cache all local information. It is advantageous to cache all local GIDs primary IP addresses. The local information can be kept in sync by the local ATS. Caching should not be performed by the ATS clients. Rather ATS clients should rely on caching services of the local SA client. All ATS records associated with a GID should be flushed when there is a failure with that GID.

2.2 ServiceRecords query

Any remote or local user that wants to connect to an IP address converts the IP address to a GID. This is done by queering SA for a service record that matches the IP address. DAPL performs this translation under the covers so users do not have to deal with IB specific addresses.

The ATS Service record database must contain at least one record for each participating GID. If it contains a single record for a GID then this record must use SID 0x10000CE100415453.

If an ATS Service Record does not exist in the SA database for an IP address of a connection request then the connection request fails locally. This can happen for either local or remote requested IP addresses or both.

Configuration information can cause multiple GIDs to register the same IP address. It is up to the ULP to define how to interpret this configuration information. A typical scenario is that all GIDs that register the same IP address can be used for a given IP address. Typically at least one GID uses an IP address as its primary IP GID. GID's Primary IP address has a Service Record with SID 0x10000CE100415453. Such a GID for a given IP address will be used by ULP directly or by DAPL API layer to establish connection. Any GID that matches this criteria will be used. If there is no Service Record with SID 0x10000CE100415453, ULP or DAPL will use any GID which matches the given IP address.

A user who is interested in obtaining the IP address of a connection requestor or an IP address of the remote end of an established connection can query the SA for a record of the remote

GID for a connection requestor or the remote connection end. DAPL does it under the covers so the user only deals with the IP addresses. DAPL will return the unique primary IP address of the GID. The primary IP address comes from the Service Record with SID 0x1000CE100415453.

Each Service Record contains the partition Key (PKEY) of the ULP of the IB HCA port. Thus, multiple service records for the same GID for a different partition may exist.

2.3 ATS SERVICE NAME

The service name is “DAPL Address Translation Service” starting from the first byte of the Service Name field of the Service Record. This service name ensures proper interoperability between ATS users as well as backwards compatibility with previous version of ATS. The relevant field in the IB Service-Record (15.2.5.14) is 64 bytes. Any bytes unused by the service name must be set to zero. Typically the unused bytes are set to zero for SA query and Service Record establishment and are ignored on receiving of a queried ServiceRecord. Even though the service name includes the word “DAPL”, it does not indicate a limit as to the scope of the ATS. The ATS can be used by any ULP.

2.4 SERVICE IDs

The DAT Collaborative owns a SID block from the ICSC[\[2\]\[3\]](#). The block consists of 256 entries. The entries are from 0x1000CE100415400 to 0x1000CE1004154FF. The SID 0x1000CE100415453 has a special meaning. For each GID there is a unique Service Record with that SID which contains the GID’s primary IP address.

ATS supports up to 256 IP addresses per GID.

2.5 IP ADDRESS ASSIGNMENT

The primary IP address associated with a GID must use SID 0x1000CE100415453. This ensures proper support of a reverse lookup by a SID/GID pair.

The IPv6 addresses occupy the 16 bytes of the Data8 field of the Service Record. The IPv4 addresses in the Service Record follow the standard encoding of IPv4 in IPv6[\[10\]](#).

The method by which the IP addresses are assigned to a GID is implementation specific. Implementations could depend on a driver such as IPoIB to register each assigned IP address. Stand alone applications or operating system daemons or services could also perform this registration. The IP address management is an administrative task that depends largely on the environment in which the systems are deployed.

Dynamic changes to IP address assignment could depend on OS provided mechanisms to detect such changes or could follow a polling model if the OS does not provide such facilities. The specific methods for detecting and managing such changes are implementation specific.

2.6 PARTITION KEY

Service PKEY (partition key) of the IB port must be used for the Service Record. If the PKEY is used by subnet administration it can be extracted from the configuration information, for example IPoIB partition specification. The same PKEY must be used by application on both side of the connection. Hence, ATS on both side will use the same PKEY for SA Service Records.

If no IB partitioning is used then ATS is free to use any PKEY. For example, ATS can use the PKEY of 0xFFFF as the default for its Service Records.

2.7 OTHER SERVICE RECORD FIELDS

ServiceLevel and Virtual Lane are outside of the scope of the ATS.

A Service Lease of an ATS Service Record should be by default 0xFFFFFFFF which represent keep the record indefinitely. Notice that record deletion rules specify when an ATS Service Record is removed from SA.

2.8 BACKWARDS COMPATIBILITY

The association of the primary IP address with ATS SID 0x1000CE100415453 allows implementations that support multiple IP address ATS to interoperate properly with implementations that support the single IP address ATS implementation. This covers all users of ATS version 0 [\[1\]](#) including earlier DAPL Providers. Keeping the service name consistent for all

ATS versions ensures that an ATS query by name retrieves a record created by previous versions of ATS.

ATS version 0 has been successfully used in DAPL Plugfest and demonstrated interoperability by IB vendors and users.

3 ATS Details

3.1 REGISTRATION and REREGISTRATION

Each Service record specifies a single IP to GID mapping.

Service registration consists of specifying the following fields in the service record:

Table 1 ATS Registration Service Record

RID			Reserved	Service Lease	Service Key	Service Name	Data	
Service ID	Service GID	ServiceP_key					Data8[0] to Data8[16]	Data16[1] - Data64[15]
8 bytes	16 bytes	2 bytes	2 bytes	4 bytes	16 bytes	64 bytes	16 bytes	3x16 bytes
0x10000CE100415400 to 0x10000CE1004154FF	local port GID	partition key	not used	0xFFFF FFFF (infinite)	0	DAPL Address Translation Service	IP Address	not used

The Service ID 0x10000CE100415453 is a base Service ID. All GIDs must have at least one Service Record which uses that SID. Additional Service Records for a GID shall increment Service ID starting from 0x10000CE100415454 wrapping around 0x10000CE1004154FF to 0x10000CE100415400 if needed.

The Service ID is an unsigned 64-bit big endian value. The Service Name is a string of length 64 bytes with “DAPL Address Translation Service” starting from byte 0.

The IP address occupies the first ServiceData8 field of length 16 bytes with the IP address starting at byte 0. The IPv6 address occupies the entire ServiceData8 field, while the IPv4 address is placed in octets 12-15 of the ServiceData8. The other Service-

Data fields are not used. The IP Address is in network byte order.

Service PKEY (partition key) of the IB port must be used for the Service Record. If the PKEY is used by subnet administration it can be extracted from the configuration information, for example IPoIB partition specification. The same PKEY must be used by application on both side of the connection. Hence, ATS on both side will use the same PKEY for SA Service Records.

Service Key is reserved and not used by ATS by this time. The default value is set to 0 on any set/delete operations and ignored on query operations to ensure interoperability.

A Service Lease of an ATS Service Record should be by default 0xFFFFFFFF.

The method for setting the Service Record is *SubnAdmSet* with an attribute 0x0031 for Service Record.

3.2 FORWARD LOOKUP: GIDs of an IP Address

IP to GID resolution consists of specifying the following fields in SA queries

Table 2 ATS Forward Query Service Record

RID			Reserved	Service Lease	Service Key	Service Name	Data	
Service ID	Service GID	ServiceP_key					Data8[0] to Data8[16]	Data 16[1] - Data 64[15]
8 bytes	16 bytes	2 bytes	2 bytes	4 bytes	16 bytes	64 bytes	16 bytes	3x16 bytes
		partition key	not used	not used	0	DAPL Address Translation Service	IP Address	not used

Note that this query could return multiple GIDs if the same IP address is assigned to multiple ports (e.g. for fail over or load balancing) or if a port has multiple GIDs assigned.

An ATS client must validate that the SID in each returned record falls within the ATS SID block and ignore any records that fail this validation.

It is assumed that any of the GIDs that falls within the ATS SID range can be used to establish connection. As stated above, an ATS user uses a GID that has the requested IP address as the *primary* IP address for the connection.

There is no requirement that SubnAdmGetTable is used to get all records. ATS implementation can get one record at a time using SubnAdmGet. ATS should start from the primary SID 0x10000CE100415453 and increment SID used for the query. Since Service Records with GID IPs increment SID once no record is, no further record need to be fetched¹. But even for a single SID and IP address combination multiple records may exist.

The operation for setting the Service Record is *SubnAdmGetTable* or *SubnAdmGet* with an attribute 0x0031 for Service Record.

3.3 REVERSE LOOKUP: IP Addresses of a GID

GID to IP resolution consists of specifying the following fields in SA queries:

Table 3 ATS Reverse Query Service Record

RID			Reserved	Service Lease	Service Key	Service Name	Data	
Service ID	Service GID	ServiceP_key					Data8[0] to Data8[16]	Data 16[1] - Data 64[15]
8 bytes	16 bytes	2 bytes	2 bytes	4 bytes	16 bytes	64 bytes	16 bytes	3x16 bytes
	GID	partition key	not used	not used	0	DAPL Address Translation Service		not used

1. Be aware that if a ServiceRecord is deleted then gaps in SIDs use by ServiceRecords can appear.

Note that this query could return multiple IP addresses. ATS does not provide a way of matching the requester's IP address with a record returned through such a lookup. However, all IP addresses returned accurately map to the requester.

The ATS client must validate that the SID in each returned records falls within the ATS SID block and ignore any records that fail this validation.

There is no requirement that *SubnAdmGetTable* is used to get all records. The ATS implementation can get one record at a time using *SubnAdmGet* and incrementing SIDs starting from the primary SID 0x10000CE100415453.

It is expected that a host name lookup for any IP address returned by a reverse lookup will resolve to the same name in order to support authentication by the host name. But it is always safe to use the primary IP address from the Service Record with SID 0x10000CE100415453.

The operation for setting the Service Record is *SubnAdmGetTable* or *SubnAdmGet* with an attribute of 0x0031 for the Service Record.

3.4 REVERSE LOOKUP: Primary IP Address of a GID

Reverse lookup for the purposes of authenticating a connection requestor requires an expedient lookup of the primary IP address associated with a GID. To support this, the SID at index 0 in the block of SIDs must always be used to represent the primary IP address associated with a port. The SID at index 0 is defined as the ATS SID 0x10000CE100415453.

GID to primary IP resolution consists of specifying the following fields in SA queries:

Table 4 ATS Reverse Query Primary Service Record

RID			Reserved	Service Lease	Service Key	Service Name	Data	
Service ID	Service GID	ServiceP_key					Data8[0] to Data8[16]	Data 16[1] - Data 64[15]
8 bytes	16 bytes	2 bytes	2 bytes	4 bytes	16 bytes	64 bytes	16 bytes	3x16 bytes
0x10000CE100415453	GID	partition key	not used	not used	0	DAPL Address Translation Service		not used

Note that this query can return at most one record due to the SA constraints of unique SID/GID pairs per service record. This limits the SA response to a single MAD, eliminating the overhead of RMPP from such a transaction. The IP address returned by such a query may not match the actual requester's IP address if it uses non-primary IP address of the IB port.

The operation for setting the Service Record is *SubnAdmGetTable* with an attribute 0x0031 for Service Record.

3.5 DEREGISTRATION

Service deregistration is the inverse of the registration and uses the following fields in the service record:

Table 5 ATS Deregistration Service Record

RID			Rese rved	Service Lease	Serv ice Key	Service Name	Data	
Service ID	Service GID	Serv iceP _key					Data8[0] to Data8[1 6]	Data 16[1] - Data 64[1 5]
8 bytes	16 bytes	2 byte s	2 byte s	4 bytes	16 byte s	64 bytes	16 bytes	3x16 byte s
0x10000CE100415400 to 0x10000CE1004154FF	local port GID	parti- tion key	not used					not used

The operation for setting the Service Record is *SubnAdmDelete* with an attribute 0x0031 for the Service Record.

A single Service Record defined by RID will be deleted at a time.

4 QUERY RECORD SETS

The following section describes query definitions for retrieving various record sets.

4.1 ALL RECORDS BY SID

An SA query to retrieve the records of all nodes with a given SID consists of specifying the following fields

Table 6 Query All Service Records by SID

RID			Rese rved	Service Lease	Serv ice Key	Service Name	Data	
Service ID	Service GID	Serv iceP _key					Data8[0] to Data8[1 6]	Data 16[1] - Data 64[1 5]
8 bytes	16 bytes	2 byte s	2 byte s	4 bytes	16 byte s	64 bytes	16 bytes	3x16 byte s
SID		parti tion key	not used	not used	0	DAPL Address Transla tion Ser vice		not used

The SID must be from the ATS SID range.

This query can be useful in validating that ATS is properly configured.

The operation for setting the Service Record is *SubnAdmGetTable* with an attribute 0x0031 for Service Record.

4.2 ALL RECORDS BY GID

This query retrieves all valid IP addresses of the GID IB port. An SA query to retrieve all records with a given GID consists of specifying the following fields:

Table 7 Query All Service Records by GID

RID			Rese rved	Service Lease	Serv ice Key	Service Name	Data	
Service ID	Service GID	Serv iceP _key					Data8[0] to Data8[1 6]	Data 16[1] - Data 64[1 5]
8 bytes	16 bytes	2 byte s	2 byte s	4 bytes	16 byte s	64 bytes	16 bytes	3x16 byte s
	GID	parti tion key	not used	not used	0	DAPL Address Transla tion Ser vice		not used

Any records returned by such a query that do not contain a valid ATS SID should be ignored.

The operation for setting the Service Record is *SubnAdmGetTable* with an attribute 0x0031 for Service Record.

5 ATS Usage

The simple sequence of events for ATS usage follows:

1) For each port on an HCA, the IB driver, API layer like DAPL, or any entity that would like to use the ATS facility will send a SubnAdmSet MAD to the SA in order to register an ATS specific Service Record.

2) When a node initiates a connection, it will do the following steps. First, it will query the SA with a SubnAdmGetTable MAD and extract Service Records. Then it will ignore Service Records that do not match ATS SID range. Next it chooses a GID for the requested remote IP Address from remaining Service Records. It is assumed that the local endpoint which requests the connection knows the local GID from which connection request will be generated.

3) The passive side of the connection will do the following steps to find the IP Address of connection requestor. First it will query the SA for a Service Record with a SubnAdmGet MAD with the primary SID and the requestor GID. Then it obtains the primary IP Address from the Service Record. The passive side of the connection may also request all IP addresses of the remote GID by issuing SubnAdmGetTable for the requestor GID.

5.1 Point-to-point case

If the topology is a simple point-to-point connection with no SA available, the initial registration (1) will fail and the IB driver may create local records to satisfy translation requests going forward. The same ATS interface provided by the IB driver will be transparent to the invoking application regardless of topology.

In this topology, address translation becomes the simple case where it is assumed that the IP Address is the same as the default port GID. Users in this topology must install the default port GID into their local IP name services, be it DNS or /etc/hosts or similar. The default port GID is an IPv6 address and is typically supported by most modern name services.

If the P2P topology is changed to a fabric topology, the default port GID will continue to work; the default port GID will essentially become just another IP address in the network, its significance as a 'GID' is no longer relevant. Users using GIDs for IP addresses in a fabric topology should be aware that replacing an HCA will not affect how this address is used; putting the original HCA into a different system with a more 'virtual' IP address can easily cause confusion.

Notice that for this scenario multiple IP addresses per GID are not supported due to the lack of SA.

6 REFERENCES

- [1] "Address Translation Proposal version 0" email to dat-discussions, sent by Steve Sears on Friday, May 23, 2003 <<http://groups.yahoo.com/group/dat-discussions/message/2391>>

- [2] ATS SID “FW: request for Service ID for ATS for DAT” email to dat-discussions, sent by Arkady Kanevsky on Friday, September 05, 2003
<<http://groups.yahoo.com/group/dat-discussions/message/2558>>
- [3] ATS SIDs “Resolving the confusion: ICSC ATS SID assignment (with attachment)” email to dat-discussion (DAT Reflector), send by Martin Kirk and forwarded by Arkady Kanevsky <<http://groups.yahoo.com/group/dat-discussions/message/3257>>
- [4] ATS proposal version 1, “Updated ATS and multiple IP addresses” email from Fabian Tillier Tue Jul 20, 2004 1:24 pm
<<http://groups.yahoo.com/group/dat-discussions/message/3100>>.
- [5] InfiniBand Architecture Specification, Release 1.2
- [6] DAPL Source Forge project: <http://sourceforge.net/projects/dapl/>
- [7] OpenIB project: <http://www.openib.org/>
- [8] uDAPL 1.2 spec, http://www.datcollaborative.org/udapl12_091504.zip
- [9] kDAPL 1.2 spec, http://www.datcollaborative.org/kDAPL12_091504.zip
- [10] [RFC2373] Hinden, R. and S. Deering. “IP version 6 Addressing Architecture”, IETF RFC 2373, July 1998.