Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 24, 2012

T. Narten
IBM
M. Karir
Merit Network Inc.
I. Foo
Huawei Technologies

February 21, 2012

Problem Statement for ARMD
draft-ietf-armd-problem-statement-01

Abstract

This document examines issues related to the massive scaling of data centers. Our initial scope is relatively narrow. Specifically, we focus on address resolution (ARP and ND) within the data center.

Table of Contents

1. Introduction

This document examines issues related to the massive scaling of data centers. Specifically, we focus on address resolution (ARP in IPv4 and Neighbor Discovery in IPv6) within the data center. Although strictly speaking the scope of address resolution is confined to a single L2 broadcast domain (i.e., ARP runs at the L2 layer below IP), the issue is complicated by routers having many interfaces on which address resolution must be performed or with IEEE 802.1Q domains, where individual VLANs form their own broadcast domains. Thus, the scope of address resolution spans both the L2 link and the devices attached to those links.

This document is intended to help the ARMD WG identify potential future work areas. The scope of this document intentionally starts out relatively narrow, mirroring the ARMD WG charter. Expanding the scope requires careful thought, as the topic of scaling data centers generally has an almost unbounded potential scope. This document aims to list "pain points" that are being experienced in current data centers. It is a separate exercise to determine which (if any) of these pain points should lead to specific protocol work, whether in ARMD or some other WG.

2. Terminology

Application: a service that runs on either a physical or virtual machine, providing a service (e.g., web server, database server, etc.)

Broadcast Domain: The set of all links and switches that are traversed in order to reach all nodes that are members of a given L2 domain. For example, when sending a broadcast packet on a VLAN, the domain would include all the links and switches that the packet traverses when broadcast traffic is sent.

Host (or server): Physical machine on which a system is run. A system can consist of an application running on an operating system on the "bare metal" or multiple applications running within individual VMs on top of a hypervisor. Traditional non-virtualized systems will have a single (or small number of) IP addresses assigned to them. In contrast, a virtualized system will use many IP addresses, one for the hypervisor plus one (or more) for each individual VM.

Hypervisor: Software running on a host that allows multiple VMs to run on the same host.

L2 domain: IEEE802.1Q domain supporting up to 4095 VLANs. The notion of an L2 broadcast domain is closely tied to individual VLANs. Broadcast traffic (or flooding to reach all destinations) reaches every member of the specific VLAN being used.

Virtual machine (VM): A software implementation of a physical machine that runs programs as if they were executing on a bare machine. Applications do not know they are running on a VM as opposed to running on a "bare" host or server.

ToR: Top of Rack Switch

3. Background

Large, flat L2 networks have long been known to have scaling problems. As the size of an L2 network increases, the level of broadcast traffic from protocols like ARP increases. Large amounts of broadcast traffic pose a particular burden because every device (switch, host and router) must process and possibly act on such traffic. In addition, large L2 networks can be subject to "broadcast storms". The conventional wisdom for addressing such problems has been to say "don't do that". That is, split large L2 networks into multiple smaller L2 networks, each operating as its own L3/IP subnet. Numerous data center networks have been designed with this principle, e.g., with each rack placed within its own L3 IP subnet. By doing so, the broadcast domain (and address resolution) is confined to one Top of Rack switch, which works well from a scaling perspective. Unfortunately, this conflicts in some ways with the current trend towards dynamic work load shifting in data centers and increased virtualization as discussed below.

Workload placement has become an issue within data centers. Ideally, it is desirable to be able to move workloads around within a data center in order to optimize server utilization, add additional servers in response to increased demand, etc. However, servers are often pre-configured to run with a given set of IP addresses. Placement of such servers is then subject to constraints of the IP addressing restrictions of the data center. For example, servers configured with addresses from a particular subnet could only be placed where they connect to the IP subnet corresponding to their IP addresses. If each top of rack switch is placed within its own subnet, a server can only be connected to the one top of rack switch. This same constraint occurs in virtualized environments, as discussed next.

Server virtualization is fast becoming the norm in data centers. With server virtualization, each physical server supports multiple virtual servers, each running its own operating system, middleware and applications. Virtualization is a key enabler of workload agility, i.e., allowing any server to host any application and providing the flexibility of adding, shrinking, or moving services among the physical infrastructure. Server virtualization provides numerous benefits, including higher utilization, increased data security, reduced user downtime, and even significant power conservation, along with the promise of a more flexible and dynamic computing environment.

The discussion below focuses on VM placement and migration. Keep in mind, however, that even in a non-virtualized environment, many of the same issues apply to individual workloads running on standalone machines. For example, when increasing the number of servers running a particular workload to meet demand, placement of those workloads may be constrained by IP subnet numbering considerations.

The greatest flexibility in VM and workload management occurs when it is possible to place a VM (or workload) anywhere in the data center regardless of what IP addresses the VM uses and how the physical network is laid out. In practice, movement of VMs within a data center is easiest when VM placement and movement does not conflict with the IP subnet boundaries of the data center's network, so that the VM's IP address need not be changed to reflect its actual point of attachment on the network from an L3/IP perspective. In contrast, if a VM moves to a new IP subnet, its address must change, and clients will need to be made aware of that change. From a VM management perspective, management is simplified if all servers are on a single large L2 network.

With virtualization, a single physical server can host 10 (or more) VMs, each having its own IP (and MAC) addresses. Consequently, the number of addresses per machine (and hence per subnet) is increasing, even when the number of physical machines stays constant. Today, it is not uncommon to support 10 VMs per physical server. In a few years, the number will likely reach 100 VMs per physical server.

In the past, services were static in the sense that they tended to stay in one physical place. A service installed on a machine would stay on that machine because the cost of moving a service elsewhere was generally high. Moreover, services would tend to be placed in such a way as to facilitate communication locality. That is, servers would be physically located near the services they accessed most heavily. The network traffic patterns in such environments could thus be optimized, in some cases keeping significant traffic local to one network segment. In these more static and carefully managed

environments, it was possible to build networks that approached scaling limitations, but did not actually cross the threshold.

Today, with the proliferation of VMs, traffic patterns are becoming more diverse and less predictable. In particular, there can easily be less locality of network traffic as services are moved for such reasons as reducing overall power usage (by consolidating VMs and powering off idle machine) or to move a virtual service to a physical server with more capacity or a lower load. In today's changing environments, it is becoming more difficult to engineer networks as traffic patterns continually shift as VMs move around.

In summary, both the size and density of L2 networks is increasing. In addition, increasingly dynamic workloads and the increased usage of VMs is creating pressure for ever larger L2 networks. Today, there are already data centers with 120,000 physical machines. That number will only increase going forward. In addition, traffic patterns within a data center are changing.

4. Generalized Data Center Design

There are many different ways in which data centers might be designed. The designs are usually engineered to suit the particular application that is being deployed in the data center. For example, a massive web sever farm might be engineered in a very different way than a general-purpose multi-tenant cloud hosting service. However in most cases the designs can be abstracted into a typical three-layer model consisting of an Access Layer, an Aggregation Layer and the Core. The access layer generally refers to the Layer 2 switches that are closest to the physical or virtual severs, the aggregation layer refers to the Layer 2 - Layer 3 boundary. The Core switches connect the aggregation switches to the larger network core. Figure 1 shows a generalized Data Center design, which captures the essential elements of various alternatives.

```
 +-----+-----+     +-----+-----+
 |   Core0   |     |    Core1  |      Core
 +-----+-----+     +-----+-----+
       /    \        /       /
      /      \----------\   /
     /    /---------/    \ /
   +-------+           +------+
 +/------+ |        +/------+ |
 | Aggr11| + --------|AggrN1| +      Aggregation Layer
 +---+---+/         +-------+/
   /     \            /      \
  /       \          /        \
+---+    +---+      +---+     +---+
|T11|... |T1x|      |T21|     |T2y|  Access Layer
+---+    +---+      +---+     +---+
|   |    |   |      |   |     |   |
+---+    +---+      +---+     +---+
|   |... |   |      |   |     |   |
+---+    +---+      +---+     +---+  Server racks
|   |... |   |      |   |     |   |
+---+    +---+      +---+     +---+
|   |... |   |      |   |     |   |
+---+    +---+      +---+     +---+
```

Figure 1: Typical Layered Architecture in DC

Figure 1

4.1. Access Layer

The Access switches provide connectivity directly to/from physical and virtual servers. The access switches might be placed either on top-of-rack (ToR) or at end-of-row(EoR) physical configuration. A server rack may have a single uplink to one access switch, or may have dual uplinks to two different access switches.

4.2. Aggregation Layer

In a typical data center, aggregation switches interconnect many ToR switches. Usually there are multiple parallel aggregation switches, serving the same group of ToRs to achieve load sharing. It is no longer uncommon to see aggregation switches interconnecting hundreds of ToR switches in large data centers.

4.3. Core

Core switches connect multiple aggregation switches and act as the data center gateway to external networks or interconnect to different

PODs within one data center.

4.4. Layer 3 / Layer 2 Topological Variations

4.4.1. Layer 3 to Access Switches

In this scenario the L3 domain is extended all the way to the Access Switches. Each rack enclosure consists of a single Layer 2 domain, which is confined to the rack. In general, there are no significant ARP/ND scaling issues in this scenario as the Layer 2 domain cannot grow very large. This topology is ideal for scenarios where servers attached to a particular access switch don't load applications that have been assigned IP addresses taken from a different subnet or where applications aren't moved to other racks under attached to different access switches, and hence, under a different IP subnet. A small server farm or very static compute cluster might be best served via this design.

4.4.2. L3 to Aggregation Switches

When the Layer 3 domain only extends to aggregation switches, hosts in any of the IP subnets configured on the aggregation switches can be reachable via Layer 2 through any access switches if access switches enable all the VLANs. This topology allows for a great deal of flexibility as servers attached to one access switch can be re-loaded with applications with different IP prefix and VMs can now migrate between racks without IP address changes. The drawback of this design however is that multiple VLANs have to be enabled on all access switches and all ports of aggregation switches. Even though layer 2 traffic are still partitioned by VLANs, the fact that all VLANs are enabled on all ports can lead to broadcast traffic on all VLANs to traverse all links and ports, which is same effect as one big Layer 2 domain. In addition, internal traffic itself might have to cross different Layer 2 boundaries resulting in significant ARP/ND load at the aggregation switches. This design provides the best flexibility/Layer 2 domain size trade-off. A moderate sized data center might utilize this approach to provide high availability services at a single location.

4.4.3. L3 in the Core only

In some cases where a wider range of VM mobility is desired (i.e. greater number of racks among which VMs can move without IP address change), the Layer 3 routed domain might be terminated at the core routers themselves. In this case VLANs can span across multiple groups of aggregation switches, which allow hosts to be moved among more number of server racks without IP address change. This scenario results in the largest ARP/ND performance impact as explained later.

A data center with very rapid workload shifting may consider this kind of design.

4.4.4. Overlays

There are several approaches regarding how overlay networks can make very large layer 2 network scale and enable mobility. Overlay networks using various Layer 2 or Layer 3 mechanisms enable interior switches/routers not to see the hosts' addresses. The Overlay Edge switches/routers which perform the network address encapsulation/ decapsulation still however see host addresses.

When a large data center has tens of thousands of applications which communicate with peers in different subnets, all those applications send (and receive) data packets to their L2/L3 boundary nodes if the targets are in different subnets. The L2/L3 boundary nodes have to process ARP/ND requests sent from originating subnets and resolve physical addresses (MAC) in the target subnets. In order to allow a great number of VMs to move freely within a data center without re-configuring IP addresses, they need to be under the common Gateway routers. That means the common gateway has to handle address resolution for all those hosts. Therefore, the use of overlays in the data center network can be a useful design mechanism to help manage a potential bottleneck at the Layer 2 / Layer 3 boundary by redefining where that boundary exists.

4.5. Factors that Affect Data Center Design

4.5.1. Traffic Patterns

Expected traffic patterns play an important role in designing the appropriately sized Access, Aggregation and Core networks. Traffic patterns also vary based on the expected use of the Data Center. Broadly speaking it is desirable to keep as much traffic as possible on the Access Layer in order to minimize the bandwidth usage at the Aggregation Layer. If the expected use of the data center is to serve as a large web server farm, where thousands of nodes are doing similar things and the traffic pattern is largely in and out a large data center, an access layer with EoR switches might be used as it minimizes complexity, allows for servers and databases to be located in the same Layer 2 domain and provides for maximum density.

A Data Center that is expected to host a multi-tenant cloud hosting service might have completely different requirements where in order to isolate inter-customer traffic smaller Layer 2 domains are preferred and though the size of the overall Data Center might be comparable to the previous example, the multi-tenant nature of the cloud hosting application requires a smaller more compartmentalized

Access layer. A multi-tenant environment might also require the use of Layer 3 all the way to the Access Layer ToR switch.

Yet another example of an application with a unique traffic pattern is a high performance compute cluster where most of the traffic is expected to stay within the cluster but at the same time there is a high degree of crosstalk between the nodes. This would once again call for a large Access Layer in order to minimize the requirements at the Aggregation Layer.

4.5.2. Virtualization

Using virtualization in the Data Center further serves to increase the possible densities that can be achieved. Virtualization also further complicates the requirements on the Access Layer as that determines the scope of server migrations or failover of servers on physical hardware failures.

Virtualization also can place additional requirements on the Aggregation switches in terms of address resolution table size and the scalability of any address learning protocols that might be used on those switches. The use of virtualization often also requires the use of additional VLANs for High Availability beaconing which would need to span across the entire virtualized infrastructure. This would require the Access Layer to span as wide as the virtualized infrastructure.

5. Address Resolution in IPv4

In IPv4, ARP provides the function of address resolution. To determine the link-layer address of a given IP address, a node broadcasts an ARP Request. The request is delivered to all portions of the L2 network, and the node with the requested IP address replies with an ARP response. ARP is an old protocol, and by current standards, is sparsely documented. For example, there are no clear requirement for retransmitting ARP requests in the absence of replies. Consequently, implementations vary in the details of what they actually implement [RFC0826][RFC1122].

From a scaling perspective, there are a number of problems with ARP. First, it uses broadcast, and any network with a large number of attached hosts will see a correspondingly large amount of broadcast ARP traffic. The second problem is that it is not feasible to change host implementations of ARP - current implementations are too widely entrenched, and any changes to host implementations of ARP would take years to become sufficiently deployed to matter. That said, it may be possible to change ARP implementations in hypervisors, L2/L3

boundary routers, and/or ToR access switches, to leverage such techniques as Proxy ARP and/or OpenFlow infused directory assistance approaches. Finally, ARP implementations need to take steps to flush out stale or otherwise invalid entries. Unfortunately, existing standards do not provide clear implementation guidelines for how to do this. Consequently, implementations vary significantly, and some implementations are "chatty" in that they just periodically flush caches every few minutes and rerun ARP.

6. Problem Itemization

This section articulates some specific problems or "pain points" that are related to large data centers. It is a future activity to determine which of these areas can or will be addressed by ARMD or some other IETF WG.

6.1. ARP Processing on Routers

One pain point with large L2 broadcast domains is that the routers connected to the L2 domain need to process "a lot of" ARP traffic. Even though the vast majority of ARP traffic may well not be aimed at that router, the router still has to process enough of the ARP request to determine whether it can safely be ignored. The ARP algorithm specifies that a recipient must update its ARP cache if it receives an ARP query from a source for which it has an entry [RFC0826].

One common router implementation architecture has ARP processing handled in a "slow path" software processor rather than directly by a hardware ASIC as is the case when forwarding packets. Such a design significantly limits the rate at which ARP traffic can be processed. Current implementations today can support in the low thousands of ARP packets per second, which is several orders of magnitude lower than the rate at which packets can be forwarded by ASICs.

To further reduce the ARP load, some routers have implemented additional optimizations in their ASIC fast paths. For example, some routers can be configured to discard ARP requests for target addresses other than those assigned to the router. That way, the router's software processor only receives ARP requests for addresses it owns and must respond to. This can significantly reduce the number of ARP requests that must be processed by the router.

Another optimization concerns reducing the number of ARP queries targeted at routers, whether for address resolution or to validate existing cache entries. Some routers can be configured to send out periodic gratuitous ARPs. Upon receipt of a gratuitous ARP,

implementations mark the associated entry as "fresh", resetting the revalidate timer to its maximum setting. Consequently, sending out periodic gratuitous ARPs can effectively prevent nodes from needing to send ARP requests intended to revalidate stale entries for a router. The net result is an overall reduction in the number of ARP queries routers receive. Gratuitous ARPs can also pre-populate ARP caches on neighboring devices, further reducing ARP traffic.

Finally, another area concerns how routers process IP packets for which no ARP entry exists. Such packets must be held in a queue while address resolution is performed. Once an ARP query has been resolved, the packet is forwarded on. Again, the processing of such packets is handled in the "slow path". This effectively limits the rate at which a router can process ARP "cache misses" and is viewed as a problem in some deployments today.

Although address-resolution traffic remains local to one L2 network, some data center designs terminate L2 subnets at individual aggregation switches/routers (e.g., see Section 4.4.2). Such routers can be connected to a large number of interfaces (e.g., 100 or more). While the address resolution traffic on any one interface may be manageable, the aggregate address resolution traffic across all interfaces can become problematic.

Another variant of the above issue has individual routers servicing a relatively small number of interfaces, with the individual interfaces themselves serving very large subnets. Once again, it is the aggregate quantity of ARP traffic seen across all of the router's interfaces that can be problematic. This "pain point" is essentially the same as the one discussed above, the only difference being whether a given number of hosts are spread across a few large IP subnets or many smaller ones.

When a L2/L3 boundary router receives data packets via its L3 interfaces destined towards hosts under its L2 domain, if the target address is not present in the router's ARP/ND cache, it usually holds the data packets and initiates ARP/ND requests towards its L2 domain to make sure the target actually exists before forwarding the data packets to the target. If no response is received, the router has to send the ARP/ND query multiple times. If no response is received after a number of ARP/ND requests, the router needs to drop all those data packets. This process can be very CPU intensive.

When hosts in two different subnets under the same L2/L3 boundary router need to communicate with each other, the L2/L3 router not only has to initiate ARP/ND requests to the target's Subnet, it also has to process the ARP/ND requests from the originating subnet. This process further adds to the overall ARP processing load.

6.2. IPv6 Neighbor Discovery

For the purposes of this document, IPv6's Neighbor Discovery behaves much like ARP, with a few notable differences. First, ARP uses broadcast, whereas ND uses multicast. Specifically, when querying for a target IP address, ND maps the target address into an IPv6 Solicited Node multicast address. From an L2 perspective, sending to a multicast vs. broadcast address may result in the packet being delivered to all nodes, but most (if not all) nodes will filter out the (unwanted) query via filters installed in the NIC -- hosts will never see such packets. Thus, whereas all nodes must process every ARP query, ND queries are processed only by the nodes to which they are intended.

Another difference concerns revalidating stale ND entries. ND requires that nodes periodically re-validate any entries they are using, to ensure that bad entries are timed out quickly enough that TCP does not terminate a connection. Consequently, some implementations will send out "probe" ND queries to validate in-use ND entries as frequently as every 35 seconds [RFC4861]. Such probes are sent via unicast (unlike in the case of ARP). However, on larger networks, such probes can result in routers receiving many such queries. Unfortunately, the IPv4 mitigation technique of sending gratuitous ARPs does not work in IPv6. The ND specification specifically specifies that gratuitous ND "updates" cannot cause an ND entry to be marked "valid". Rather, such entries are marked "probe", which causes the receiving node to (eventually) generate a probe back to the sender, which in this case is precisely the behavior that the router is trying to prevent!

It should be noted that ND does not require the sending of probes in all cases. Section 7.3.1 of [RFC4861] describes a technique whereby hints from TCP can be used to verify that an existing ND entry is working fine and does not need to be revalidated.

6.3. MAC Address Table Size Limitations in Switches

L2 switches maintain L2 MAC address forwarding tables for all sources and destinations traversing through the switch. These tables are populated through learning and are used to forward L2 frames to their correct destination. The larger the L2 domain, the larger the tables have to be. While in theory a switch only needs to keep track of addresses it is actively using, switches flood broadcast frames (e.g., from ARP), multicast frames (e.g., from Neighbor Discovery) and unicast frames to unknown destinations. Switches add entries for the source addresses of such flooded frames to their forwarding tables. Consequently, MAC address table size can become a problem as the size of the L2 domain increases. The table size problem is made

worse with VMs, where a single physical machine now hosts ten (or more) VMs, since each has its own MAC address that is visible to switches.

When layer 3 extends all the way to access switches (see Section 4.4.1), the size of MAC address tables in switches s not generally a problem. When layer 3 extends only to aggregation switches (see Section 4.4.2), however, MAC table size limitations can be a real issue.

7. Summary

This document has outlined a number of problems or "pain points" related to address resolution in large data centers. It is hoped that describing specific pain points will facilitate a discussion as to whether and how best to try and address them.

8. Open Issues

1. The document concentrates on ARP, but the same analysis needs to be performed for IPv6's Neighbor Discovery. Is section 6.2 enough?

9. Acknowledgments

This document has been significantly improved by comments from Linda Dunbar and Sue Hares. Igor Gashinsky deserves addition credit for highlighting some of the ARP-related pain points and for clarifying the difference between what the standards require and what some router vendors have actually implemented in response to operator requests.

10. IANA Considerations

This document makes not request of IANA.

11. Security Considerations

This documents lists existing problems or pain points with address resolution in data centers. This document does not create any security implications nor does it have any security implications. The security vulnerabilities in ARP are well known and this document does not change or mitigate them in any way.

12. Change Log

12.1. Changes between -00 and -01

1. Merged draft-karir-armd-datacenter-reference-arch-00.txt into this document.

2. Added section explaining how ND differs from ARP and the implication on address resolution "pain".

13. Informative References

[DATA1] Cisco, Systems., "Data Center Design - IP Infrastructure", October 2009.

[DATA2] Juniper, Networks., "Government Data Center Network Reference Architecture", 2010.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[STUDY] Rees, J. and M. Karir, "ARP Traffic Study", NANOG 52, URL http://www.nanog.org/meetings/nanog52/presentations/Tuesday/Karir-4-ARP-Study-Merit Network.pdf, June 2011.

Authors' Addresses

Thomas Narten
IBM

Email: narten@us.ibm.com

Manish Karir
Merit Network Inc.

Email: mkarir@merit.edu

Ian Foo
Huawei Technologies

Email: Ian.Foo@huawei.com