Proxy ARP as a Performance Enhancer: Reducing Latency and Improving Connectivity in Multi-Subnet Networks

Amaresan Venkatesan

v.amaresan@gmail.com

Abstract:
In contemporary network environments, optimizing performance is essential for ensuring efficient communication and resource utilization in the network. Proxy Address Resolution Protocol (ARP) is a technique that can significantly enhance network performance by enabling devices to communicate across different subnets without the need for complex routing configurations. This article delves into the principles of Proxy ARP, its benefits, implementation strategies, and best practices for optimizing network performance. This paper introduces a novel approach to handling Address Resolution Protocol (ARP) in Software-Defined Networking (SDN) by offloading the ARP functionality from the control plane to the data plane. Our proposed solution, Switch-based Proxy ARP (S-Proxy ARP), significantly reduces the controller load and ARP response time without relying on non-standard switch extensions, ensuring full compliance with the OpenFlow standard.

Keywords: Proxy ARP, Switch-based Proxy ARP, Software-Defined Networking (SDN), Neighbor Discovery Protocol (NDP), Media Access Control (MAC) address, VLAN.

1. INTRODUCTION

As networks grow in complexity, the need for efficient communication between devices on different subnets becomes increasingly important. Traditional routing methods can introduce latency and overhead, impacting overall network performance. Proxy ARP offers a solution by allowing a network device, such as a switch or router, to respond to ARP requests on behalf of other devices, effectively bridging different subnets and facilitating seamless communication. Layer 3 to Layer 2 address mapping is a critical functionality in packet-switched networks. In IPv4, this service is provided via the Address Resolution Protocol (ARP), and in IPv6 via the Neighbor Discovery Protocol (NDP). Efficient handling of ARP is crucial for the scalability of networks. In SDN, Proxy ARP is commonly used to manage ARP requests, but it places a significant load on the controller, limiting network performance and scalability.

2. Understanding Proxy ARP

A. ARP

The Address Resolution Protocol (ARP) provides a mechanism for hosts to map an IP address to a MAC address on the local network. The ARP packet structure includes key fields such as Sender Hardware Address (SHA) and Target Hardware Address (THA), which are the MAC addresses of the sender and the intended receiver.

Preamble	**Dest MAC**	**Src MAC**	**Ether Type (0x0806)**
Hardware Type		**Protocol Type**	
Hardwre Length	**Protocol Length**	**Operation (Request 1, Reply 2)**	
Sender Hardware Address (SHA)			
Sender Protocol Address (SPA)			
Target Hardware Address (THA)			
Target Protocol Address (TPA)			
Frame check sequence			

Fig1. ARP Frame

B. OpenFlow

OpenFlow is a predominant SDN southbound interface that allows a controller to manipulate forwarding rules of switches. It supports the basic match-action paradigm, enabling the controller to manage packet forwarding and rewriting packet fields.

C. ARP Handling

There are two basic methods for handling ARP in OpenFlow-based SDNs: Regular ARP and Proxy ARP. Regular ARP works as in traditional networks, while Proxy ARP involves the controller handling ARP requests on behalf of hosts. Proxy ARP reduces broadcast traffic but increases controller load. Proxy ARP enables a network device to respond to ARP queries for network addresses by offering its own Ethernet Media Access Control (MAC) address. When a device sends an ARP request to determine the MAC address of a destination IP address, the proxy ARP-enabled device intercepts the request and responds with its own MAC address. This allows the requesting device to send packets to the proxy ARP device, which then forwards them to the intended destination.

D. Switch-based Proxy ARP (S-Proxy ARP)

S-Proxy ARP offloads ARP handling from the control plane to the data plane. The switch directly replies to ARP requests without involving the control plane. Typically the ARP packets has to reach the control plane and then software responds. This approach leverages OpenFlow's ability to match and rewrite ARP payload fields, converting ARP requests into ARP replies at the switch.

3. Benefits of Using Proxy ARP

Proxy ARP provides several advantages for network performance optimization:

1. **Simplified Network Configuration:** By eliminating the need for complex routing configurations, Proxy ARP simplifies network management and reduces administrative overhead.
2. **Enhanced Communication Across Subnets:** Proxy ARP allows devices on different subnets to communicate seamlessly, improving overall network efficiency.
3. **Reduced Latency:** By enabling direct communication between devices, Proxy ARP reduces the latency associated with traditional routing methods.

4. Implementation Strategies

Implementing Proxy ARP involves configuring network devices to respond to ARP requests on behalf of other devices. This can be done on a per-interface basis or for specific VLANs. There are two modes of Proxy ARP: restricted and unrestricted. In restricted mode, the device responds to ARP requests only if the

source and target are on different subnets. In unrestricted mode, the device responds to all ARP requests for which it has a route to the destination.

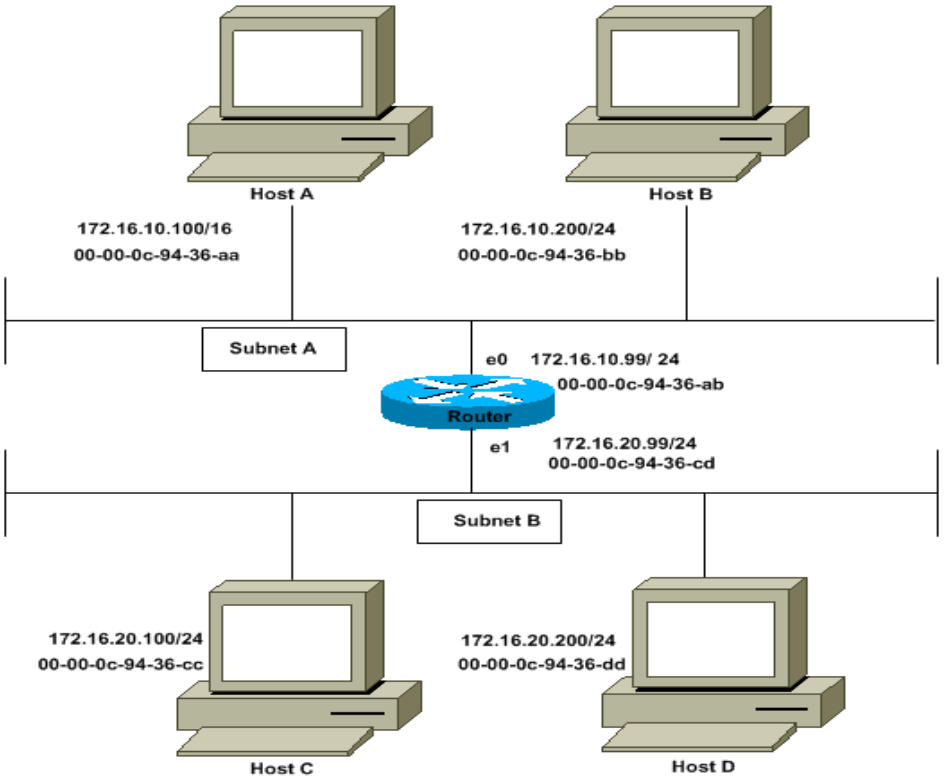

Fig2. Example of proxy ARP

Our experimental evaluations demonstrate the performance benefits of SProxy ARP, showing a reduction in ARP response timemore than an order of magnitude and a significant reduction in controller load.

5. Best Practices for Proxy ARP

To optimize network performance using Proxy ARP, consider the following best practices:

1. **Use Restricted Mode**: Configure Proxy ARP in restricted mode to ensure that the device only acts as a proxy for devices on different subnets, reducing unnecessary ARP traffic.
2. **Disable Gratuitous ARP Requests**: If using unrestricted mode, disable gratuitous ARP requests to prevent potential IP conflicts.
3. **Monitor ARP Statistics**: Regularly monitor ARP statistics to ensure that Proxy ARP is functioning correctly and to identify any potential issues.

6. Conclusion

In this study, we have explored the potential of Proxy ARP and S-Proxy ARP as a performance-enhancing technique for reducing latency and improving connectivity in multi-subnet networks. By allowing devices to communicate seamlessly across subnet boundaries without the need for complex routing, Proxy ARP simplifies network configurations and boosts operational efficiency. The ability of Proxy ARP to bridge subnets directly results in faster communication paths, minimizing the overhead typically associated with routing and reducing latency in inter-subnet communication.

Additionally, Proxy ARP enhances network connectivity by enabling devices to resolve IP addresses that might otherwise be inaccessible due to subnet segregation. This improves overall network reliability, especially in environments where devices are spread across multiple subnets but must function as part of the

same logical network.

Overall, Proxy ARP can serve as a powerful tool for optimizing network performance in multi-subnet configurations, but its use requires careful consideration of both the technical and security challenges. Future research should focus on developing more secure and scalable solutions to integrate Proxy ARP into larger, more complex network environments while maximizing its performance benefits.

References

1. K. Elmeleegy and A. L. Cox, “Etherproxy: Scaling Ethernet by Suppressing Broadcast Traffic,” in Proc. of IEEE INFOCOM 2009, pp. 1584–1592.
2. A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, “Applying NOX to the Datacenter,” in Proc. HotNets, 2009.
3. OpenFlow Standard. [Online]. Available: https://www.opennetworking.org/sdn-resources/onf-specifications/openflow
4. D. Plummer, “Ethernet Address Resolution Protocol: or Converting Network Protocol Addresses to 48-bit Ethernet Address for Transmission on Ethernet Hardware,” 1982.
5. K. R. Fall and W. R. Stevens, TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley, 2011.
6. K. Kwon, S. Ahn, and J. W. Chung, “Network Security Management using ARP Spoofing,” in Computational Science and Its Applications – ICCSA 2004. Springer, 2004, pp. 142–149.