**A PROJECT REPORT ON**

**INTRODUCTION TO COMPUTER NETWORKS**

**“CROSS-PROTOCOL ANALYSIS USING WIRESHARK”**

**Submitted by**

SANGGIT SAARAN K C S (CB.SC.U4AIE23247)

SURYA HA (CB.SC.U4AIE23267)

VENKATRAM KS (CB.SC.U4AIE23236)

VISHAL SESHADRI B (CB.SC.U4AIE23260)

In partial fulfilment for the award of the degree of BACHELOR OF TECHNOLOGY CSE(AI)

Under the Guidance of **Mr.** **Jaisooraj J** Centre for Computational Engineering and Networking AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE AMRITA VISHWA VIDYAPEETHAM COIMBATORE - 641 112 (INDIA)

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Dr. K P Soman, Dean of our Department, for providing us the opportunity to work on the project entitled “**CROSS-PROTOCOL ANALYSIS USING WIRESHARK” ,** is the record of the original work done by our team underthe guidance of **Mr.** **Jaisooraj J** Assistant professor, Centre for Computational Engineering and Networking (CEN), Amrita Vishwa Vidyapeetham, Coimbatore. To the best of our knowledge this work has not formed the basis for the award of any degree/diploma/associate ship/fellowship/or a similar award to any candidate in any university.

**Signature of the Faculty**

|  |  |  |
| --- | --- | --- |
| **Project Team:** | |  |
|  | **Name** | **(Roll Number)** |
| 1. | SANGGIT SAARAN K C S | (CB.SC.U4AIE23247) |
| 2. | SURYA H A | (CB.SC.U4AIE23267) |
| 3. | VENKATRAM K S | (CB.SC.U4AIE23236) |
| 4. | VISHAL SESHADRI B | (CB.SC.U4AIE23260) |

**Date:** 12-11-2024

1. **Introduction**

This project explores cross-protocol analysis using Wireshark, a powerful tool for network protocol analysis. The focus here is on understanding how DNS (Domain Name System) and HTTP (Hypertext Transfer Protocol) interactions affect network efficiency. These two protocols are foundational for internet connectivity -DNS resolves domain names to IP addresses, enabling web browsing, while HTTP governs the transmission of data over the web. However, the time dependency between DNS and HTTP processes can lead to performance delays, impacting user experience, especially on high-traffic or latency-sensitive networks.

The analysis was centered on determining how DNS response times influence HTTP request efficiency and exploring optimization strategies, primarily through DNS caching, to mitigate delays. By reducing DNS lookup times, we aim to see how quickly HTTP requests can be processed without waiting for DNS resolutions, ultimately enhancing network performance.

The objectives of this analysis were twofold:

* **Analyse DNS-HTTP Dependencies**: This project sought to capture and quantify the dependencies between DNS response times and the subsequent HTTP requests. Understanding these dependencies allows for better insight into how DNS latency could lead to bottlenecks in HTTP traffic.
* **Develop an Optimization Strategy**: By implementing DNS caching, we aimed to reduce DNS latency and thus improve HTTP performance. The caching strategy was intended to show a measurable reduction in dependency times, highlighting the benefits of cross-protocol optimization.

1. **Methodology**

* **Packet Capture**: Wireshark was used to capture and timestamp DNS queries, DNS responses, and corresponding HTTP requests. This data formed the basis for calculating dependency times between DNS responses and HTTP requests.
* **Dependency Analysis**: The study primarily focused on the dependency between DNS response times and the initiation of HTTP requests. This dependency means that each HTTP request must wait for DNS resolution before proceeding, causing potential delays in HTTP performance.
* **Optimization by DNS caching**: To reduce this dependency, DNS caching was introduced. With caching, HTTP requests could bypass repeated DNS lookups, resulting in consistently lower dependency times. The optimized setup showed more stable, shorter dependency times, reflecting a smoother HTTP performance.
* **Unoptimized vs. Optimized Dependency Times**: In the unoptimized scenario (without caching), each HTTP request is matched with the nearest DNS response, regardless of whether it’s the most recent. This resulted in varying DNS-HTTP dependency times, depending on network conditions and DNS server response times. In contrast, optimized scenario will use the latest DNS response for each destination before the HTTP request, simulating a cached scenario, where dependencies are minimized by avoiding repeated DNS lookups.

1. **Results and Analysis:**

* **Scatter Plot of DNS-HTTP Dependency Time**: The optimized results, exhibit lower DNS-HTTP dependency times compared to unoptimized results. This indicates that caching has successfully reduced the DNS dependency for HTTP requests, as seen by the tighter clustering of lower dependency times.
* **Histogram Analysis**: The histogram comparing unoptimized and optimized setups provided a visual representation of the frequency of dependency times. For the unoptimized setup, a broader distribution with higher dependency times indicated frequent delays due to DNS lookups. The optimized setup showed a shift toward lower dependency times, indicating that cached DNS information allowed HTTP requests to proceed faster, resulting in a higher frequency of lower times.
* **Cumulative Average Plot**: The cumulative average plot illustrated a steady rise in dependency times for the unoptimized setup, with times increasing linearly as HTTP requests continued to rely on DNS lookups. In the optimized setup, while initial dependency times were high, they levelled off over time. This levelling effect demonstrates that caching reduces cumulative dependency time, as fewer DNS queries are required over the session.

1. **Conclusion**

Through this cross-protocol analysis, Wireshark proved invaluable in highlighting the impact of DNS response times on HTTP requests. DNS caching was shown to effectively reduce delays by matching nearest DNS response, allowing HTTP requests to proceed without waiting for repeated DNS lookups. This strategy led to faster DNS-HTTP transactions and improved network performance, demonstrating that even modest optimizations in protocol interaction can lead to significant efficiency gains.

By examining DNS-HTTP interactions, this project demonstrates how cross-protocol optimization can benefit real-world network performance, potentially offering a model for similar analyses across different network protocols. This has practical applications in network management:

* **Improved User Experience**: Reduced dependency times mean faster loading times for users, especially in high-traffic environments where DNS lookups can otherwise create bottlenecks.
* **Resource Efficiency**: Caching reduces the load on DNS servers, as fewer queries are required. This also decreases the bandwidth consumed by repeated DNS lookups, offering a more resource-efficient setup.
* **Potential for Broader Application**: While this project focused on DNS and HTTP, similar cross-protocol analyses could be applied to other dependencies within network traffic, like HTTPS and TCP, to identify and mitigate other latency issues.

Future work could extend this study to other protocol combinations, leveraging caching and other optimization techniques to further improve network speed and reliability.