INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 4 ISSUE 2 (2018) PAGES 2849 - 2854
RECEIVED : 25.02.2018 PUBLISHED : 21.03.2018
March 21, 2018

LATENCY EQUALIZATION AS A NEW NETWORK SERVICE PRIMITIVE

Ms.A.Kowsar.,M.Sc (Computer science) IInd year.
Asst Prof.K.R.Aruna.,M.Sc.,M.Phil.,
Department of Computer Science, Kamban Arts And Science College for Women, Thiruvannamalai.

ABSTRACT

Multiparty interactive network applications such as teleconferencing, network gaming, and online trading are gaining popularity. In addition to end-to-end latency bounds, these applications require that the *delay difference* among multiple clients of the service is minimized for a good interactive experience. We propose a **Latency EQualization (LEQ) service**, which equalizes the perceived latency for all clients participating in an interactive network application. To effectively implement the proposed LEQ service, network support is essential. The LEQ architecture uses a few routers in the network as *hubs* to redirect packets of interactive applications along paths with similar end-to-end delay. We first formulate the hub selection problem, prove its NP-hardness, and provide a greedy algorithm to solve it. Through extensive simulations, we show that our LEQ architecture significantly reduces delay difference under different optimization criteria that allow or do not allow compromising the per-user end-to-end delay. Our LEQ service is incrementally deployable in today's networks, requiring just software modifications to edge routers.

1. INTRODUCTION

The increased availability of broadband access has spawned a new generation of netizens. Today, consumers use the network as an interactive medium for multimedia communications and entertainment. This growing consumer space has led to several new network applications in the business and entertainment sectors. End-to-end delay requirements can be achieved by traffic engineering and other QoS techniques. However, these approaches are insufficient to address the needs of multiparty interactive network applications that require bounded delay difference across multiple clients to improve interactivity. Previous work on improving online interactive application experiences considered application-based solutions either at the client or server side to achieve equalized delay. Client side solutions are hard to implement because they require that all clients exchange latency information to all other clients. They are also vulnerable to cheating .Server-side techniques rely on the server to estimate network delay, which is not sufficiently accurate in some scenarios. Moreover, this delay estimation places computational and memory overhead on the application servers, which limits the number of clients the server can support. Previous studies have investigated different interactive applications, and they show the need for network support to reduce delay difference since the prime source of the delay difference is from the network. The importance of reducing latency imbalances is further emphasized when scaling to wide geographical areas as witnessed by a press release from AT&T.

We design and implement *network-based* Latency EQualization (LEQ), which is a service that Internet service providers (ISPs) can provide for various interactive network applications. Compared to application-based latency equalization solutions, ISPs have more detailed knowledge of current network traffic and congestion, and greater access to network resources and routing control. Therefore, ISPs can better support latency equalization routing for a large number of players with varying delays to the application servers. This support can significantly improve game experience, leading to longer play time and thus larger revenue streams.

Our network-based LEQ service provides equalized-latency paths between the clients and servers by redirecting interactive application traffic from different clients along paths that minimize their delay difference.1We achieve equalized-latency paths by using a few routers in the network as *hubs*, and interactive application packets from different clients are redirected through these hubs to the servers. Hubs can also be used to steer packets away from congested links.

Our LEQ architecture provides a flexible routing framework that enables the network provider to implement different delay and delay difference optimization policies in order to meet the requirements of different types of interactive applications. In one policy scenario, latency equalization among different interactive clients can be achieved without compromising the end-to-end delay of individual clients. Similar to these works, our LEQ routing can minimize delay difference without compromising the end-to-end delay. In the other policy scenario, if the application can tolerate some moderate increase in the end-to-end delay, it is possible to achieve even better latency equalization among clients. To achieve LEQ routing, we formulate the *hub selection problem*, which decides which routers in the network can be used as hubs and the assignment of hubs to different client edge routers to minimize delay difference. We prove that this hub selection problem is NP-hard and inapproximable. Therefore, we propose a greedy algorithm that achieves equalized-latency paths.

2. STUDY OF EXISTING SYSTEM

Improving online interactive application experiences considered application-based solutions either at the client or server side to achieve equalized delay. Client side solutions are hard to implement because they require that all clients exchange latency information to all other clients. They are also vulnerable to cheating. Server-side techniques rely on the server to estimate network delay, which is not sufficiently accurate in some scenarios. This delay estimation places computational and memory overhead on the application servers, which limits the number of clients the server can support . Previous studies have investigated different interactive applications, and they show the need for network support to reduce delay difference since the prime source of the delay difference is from the network. The importance of reducing latency imbalances is further emphasized when scaling to wide geographical areas as witnessed by a press release from AT&T.

PROBLEM DESCRIPTION: Client side solutions are hard to implement because they require that all clients exchange latency information to all other clients. They are also vulnerable to cheating. Server-side techniques rely on the server to estimate network delay, which is not sufficiently accurate in some scenarios. We design and implement *network-based* Latency EQualization (LEQ), which is a service that Internet service providers (ISPs) can provide for various interactive network applications. Compared to application-based latency equalization solutions, ISPs have more detailed knowledge of current network traffic and congestion, and greater access to network resources and routing control. Therefore, ISPs can better support latency equalization routing for a large number of players with varying delays to the application servers. This support can significantly improve game experience, leading to longer play time and thus larger revenue streams. Due to the problems of client-side solutions, several delay compensation schemes are implemented at the server side. However, while introducing CPU and memory overhead on the server, they still do not completely meet the requirements of fairness and interactivity. For example, with the *bucket synchronization* mechanism , the received packets are buffered in a bucket, and the server calculations are delayed until the end of each bucket cycle. The performance of this method is highly sensitive to the bucket (time window) size used, and there is a tradeoff between interactivity versus the memory and computation overhead on the server. In the *time warp synchronization* scheme, snapshots of the game state are taken before the execution of each event. When there are late events, the game state is rolled back to one of the previous snapshots, and the game is reexecuted with the new events. This scheme does not scale well for fast-paced, high-action games because taking snapshots on every event requires both fast computation and large amounts of fast memory, which is expensive. In , a game-independent application was placed at the server to equalize delay differences by constantly measuring network delays and adjusting players' total delays by adding artificial lag. However, experiments in [12 suggest that using server-based round-trip-time measurements to design latency compensation across players fails in the presence of asymmetric latencies.

3. PROPOSED SYSTEM FOR LATENCY EQUALIZATION

The design and implement network-based Latency EQualization (LEQ), which is a service that Internet service providers (ISPs) can provide for various interactive network applications. Compared to application-based latency equalization solutions, ISPs have more detailed knowledge of current network traffic and congestion, and greater access to network resources and routing control. Therefore, ISPs can better support latency equalization routing for a large number of players with varying delays to the application servers. This support can significantly improve game experience, leading to longer play time and thus larger revenue streams. Our network-based LEQ service provides equalized-latency paths between the clients and servers by redirecting interactive application traffic from different clients along paths that minimize their delay difference. We achieve equalized-latency paths by using a few routers in the network as hubs, and interactive application packets from different clients are redirected through these hubs to the servers. Hubs can also be used to steer packets away from congested links.

Our LEQ architecture provides a flexible routing framework that enables the network provider to implement different delay and delay difference optimization policies in order to meet the requirements of different types of interactive applications.

To achieve LEQ routing, we formulate the hub selection problem, which decides which routers in the network can be used as hubs and the assignment of hubs to different client edge routers to minimize delay difference. We prove that this hub selection problem is NP-hard and inapproximable. Our LEQ routing significantly reduces delay difference in different network settings

Advantages: LEQ is achieved by optimized hub selection and assignment. Each client edge router is assigned to more than one hub, so it has the flexibility to select among its assigned hubs to avoid congestion. Our LEQ routing significantly reduces delay difference in different network settings.

4. IMPLEMENTATION

Client Node: Client traffic from an interactive application enters the provider network through edge routers R1 and R2. The server for the interactive application is connected to the network through edge router R10. Using the LEQ routing architecture, R6 and R7 are chosen as hubs for R1 , R7 and R8 are chosen as hubs for R6.Usingredirection through hubs, R1 has two paths to the server edge router R10:R1-R6-R10 and R1-R7-R10, both of which have a delay of 10ms. R2 has two paths R2-R7-R10 and R2-R8-R10, whole delay is also 10ms.Each client edge router is assigned to more than one hub, so it has the flexibility to select among its assigned hubs to avoid congestion.

Server Node: R1 ,R2 … are routers in the network. Server side edge router R10.Server node from an interactive application enters the provider network through edge routersR10.The number of the links represent the latency of each link R6,R7 and R8 are the hubs for R1,R2 and R10.Client edge routers redirect the application packets corresponding to the LEQ service through the hubs to the destined servers. By redirecting through the hubs, application packets from different client edge routers with different delays to the servers are guaranteed to reach the servers within a bounded delay difference. The average delay difference for packets from different sites to the server through these two hubs.

Hub selection process: To solve the hub selection problem, we design a simple greedy heuristic algorithm to pick the hubs. Our algorithm first sorts in increasing order all the delays from each client edge router through each possible hub to its associated servers. This sorted list is denoted by the array.

One could use source routing to address the problem of latency equalization. Source routing can be used by the sender to choose the path taken by the packet. However, this requires that all clients are aware of the network topology and coordinate with each other to ensure that the delay differences are minimized.

LEQ: We design and implement *network-based* Latency EQualization (LEQ), which is a service that Internet service providers (ISPs) can provide for various interactive network applications. Compared to application-based latency equalization solutions, ISPs have more detailed knowledge of current network traffic and congestion, and greater access to network resources and routing control. Therefore, ISPs can better support latency equalization routing for a large number of players with varying delays to the application servers. This support can significantly improve game experience, leading to longer play time and thus larger revenue streams.

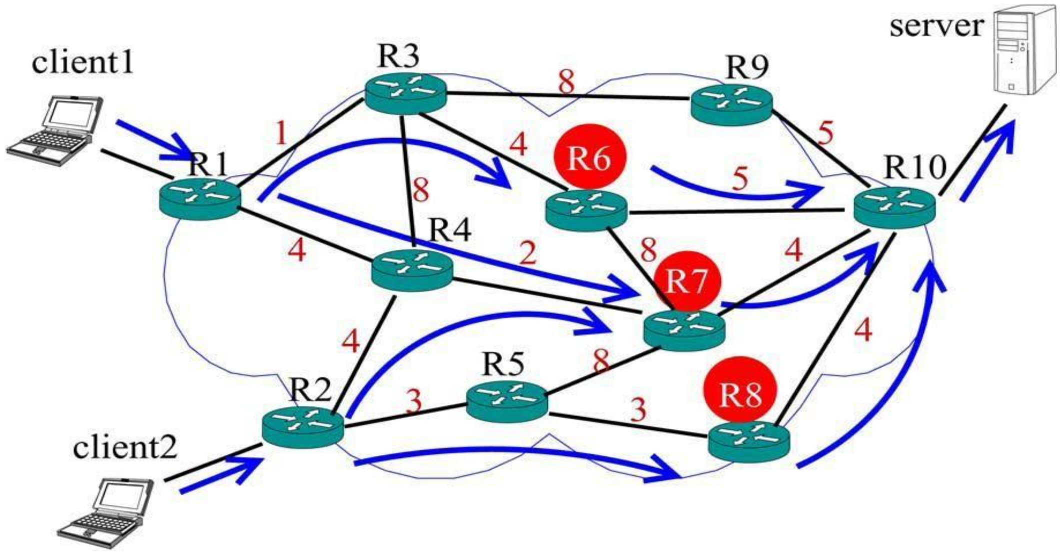
Network-based LEQ service provides equalized-latency paths between the clients and servers by redirecting interactive application traffic from different clients along paths that minimize their delay difference.

LEQ Hub Routing: The network-based LEQ architecture is implemented using a hub routing approach: Using a small number of hubs in the network to redirect application packets, we equalize the delays for interactive applications. To explain the basic LEQ architecture, we consider a single administrative domain scenario and focus on equalizing application traffic delays between the different client edge routers and the server edge routers without considering access delay. Based on the application's LEQ requirements, the application traffic from each client edge router is assigned to a set of *hubs*. Client edge routers redirect the application packets corresponding to the LEQ service through the hubs to the destined servers. By redirecting through the hubs, application packets from different client edge routers with different delays to the servers are guaranteed to reach the servers within a bounded delay difference.

5. LATENCY EQUALIZATION ARCHITECTURE

LEQ routing in a single administrative domain. We achieve LEQ routing by selecting a few routers as hubs and directing interactive application traffic through these hubs. Next, we extend the basic LEQ architecture to support access network delay and multiple administrative domains (e.g., across a content distribution network and ISPs).

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 4 ISSUE 2 (2018) PAGES 2849 - 2854
RECEIVED : 25.02.2018 PUBLISHED : 21.03.2018
March 21, 2018



DISCUSSION AND ANALYSIS

We evaluate our LEQ routing architecture using both static and dynamic scenarios on ISP network topologies. In the static case, we only consider propagation delays, and this corresponds to the scenario of a lightly loaded network. We also evaluate the delay difference under different optimization policies both with and without compromising the delay of individual clients, and different network settings such as considering access network delay and multiple administrative domains. In the dynamic case, we evaluate the LEQ routing architecture under transient congestion. In each simulation scenario, we compare the performance of the LEQ routing scheme to that of shortest-path routing (OSPF).

SIMULATION SETUP

For our network simulations, we use large ISP network topologies such as AT&T and Telstra. These topologies were obtained from Rocketfuel. For the dynamic case, we consider the Abilene network topology. LEQ Without Compromising End-to-End Delay We first explore the potential of the LEQ routing architecture to discover latency equalized paths, under the optimization constraint that the end-to-end delays of individual clients are not compromised.

6. CONCLUSION

The LEQ routing architecture and algorithms presented in this paper clearly provide a pathway for networks to support scalable and robust multiparty interactive applications. Based on the evaluation of our LEQ architecture, we conclude that, with only minor enhancements to the edge routers, provider networks can easily support and enhance the quality of multiparty interactive applications.We show that the LEQ scheme can support different optimization policies that can achieve overall application performance in terms of latency equalization both with and without compromising end-to-end application latencies.

7. REFERENCES

1. Kapur, G.Wang, P. Davidson, and P. R. Cook, "Interactive network media: A dream worth dreaming?," Organized Sound, vol. 10, no. 3, pp. 209–219, 2005.
2. A. R. Greenwald, J. O. Kephart, and G. Tesauro,"Strategic pricebot dynamics," in Proc. ACM Conf. Electron. Commerce, 1999, pp. 58–67.
3. "Cisco telepresence solutions," Cisco, San Jose, CA [Online].
4. L. Pantel and L. C. Wolf, "On the impact of delay on real-time multiplayer games," in Proc. NOSSDAV, New York, 2002, pp. 23–29.
5. S. Zander and G. Armitage, "Empirically measuring the QoS sensitivity of interactive online game players," in Proc. ATNAC, Dec. 2004, pp. 511–518.
6. J. Brun, F. Safaei, and P. Boustead, "Managing latency and fairness in networked games," Commun. ACM, vol. 49, no. 11, pp. 46–51, Nov. 2006.
7. D. Bauer, S. Rooney, and P. Scotton, "Network infrastructure for massively distributed games," in *Proc. NetGames*, 2002, pp. 36–43.
8. A. Kapur, G.Wang, P. Davidson, and P. R. Cook, "Interactive network media: A dream worth dreaming?," *Organized Sound*, vol. 10, no. 3, pp. 209–219, 2005.
9. A. R. Greenwald, J. O. Kephart, and G. Tesauro, "Strategic pricebot dynamics," in *Proc. ACM Conf. Electron. Commerce*, 1999, pp. 58–67.
10. "Cisco telepresence solutions," Cisco, San Jose, CA [Online]. Available: http://www.cisco.com/en/US/netsol/ s669/networking_solutions_solution_segment_home.html.
11. L. Pantel and L. C. Wolf, "On the impact of delay on real-time multiplayer games," in *Proc. NOSSDAV*, New York, 2002, pp. 23–29.
12. D. Bauer, S. Rooney, and P. Scotton, "Network infrastructure for massively distributed games," in Proc. NetGames, 2002, pp. 36–43.