ChoiceNet Gaming: Changing the Gaming Experience with Economics

D. Brown, H. Nasir, C. Carpenter, O. Ascigil, J. Griffioen, and K. Calvert
Laboratory for Advanced Networking
University of Kentucky
Lexington, KY 40506
Email: [davidb,nasir,charles,onur,griff,calvert@netlab.uky.edu

***Abstract*—When playing online games, the user experience is often dictated by the performance of the network. To deliver the best possible gaming experience, game developers often find themselves developing work-arounds that try to mask the lack of control they have over of the existing TCP/IP Internet.**

***ChoiceNet*, an emerging future Internet architecture, attempts to give applications enhanced control (choice) over the service they receive from the network. In particular, ChoiceNet supports an *economic plane* in which applications can purchase services from any provider. Because providers are compensated, they are motivated to offer a variety of innovative, excellent services, enabling applications to select the service best suited for its needs. Instead of coding work-arounds, game developers can obtain precisely the network service that is needed to optimize the game experience.**

In this paper, we describe the emerging ChoiceNet architecture and show how computer games can benefit from the alternatives enabled by ChoiceNet. To demonstrate the benefits of the ChoiceNet architecture, we implemented a first person shooter game that uses ChoiceNet to "purchase" and then send data over the purchased path resulting in substantially lower latency than the default path. We describe the ChoiceNet services used to implement the game, and we present performance results that show a significant reduction in latency. We also show how ChoiceNet can be used to purchase reliable (non-lossy) communication paths that improve the user's experience.

I. Introduction

A result of the Internet becoming pervasive and ubiquitous over the last decade is that users and their applications are always online. The ability to "always be connected" has had a significant impact on the gaming industry. The explosive growth of mobile and online gaming [1], [2], for example, is a direct result of the fact that users are always connected. Even games that would not be considered "online games" rely heavily on the network for in-app purchases, advertising, content/scene generation, upgrades, documentation, tutorials, monitoring, and a variety of other services now embedded in games.

While the Internet has been hugely successful at enabling and supporting the gaming industry, the Internet's limitations have become evident in recent years as the demands for new network layer functionality have increased. The thin "waist of the hourglass" that made the Internet so successful is now viewed as having ossified and is hindering the Internet's ability to adapt to the ever-increasing demands of applications [3].

The inability of the Internet to offer new network layer services creates a variety of challenges for online game developers, forcing them to develop some rather novel work-arounds to deliver the desired gaming experience to users. For example, the Internet's inability to provide quality of service (QoS) guarantees to applications—e.g., bandwidth, latency, or loss rate guarantees—has caused game developers to design and embedded complex solutions in the game itself—e.g., parallel content retrieval, preloading/buffering, scene prediction, layered/graceful video degradation, etc). Furthermore, the lack of more complex in-network services such as caching, transcoding, compression, and encryption have caused gaming companies to build their own network infrastructure services to perform these tasks (e.g., deploying game servers at strategic locations in the network).

An emerging future Internet architecture designed to address the current Internet's inability to change, adapt, and enhance its network layer is the architecture ChoiceNet [4], [5]. The ChoiceNet architecture is based on the observation that the current Internet architecture provides little reason or motivation for Internet Service Providers (ISPs) to offer new network layer services. Unless there is an economic incentive for ISPs to design, deploy, and support new network layer services, they will not do it. Consequently, ChoiceNet includes an *economic plane* (in addition to the conventional control and data planes) in which users (or their applications) can discover and "purchase" new services deployed by ISPs. Moreover, users are not confined to services offered by their directly-attached ISP. Instead, they are free to purchase services from any provider, much like telephone customers are allowed to purchase long distance service from any long distance provider regardless of who their local phone company is. In ChoiceNet, the economic plane enables (1) providers to advertise their services in a "marketplace", (2) customers to search and discover services in the "marketplace", and (3) customers to purchase (i.e., pay providers for) services.

Because providers have direct access to customers and can be compensated for the services they invent and offer, they are motivated to develop and deploy new and innovative services at the network layer. Moreover, users have the ability to dynamically select the network service that meets their transmission needs and fits within their budget. Different levels of service will have different costs, and users can weigh the trade-offs between cost and performance and select the

This work is supported in part by the National Science Foundation under grant CNS-1111040

service that is best for them. In fact, two users of the same game may decide to play the game with different performance characteristics (and pay different prices for their different experiences).

The ChoiceNet approach opens up completely new options for game developers. Rather than trying to invent solutions to work around the limited "one size fits all" functionality offered by the current Internet, ChoiceNet game developers can instead leverage a wide range of network layer services and alternatives offered by ISPs to achieve precisely the type of network layer delivery required by the game or user.[1] At the same time, ISPs are able to receive financial compensation for offering new services and are motivated to offer services that can "compete" in the marketplace.

For example, consider a *network path service* that a game could invoke to find paths between the game client and the game server that meet a certain criteria—say round trip delays less than 50 ms—along with the (financial cost) of using those paths. Based on policies defined by the user or system administrator (such as "always use the path with the smallest RTT that does not exceed cost C"), the application would select and purchase (on behalf of the user) the appropriate path, and then invoke the service to transmit its packets along the selected low latency path thereby achieving the desired interactive responsiveness needed by the game.

As another example, consider the growing number of mobile games that place certain demands on the wireless network. Mobile devices often connect to only one wireless network—typically the one without data caps—irregardless of the requirements of the application that will be using the network. Existing mobile applications may be configured so that they will not run when they find themselves in a particular network context (e.g., Facetime may refuse to work over a cellular network), or even if they are allowed to run, they run poorly. In ChoiceNet, the game on the mobile device would be offered a set of alternative network services including multiple wireless networks to connected to, complete with different costs, caps, bandwidth, latency, loss rates, etc. If the game used relatively little bandwidth and was not concerned about bandwidth caps, it might select a path across the cellular network rather than the WiFi network because the WiFi is congested and has higher loss rate and latency than the cellular network.

In short, the ability to "market" fine-grained services (e.g., the ability to dynamically created a path designed specifically for an application) and have the applications dynamically select and pay for the service offers game developers and game players the ability to control their gaming experience in ways previously not possible.

In this paper we provide a brief description of the ChoiceNet architecture that makes alternatives and choices possible, and then focus our attention on the design of a first person shooter game in ChoiceNet. The game makes use of a network path service in ChoiceNet to identify (and use) the path that provides the best response time for users of the game. In particular, we describe the ChoiceNet path service and the mechanisms used to dynamically purchase and use paths across the network. We then present some performance results from our prototype implementation showing that overhead introduced by the economic plane is negligible and that the ability to select paths significantly changes the responsiveness of the game.

[1] Throughout the paper we will refer to users choosing the services they need. In reality however, we expect the choice will be made by the application based on policies set by the user or the system administrator, or encoded into the application by the game developer. In other words, users themselves will rarely be involved in choosing specific services.

II. ChoiceNet

The ChoiceNet architecture [4], [5] is designed to encourage alternatives and competition between network providers. It introduces the concept of an *economy plane* that allows users to directly access and compensate service providers who offer services useful to the user.

In ChoiceNet, all network layer functionality is defined in terms of *network layer services* that can be *composed* together to form more complex services. Example network layer service that an ISP might offer include *packet forwarding services* that forward a packet along a user-specified path, *data caching services* that can be used to store data in the network for later retrieval (e.g., useful for implementing content distribution networks), *transcoding services* that convert data into a format suitable for a particular endpoint (e.g., a mobile device), *encryption services* to provide security, and any other service an ISP wants to invent and offer. Network layer services can be composed together to form higher-level service abstractions. For example, the packet forwarding services at a set of routers can be composed to form a service that forwards packets along a particular path. Alternatively a forwarding service might be combined with storage services to deliver data to a node, but also cache the data to speed future access from that node or other nearby nodes.

Before using a service, an application must contact the marketplace to discover what services are available. Network layer services register themselves with the marketplace so that they can be found by applications. Having found a service, applications must "purchase" the right to use the service. The act of purchasing a service can occur in several ways. In some cases the service may be offered for free and require no payment. In other cases the user may need to provide some monetary value (e.g., a credit card number or bitcoins [6], [7]). In other cases, the a user may simply need to provide proof of membership (e.g., being a faculty member at a particular university) in order to receive service.

Having purchased a service, the application receives a token that can be used when invoking the service in the *use plane*. ChoiceNet tokens serve as a "proof of purchase" and are cryptographically generated so that they cannot be stolen or reused. *Use plane* services only perform the service if the token (proof of purchase) that accompanies the service request can be verified. Note that token verification can be done much more efficiently than the more complex exchange of value (e.g., processing a credit card) that occurred in the economy plane. As a result, services in the use plane are able to enforce the business relationships that have been established in the economic plane, thereby ensuring that paying customers receive the service they paid for while at the same time protecting the service from access by non-paying users.

For example, a provider may offer a *packet forwarding service* at each of its routers, and then compose them in various was to form *path services* that forward packets along paths having certain properties (e.g., high bandwidth paths, or low latency paths). Each path might be offered for a particular price, or, more likely, a user (application) might purchase the right to use a particular path service which comes with the right to use any of the paths that it offers. For example a game might purchase access to a path service capable of finding low-latency paths across the network. In response to a request for a low-latency path, the path service would return a list of tokens (proofs of purchase) needed to send packet through each of the packet forwarding services along the path found by the path service.

In short, ChoiceNet provides the infrastructure necessary to:

- Advertise services in the marketplace
- Discover services that meet the needs of the application.
- Select and compose services together to create application-specific services (e.g., a path from a particular source node to a particular destination).
- Pay for and provision (setup/reserve) the service.
- Use the service, including verifying the tokens (thereby demonstrating proof of purchase).

III. ChoiceNet and FPS Games

The goal of ChoiceNet is to foster the development of new and innovative services inside the network. One can imagine any number of new network layer services that would be highly beneficial to game developers including rather obvious services such as QoS routing/forwarding all the way up to highly specialized services such as in-network storage of game content or the dynamic creation and placement of game servers. In short, ChoiceNet has the potential to impact the gaming industry in a wide range of ways.

As an initial application of the ChoiceNet model to gaming, we have been exploring games that fall within the first-person shooter (FPS) genre as these games tend to have exacting requirements on network performance. Furthermore, the network profile of FPS games—namely very low latency, but minimal bandwidth requirements—is almost directly at odds with the needs of high-bandwidth, latency-forgiving media streaming that has dominated recent Internet infrastructure growth. In other words, many of the optimizations we see ISPs making today run completely contrary to the needs of serious FPS gaming.

To demonstrate ChoiceNet's utility in FPS gaming situations, we have been using the free and open-source first person shooter game Xonotic [8]. Xonotic's usual configuration is fairly common among traditional multiplayer FPS games—players log in to remote servers (typically either run by other players on their gaming computer or a dedicated server, but usually hosted by private individuals (i.e., other players) as opposed to a corporation). Gameplay is fast-paced action and requires low latency for optimal play (ideally well under a one hundred millisecond round trip time). When a player activates the in-game list of available servers they are, by default, sorted by "ping" times; that is, round-trip time to the available Xonotic servers. Xonotic's client-server communication, much like similar FPS games, is carried over UDP running on a IPv4 or IPv6 network.

To demonstrate the advantages of the ChoiceNet architecture, we implemented two ChoiceNet network layer services: (1) a *packet forwarding* service, and (2) a *path selection service*. Both services require "payment". The *packet forwarding service* forwards packets from a specified ingress port on a router to a particular egress port on a router. In order to use the service, the sender must include a token in the packet that proves to the router that the forwarding service has been paid for. The *path selection service* identifies paths through the network, purchases the packet forwarding functions needed to realize those paths, and then sells the paths to applications that need a particular type of path (e.g., a low-latency path). The path selection service advertises itself in the marketplace. Applications that want to send packets contact the marketplace to discover the available path services. They then send a path request to the path selection service requesting a path between a source and destination having some specified characteristics (say a latency less than N for some value of N). The path request must include payment, and the returned result includes a set of packet forwarding services along with the tokens required to use those services. The application then includes those tokens in the "source routed" packets it sends.

Because routers along the path must cryptographically validate the token carried in the packet before forwarding the packet, there is an additional processing overhead at each router. However, our experiments show that the processing overhead is negligible. On the VM-based routers used in our experiments, the cryptographic validation check amounts to roughly 11 microseconds of processing time at each hop. This additional processing time tends to be greatly overshadowed by latency gains of selecting a low-latency path.

The application (game) accesses the game server normally—that is, it is unaware of the fact that it is running on ChoiceNet. ChoiceNet functionality is provided via a wrapper library that is dynamically loaded when the game is run. The wrapper library consults a policy file to determine what type of path is best for the application. The wrapper library determines the preferred route to the destination (by requesting it from a ChoiceNet path service) and then intercepting network I/O calls from the game and appending (source routed) extension headers and tokens to outgoing packets. ChoiceNet routers along the way process these extension headers with response packets flowing back along the same path.

The benefits of ChoiceNet to games happens in the policy file which chooses the best path for each application. In the case of our Xonotic game, the policy file is configured to request low-latency paths. As a result, the path service will search for, and return, packet forwarding services along the lowest latency path between the specified source and destination. Note that other applications may be configured to purchase different types of paths. For example, a Netflix application may be configured to purchase high-bandwidth paths. Each application purchases services specifically tailored to its needs. Both can be run simultaneously, each receiving precisely the type of service it needs.

To test Xonotic on ChoiceNet, we first deployed ChoiceNet on two different GENI topologies: one with paths having various latencies, and another having paths with various loss rates. We then set up a Xonotic server running on one of the nodes in the GENI topology and then ran the Xonotic game client on our local desktop machine connected into GENI via a GRE tunnel. The following section describes our results.

IV. Experimental Results

To demonstrate the benefits of the ChoiceNet architecture, we implemented two different ChoiceNet network topologies. The first topology, shown in Figure 1 represented a network with multiple paths having different round trip delays between the Xonotic game client and the Xonotic game server. The second topology shown in Figure 3 represented a mobile user with two wireless networks each having a different loss rate. In both networks, there were multiple paths between the Xonotic game client (located on Host A) and the Xonotic game server (located on Host B).

We deployed ChoiceNet-enabled Click routers [9] on all non-host nodes in the network. Each click router supported our packet forwarding service – only forwarding packets that had been "paid for" by the sender. We also ran a path server on one of the nodes in the topology that purchased packet forwarding service in order to form low latency paths and high bandwidth paths that could be "sold" to users. To measure the latency actually experienced by applications (the Xonotic game), we used a UDP-based echo server (located at Host B) with an associated UDP echo client (located at Host A) which continually measured the round-trip time between A and B along the same path being used by Xonotic. Trials consisted of the UDP echo client generating one request per second over one hundred seconds.

A. Varying Round Trip Times

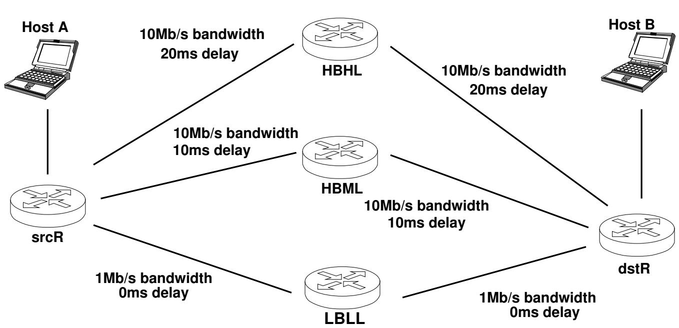

Fig. 1: Lossless topology with varying latency/bandwidth

The first topology, Figure 1, was designed with three paths between the game client and game server. Each path exhibited a different latency and bandwidth combination. In particular we constructed a high bandwidth high latency path (HBHL), a high bandwidth medium latency path (HBML), and a low bandwidth low latency path (LBLL). Each of the click routers along the path continually sent estimates of it current latency and available bandwidth to the path service, which in turn used that information to compute paths.

We then configured the policy file at Hosts A and B to identify the types of paths that are the best for each application to purchase. By default, ChoiceNet calculates routes through the network seeking out high-bandwidth routes with *reasonable* latency. This decision, while appropriate for many tasks (e.g., web browsing or streaming video) is sub-optimal for gaming, where latency and packet loss dominate all other factors.

To see the effects of ChoiceNet policy, we first ran a test using ChoiceNet's default settings (allowing it to select the high bandwidth/moderate latency path more appropriate to non-gaming applications). In this case ChoiceNet selected the "middle" path in figure 1. We then ran another test forcing ChoiceNet to choose the lowest latency route at the expense of maximum bandwidth for packets coming from or going to the simulated game server. In this case, ChoiceNet selected the "bottom" path in the topology.

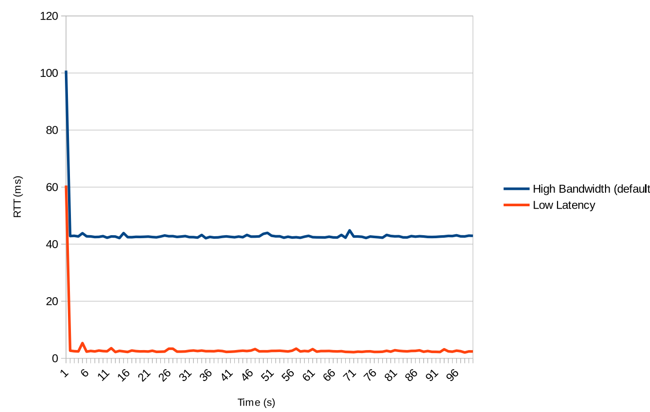

Fig. 2: Observed round trip time

While both settings saw a round-trip time spike on the initial request (due to the overhead of contacting the path service to calculate the most appropriate route through the network), both trials immediately stabilized to the typical latency of their respective routes through the network – the default setting trial showing an average of 43.3 ms per round trip and the low latency trial generating an average of 3.1 ms round-trip times.

B. Varying Loss Rates

In an attempt to understand performance over wireless networks, we considered topologies with differing levels of packet loss rates caused by wireless paths. Mobile devices are regularly faced with a choice of wireless network, each having different bandwidth, latency, and loss rates. Consider, for example, the topology shown in Figure 3. In this case, the mobile device can connect to access point 1 (AP_1) or access point 2 (AP_2) and will experience different loss rates depending on which access point the device connects to.

Existing mobile devices tend to select the wireless network to connect to via simple criteria; preference is often given to networks with higher bandwidth, or possibly to networks that do not have bandwidth caps (e.g., WiFi as opposed to cellular networks). In other cases it may be selected based on

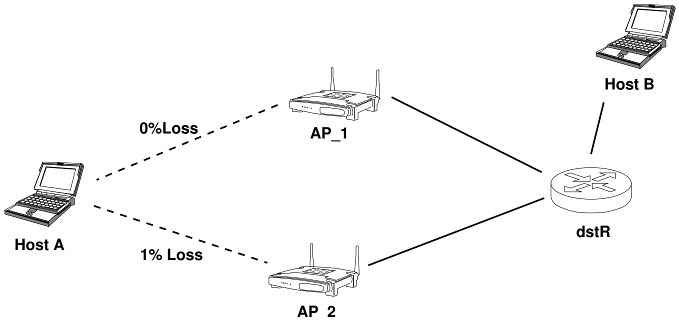

Fig. 3: Example topology with varying packet loss

signal strength regardless of the network's actual performance. In ChoiceNet, the user can specify the path it wants its packets to take on a per application basis.

Games, for example, may want to select the wireless network with the lowest packet loss rate as packet loss can lead to incorrect behavior of the game. Packet loss can be wildly variant over wireless network connections and can lead to frustrating gaming conditions. While poor latency can have unfortunate but predictable results, packet loss tends to lead to more prominent lapses in the client's view of the game world.

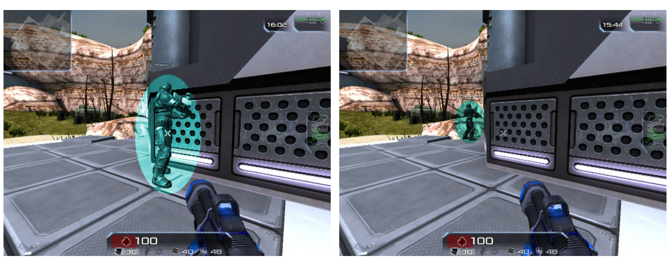

Fig. 4: Comparison of (a) erroneous rendering due to packet loss, and (b) correct rendering without packet loss.

Much like its ability to select low latency paths, the ChoiceNet path service could also select low-loss paths based on information it receives from routers. Figure 4 illustrates what can happen when a lossy wireless link is selected. In this example the loss causes the scene prediction algorithm to kick in and predict that the player (highlighted by the circle) continues on the same path and runs into a wall (Figure 4a). In reality the packet loss prevented the user from seeing that the player (again highlighted by the circle) changed directions and averted the wall (Figure 4b). In this case, selecting the reliable wireless link (0% loss) is preferable to the lossy link (1% loss) even if the bandwidth is less or the latency is slightly higher. Most importantly, through configuration of ChoiceNet, the user gains the *choice* as to which metrics should be prioritized.

V. CONCLUSION

The requirements for optimal user experience in games is often significantly different from media streaming. In the current Internet, providers scramble to provide more bandwidth to satisfy customers watching video streams—while low latency connections are what drive online gaming. Potential providers of low-latency interconnects lack financial motivation to build these links, and even if low latency paths existed, gamers lack the ability to choose the low latency path through the net to actually take advantage of the services offered by providers. Similarly, the rapid growth in mobile gaming is placing demands on the network that are often counter to the way mobile devices deal with wireless networks. Mobile devices are often configured to select the wireless connection that is free from bandwidth caps as opposed to the one that offers the lowest latency needed by interactive games.

To address this problem, we described the emerging ChoiceNet architecture and showed how ChoiceNet enables applications to select precisely the types of network service they need. At the heart of the ChoiceNet architecture is an economy plane where users "purchase" the specific services they need, allowing them to obtain the network layer performance they desire while at the same time motivating ISPs to offer the services needed by users and their applications.

To demonstrate the impact the ChoiceNet architecture can have on online games, we implemented an FPS game in ChoiceNet using the GENI testbed network. We demonstrated how the Xonotic FPS game was able to interact with the ChoiceNet marketplace to find and purchase low latency paths across the GENI topology that would minimize the delay experienced by players. We showed that the overhead needed to "verify payment" at each router along the path was negligible compared to the performance improvement obtained by using the lowest latency path. We also showed how ChoiceNet policies could be used to select low-loss wireless connections for mobile devices in situations where packet loss could lead to significant errors and artifacts in a multiplayer game.

In summary, ChoiceNet allows users to both provide payment to infrastructure providers as well as request specific services and performance from providers. As such we believe ChoiceNet is an effective solution to the lack of network services currently available to game developers—the economic model allows payment for use of infrastructure motivating its construction, and the choices themselves allow users and their applications to benefit from gaming-oriented network services that ISPs will now be motivated to support.

REFERENCES

[1] "U.S. Online Gaming Industry - Statistics & Facts," http://www.statista.com/topics/1551/online-gaming/.

[2] "2014 Sales, Demographic, adn Usage Data: Essential facts about the Computer and Video Game Industry," 2014, http://www.theesa.com/facts/pdfs/esa_ef_2014.pdf.

[3] S. Akhshabi and C. Dovrolis, "The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture," in *Proceedings of the ACM SIGCOMM 2011 Conference*, August 2011.

[4] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldine, and A. Nagurney, "Choice as a Principle in Network Architecture," in *Proc. of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM)*, August 2012, pp. 105–106, (Poster).

[5] T. Wolf, J. Griffioen, K. Calvert, R. Dutta, G. Rouskas, I. Baldine, and A. Nagurney, "ChoiceNet: Toward an Economy Plane for the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 44, 2014.

[6] "Bitcoin," https://bitcoin.org/.

[7] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf.

[8] "Xonotic," http://www.xonotic.org/.

[9] E. Kohler, R. Morris, B. Chen, J. Jannotti, and F. Kaashoek, "The Click Modular Router," *TOCS*, vol. 18, no. 3, pp. 263–297, 2000.