NDNGame: A NDN-based Architecture for Online Games

Diego G Barros

Universidade Estadual do Ceará (UECE)
Fortaleza-CE, Brazil
Email: diego.barros@larces.uece.br

Marcial P Fernandez

Universidade Estadual do Ceará (UECE)
Fortaleza-CE, Brazil
Email: marcial@larces.uece.br

***Abstract*—The content-oriented network paradigm is an alternative approach for computer's networks, proposing a new communication architecture compatible with the dynamic nature of the current Internet. Among these models, we can mention Named Data Network (NDN). Its basic idea is to retrieve data through content names, instead of source and destination IP address. Using in-network caches, this approach allows to achieve good performance to distribute content in large-scale, improving the usage of the network. However, this model is not a consensus on end-to-end applications such as e-mail, VoIP, games and client-server application. The NDN protocol overhead reduces the performance for these applications. This work proposes a hybrid network architecture in online games, using NDN for content dissemination and point-to-point IP communication to deliver control messages. Our proposal demonstrates how NDN networks can be used to improve online game's distribution network maintaining the user experience.**

Keywords–Future Internet; Online game architecture; Named Data Network (NDN).

I. INTRODUCTION

The Internet project was made 50 years ago, basically, focusing on point-to-point communication and technical users. Due to the telecommunication evolution and popularization of computers, the Internet became a successful, effective, global-scale communication.

As a consequence, new service demand and products were offered, like, e-commerce, social networks, file sharing, Voice over Internet Protocol (VoIP), online games, video stream, and others. However, to make this possible, the Internet's architecture suffered many amendments, the Internet infrastructure becoming more complex, increasing the cost for implementation, maintenance and management of these applications. This process is known as the Internet ossification [1].

Games are popular applications, and deliver a huge amount of multimedia content. The data flow in merely one game distributor [2] can reach 13.2 PB per week. Only in USA [3], 59% of North-Americans play some video game, spending in 2013 a total US$ 21.53 billions, out of which US$ 15.39 billions was just for content purchase. According to gamer's company demand, we believe Named Data Network (NDN) architecture could support games provider needs.

Online games impose a significant challenge in current Internet architecture. The huge amount of data delivered as scenario, video and images, and the necessity to provide a fast response to game commands brings new challenges to network researchers. It is easy to guarantee reduced packet delay for low bandwidth application. However, when we consider high bandwidth, the Quality of Service (QoS) guarantee becomes more difficult.

The online game application produces an incompatibility between models, the original Internet's architecture and current applications. IP packets were predicted for simple end-to-end applications, but the dynamic nature of the Internet requires more flexibility. The great content production, 500 exabytes in 2008 [4], is only one example. Considering the Internet growth rate, we estimate more than 1.5 zettabyte in 2014.

Today, the websites are evaluated by "what" content they contains. However, the Internet communications works in terms of "where" the content is. Then, the current architecture causes incompatibility issues to new applications, as availability, security and local-dependence.

Due to this scenario, some researches propose to reorganize the Internet's architecture. These proposals are divided into two types. The first type proposes small incremental changes, while the second proposes a redesign from scratch, changing the core principles. The second approach is known as Clean Slate [5].

Among the different approaches, we can highlight the content-based networks. In 2009, Palo Alto Research Center (PARC) defended a proposal Content-Centric Network (CCN) that today is known as NDN. The basic idea is to retrieve content by the name, instead of origin and destination address [6]. This approach has a new communications principle that improves abstraction and performance of networks.

Basically, a NDN node attends content requests through data sharing. This model supposes that each node can provide caching service, according to its resource dependence and policy.

Within this proposal, the NDN shows simplicity and flexibility with similar functions of current networks. However, among many advantages to this model, its main virtue is to improve content distribution. The NDN works on demand, improving performance and scalability. Thus, services with great content dissemination, e.g., video stream and online games, will be benefited.

However, NDN model is not well suitable for point-to-point applications. It is not clear how it provides efficiently traditional applications such as VoIP, e-commerce, online games, e-mail. Then, a pure deployment of NDN networks it is very improbable. Thus, we believe that the best way to deploy NDN is through an overlay network, e.g., torrent application to share content over IP infrastructure. Therefore, it is significantly important to validate the NDN approach to provide a good performance in generic web applications.

To overcome this challenge, this paper proposes a Hybrid Network Architecture for online games, using NDN for content dissemination and point-to-point IP communication to deliver control messages. Our proposal demonstrates how NDN networks can be used to improve online game's distribution network maintaining the user experience. A prototype over Mininet tool [7] running Quake 3 game server to evaluate the proposal was developed.

The rest of the paper is structured as follows. In Section II, we present some related work about real time application on content networks. Section III introduces the NDN concepts and Section IV presents the online game architecture. In Section V, we present the NDNGame Architecture, and its main blocks. Section VI shows the proposal evaluation, and finally, in Section VII, we conclude the paper and present some future work.

II. Related Works

A. Donny Brook

Donny Brook [8] is a game architecture based on a peer-to-peer model to run Quake 3. It aims to improve bandwidth, reducing the set of interest objects, as consequence, promoting a significant decrease in updating messages. Another relevant contribution is the use of a multicast message system, allowing multiple updating sources, sensitive to the response time. It also implements a load balance mechanism, where powerful nodes can support others. This work is important to our proposal, because it evaluates a valid Quake 3 implementation with similar goals, which is used as an alternative client-server architecture with enhanced performance using full advantage of network.

B. Voice over CCN

There is a meaningful importance in validating point-to-point applications, like email, VoIP, in Content-Oriented Network [9]. Jacobson et al. adapt a VoIP application to a content network [10]. They use an "on demand" publishing system, which serves as a contact point to the service, allowing users to initiate the session. Due to the use of names on NDN networks instead of IP's addresses, it was introduced the concept of constructable names, where is possible to build names of desirable contents without having seen the exact content name before. Thus, with a deterministic algorithm, the consumer and the producer can retrieve the same name on information available to both. This work serves as main reference to the use of a point-to-point application in NDN.

C. G-COPSS

GCOPSS is a distributed game platform that uses a Content-Oriented Network to deliver objects [11]. It adapts the COPSS [12] to improve scalability, which is an important goal on game's environments. It discovers network topology in order to offer an efficient system to disseminate the content. G-COPSS uses a hierarchical content descriptor and also implements a framework to provide content dissemination based on publishing requests. A user expresses interest in Content Descriptors (CD), e.g., /sports/soccer. The content publishers send announcements related to a specific CD when new parts of the content arrive. CDs are hierarchically organized. High-level users can receive announcements from users in a different level (lower), e.g., /sports receives /sports/soccer or /sports/swimming. NDN requires a new forward engine. The routers implement a Subscription Table (ST). STs maintain a CD base with subscriber's information, working in a distributed manner, as well using IP multicast to deliver content. This work shows the updating message exchange on an online game network using the NDN paradigm.

D. MERTS

More Efficient Real-time Traffic Support (MERTS) reinforces the importance to optimize NDN networks for point-to-point applications. Video and audio stream are much more sensible to network delay, thus, MERTS proposes a content classification in real-time and not real-time for on demand traffic. However, it is necessary to add a new field in NDN packets, modifying its basic structure. Our approach does not impose any modification on NDN design; instead, we propose a modification at the application layer, maintaining the NDN architecture.

III. Named Data Networks

The basic idea about Content-Oriented Network is not new; research like TRIAD project in 1999 and Data-Oriented Network Architecture (DONA) in 2006, already used content object name to forward packets [13]. In 2009, the PARC group published the proposal of content-centric architecture, which then became known as NDN [9]. Nevertheless, the NDN model stands out since it does not need an origin and destination address like IP on traditional networks. Therefore, an NDN network requires only the content name to retrieve it. This philosophy is simple and it can solve many problems like availability, security and location-dependence.

In order to understand this subject, it is necessary to understand how an NDN network works. However, before defining an NDN node, it is important to know that it works basically with only two packet types: Interest packet and Data packet. When a consumer needs a content, he expresses it by content name in the Interest packet; this packet is sent via broadcast over all network connections. The Data packet is the content which attends the Interest packet. Technically, it only occurs when both possess the same Content Name NDN works with the "face" concept, which is a reference of the requested origin, and it may be anything as an IP address, MAC, proxy, application, among others.

A NDN node is composed by the following entities:

1) Content Store: it works basically like a content buffer memory, storing the content disseminated by the network, but with a distinct replacement policy. NDN packets have an idem potent property, for different requests it may return the same result, like a Youtube video can satisfy user A, as well user B, C and D.
2) Pending Interest Table (PIT): it is essentially the table of interests not attended yet. When an interest is disseminated on NDN network, the correspondent PIT table in each node stores the interest name, and the face which it was requested. So, when a content matches the interest, it follows back the path described in each NDN node. This is what the authors calls "bread crumbs".
3) Forwarding Information Base (FIB): it stores information about potential location of content matching, forwarding I-packet to the data source. The NDN FIB is very similar to IP FIB table, but due to the NDN philosophy, it is not limited by spanning tree, it can use the advantage of multiple face's sources.

A longest-match lookup is done on its *Content Name* field every time an interest packet arrives on any face. There is a

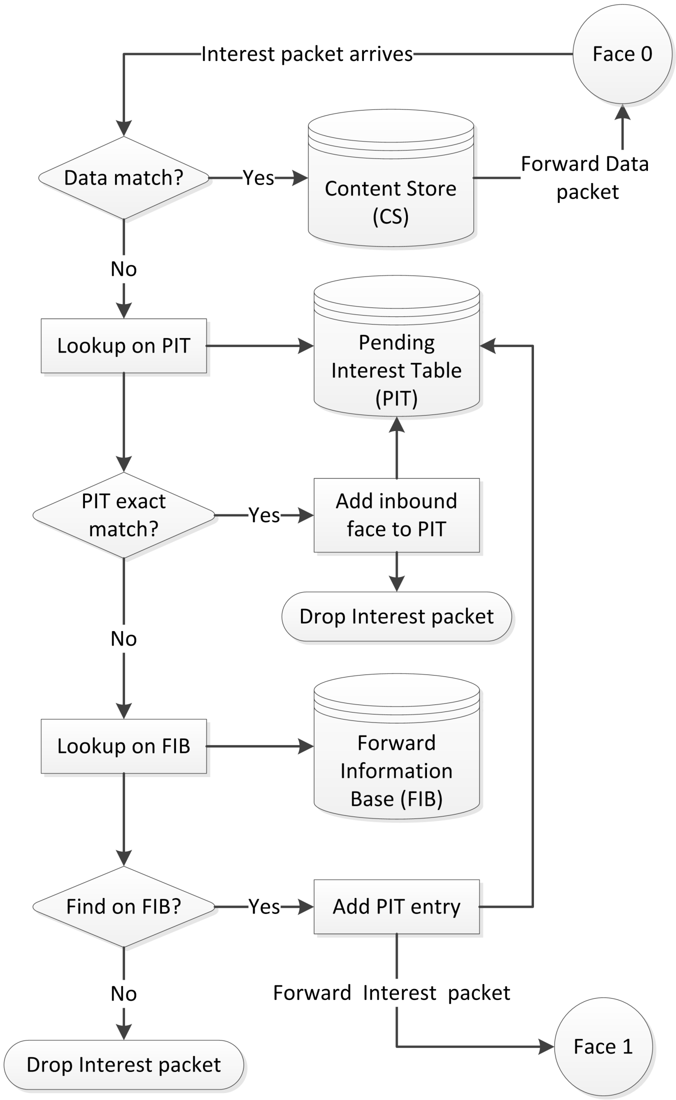

Figure 1. NDN basic operation

structure index to sort a precedence of search, first *Content Store*, next PIT and then, FIB.

If there is a data matching in Content Store, with the same prefix name, the content is sent back through arrival face. Otherwise if there are no data matching, it made a lookup in the PIT table. If there is a prefix match in PIT, the face is added in the Request Face List, and the Interest packet is dropped.

Otherwise, a lookup is done on FIB table. If there is a matching entry, next it includes an entry on PIT table, indicating the face where the pending interest was done. Then, the Interest packet is forwarded to the potential face pointed by FIB table.

If there is any matching entry, the Interest packet is dropped. This process is shown in Figure 1.

The data packet has a simple mechanism; it does not need to be routed. It just needs to track the path created by Interest packets in each NDN node. The path is traced, through PIT entries chain, until the origin request. It follows the "bread crumbs".

When a data packet arrives in NDN node, it is done a lookup by prefix name in Content Store. If there is an entry matching, it means a duplicate content, and then, this data packet is dropped.

Another packet discards occurs when the Data packet not match any entry in PIT, it means that this Data was not required before, it did not receive any Interested.

However, if there is a matching entry in PIT, the data were required by a face. The Data packet is authenticated and added to Content Store. Then, it is created a list with all faces that requested this Interest, and the Data packet is sent to each face in the list.

The treatment of Interest and Data packets, allows to retrieve content only by content name, that is simple and robust. Moreover, NDN is not limited by loops in layer 2; therefore, NDN take advantages of multiple face's sources, processing parallel requests. The NDN hop-by-hop information forwarding does not need to link layer 3 addresses to layer 2 identities, like IP and MAC address. Each NDN node can use information from the request packet. The request time and rate are able to measure the best way finding an interest.

Amendments initiated by the NDN model reflects it is more suitable to content distribution. According to evaluation by Jacobson's [6], showing that, comparing content dissemination performance between TCP/IP networks and NDN, the NDN approach does not increase the traffic according to the number of the users. Basically, for a unique client, TCP/IP was better than NDN. However, while the number of clients increase the total download time increases linearly proportional to the number of client. Otherwise, NDN network maintains download time constant with client number increases.

In NDN architecture, contents may be cached at intermediate nodes along the path from content providers to content users. This strategy, called *on-path cache*, provides contents near to users, reducing the bandwidth and the retrieval time. However, some works demonstrated that this strategy is not efficient because it may imply in a high content replication in nodes.

Another approach, called *off-path cache*, can reduce duplication maintaining the overall hit rate. This approach consists in three strategies: (1) caching only the most popular contents; (2) choose the best cache to push the content improving the hit ratio; and (3) redirect user's requests to the best cache.

The NDN network model shows a simple and flexible proposal, desired to dynamic nature of current Internet. However, it is very difficult to deploy this model purely, due to many open issues, like security and point-to-point applications like VoIP, video stream, e-mail, games, and others. Then, it is important to validate the NDN network model to solid web applications.

Thus, the best way to use NDN is over an IP infrastructure, to disseminate popular content and relevant size. The reduction in the download time for users, and better accuracy of investments, required by producers of content for allocation and content distribution, are benefits expected.

CCNx is the official implementation of NDN, which is currently under development. To be compatible with the existing architecture, CCNx builds an IP overlay to transport Interest and Data packets. The current version of CCNx is 0.8.2, and it supports Linux and Android platform [14].

 ISBN: 978-1-61208-398-8

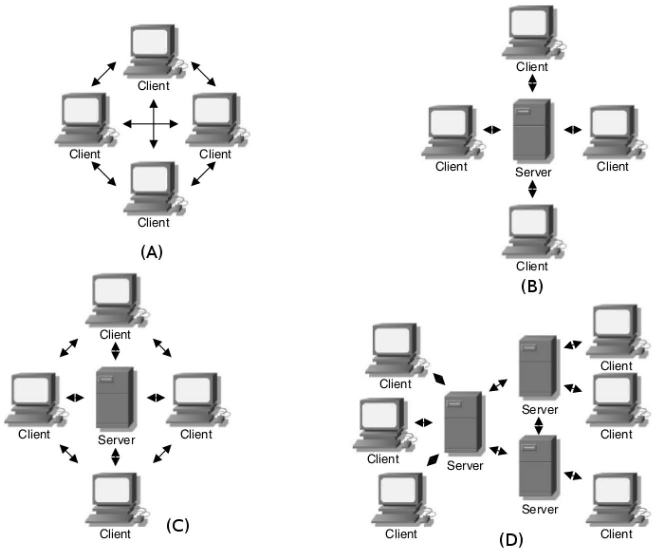

Figure 2. Game network architectures [18]

IV. Online Games Architecture

The most-recent game market research [15] have shown increased from 41 billion US dollars in 2007 to 68 billion in 2012, an increase of 10% per annum. In order to attend the market demand, was designed the Digital Distribution, where the content is delivered on a digital format, dispensing a physical media. This distribution scheme became more viable from 2000, accompanying the growth and evolution of the telecommunication network's and Internet bandwidth increase.

Most of the big gaming companies developed their own platform of distribution content. The games' distribution platforms were deployed, such as Steam, Origin, Live, PSN [16]. The basic idea is to provide a central service to storing contents in a digital format. Moreover, the platform also delivers other contents related to games, like movies and soundtrack. To provide QoS in a content distribution system, it is required a huge investment on network infrastructure. Steam has 8 millions costumers [17], and 13.5 PB of content per week in USA, representing 21.2% of global Internet traffic [2].

The games' market has popularity and a huge content dissemination, giving many opportunities to create new distribution architecture, e.g., using NDN networks.

A. Game Communication Architectures

Before presenting the NDNGame architecture proposal, it is necessary to know how the legacy network game infrastructure works. This section presents an overview of online game's evolution and some interesting issues used in this proposal. The communication model about network games is not different from the legacy network to distributed applications, i.e., peer-to-peer and client-server. However, each architecture produces little relevant modifications to our proposal. We can see an overview in Figure 2.

B. Peer-to-peer

In 1993, DOOM was the first First-Person Shooter (FPS) game with multiplayer cooperation mode up to four players. This game used peer-to-peer model in LAN over IPX protocol with broadcast transmission. Each player runs a game instance locally, and it should send messages to other players in a decentralized way [18].

The main challenges in the peer-to-peer model are related to synchronization. The game should be completely deterministic; each machine should have to execute the same set of instruction in a specific time interval, independent of network behavior. To guarantee this feature, it is necessary to wait a certain time until all players receive the messages in order to update the game state. Then, the game latency is equal for all players, i.e., the biggest delay of all players. This model has been used for Real-Time Strategy (RTS) games [19].

C. Client-server

The peer-to-peer game model works well over the LAN, but not on the Internet. The nature of the global Internet supposes that some users could have low bandwidth links producing long delays. As the game latency is the largest delay of all players, if one is connected in a high delay link, the game experience became bad for all players.

To overcome this problem, in 1996, Quake 3 was released using a client-server architecture [20]. In Quake 3 all players (client) send control messages and update messages only to one machine, a centralized communication server. The clients send to the server all necessary information to process the game state. The server receives and processes the next game state, and it sends a response to all players in order to update the client local state [19].

However, the client machine has just an approximation of actual game state, working like a "dumb terminal". Then, it is not necessary to guarantee deterministic game state in all player's machine; the real game state occurs only on the server. The game Quality of Experience (QoE) is directly related to latency between client-server. As the delay, inside network backbone is significantly lower compared to client bandwidth, the game experience in client-server architecture is better than peer-to-peer architecture. This model is adopted by most online games companies [21].

The evolution of the client-server model is the server-pool [21]. In this model, there is a server pool, in peer-to-peer or client-server architecture, connected to local servers near to the clients. A client can connect to the server pool, through the local server. Server pool increases the architecture complexity, but provides better scalability and game experience for users.

The last model is a combination of client-server with peer-to-peer model [18]. A hybrid network can provide the best from both worlds. The game control plane works on the central server, like traditional client-server. However, the clients are able to connect direct to exchange information and process the game state. This approach is used in VoIP and message chats.

V. The NDNGame Architecture

According to the previous section, the most-used model by game companies is the client-server. Problems like scalability and unique point of failure are recurrent on this approach [22]. For example, in a new game release or Downloadable Content (DLC), it is normal a huge increase on server's load. And this overload might cause failures and service interruption [23].

To reduce the failure risk, the game companies invest in network infrastructure like virtual machine allocations and cloud computing architecture.

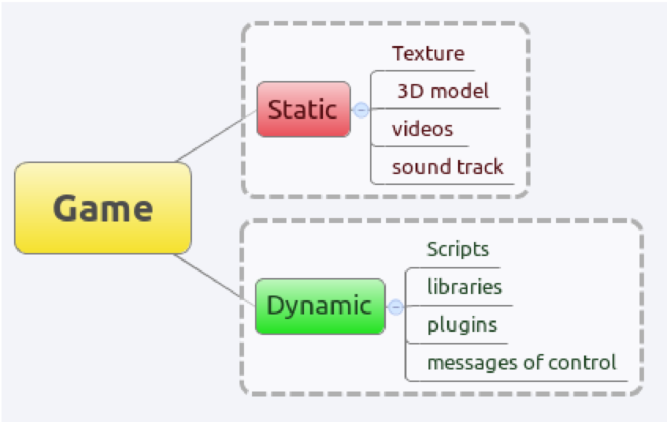

Figure 3. Abstract content

Due to these issues, our approach uses a hybrid network model. A unique local server can support a limited client number. However, if there is content sharing among clients, the local server will be able to support a large client number, i.e., more clients with same infrastructure.

The publisher releases the content on the local server to disseminate it to the first cache. Next, each client will be able to release same content on NDN network. Then, the content availability is directly proportional to demand. This approach is scalable and there is not any central point of failure.

A. *Content Classification*

The basic idea is to classify games content in two types: static and dynamic content. Figure 3 shows the content classification.

A multimedia game is composed by files related to environment and characters, as texture used to building maps and characters. Computer graphic scene and sound track also comprise this type of packet. This content represents most part of bandwidth, and it does not change from client to client. Storing and renderization are made locally using the player device. So, this content is declared as static, viable to share among all clients.

On the other hand, logical parts as scripts, libraries, plugins, control messages, have a dynamic behavior, change over time. They are dependent of hardware and client Operational System (OS), moreover, requires lower bandwidth.

Dynamic packets are extremely sensible to network latency. The response time of dynamic content is very important for QoE. According to Chen [24], the game-play time is reduced when network latency is increased. Thus, the best way to deliver this content is on traditional client-server networks based on TCP/IP, or even better, UDP/IP protocol.

The multimedia packets have other necessities. This content causes a great impact on server load, due to larger files. We believe the best strategy to work this content is to divide the responsibility among clients. The client community can share multimedia packets for the same game, reducing traffic on local servers.

In the architecture overview, we have a traditional client-server network, working on TPC/IP as base for a NDN overlay network. We basically split content traffic between static and dynamic, and forward it to the network layer which best attends to player's need. We can see an overview in Figure 4.

When a client purchases a new digital game, it downloads the game's dynamic pack with the authenticated files. The entire transaction is done by the traditional IP network. After that, the publisher sends a list of static content, which can attend to requests from the new recently deployed game. Cache's networks are composed by the local server and the clients who have the game or just part of it.

A client is also an NDN node, possessing a FIB. This database will be fed by the local server, updating the new cache availability. Thus, it there is not the necessity to broadcast the interest packets, it is only needed to forward them to caches, which have the requested content. This will help to avoid packet's flood, redirecting the data flow on the network. After send the Interest packet, clients wait for desired data while it shares the content already downloaded.

The NDNGame proposal provides some advantages:

- Low complexity, our proposal works on application's layer; it is not necessary to modify the basic structure of IP packets, neither NDN network core. In this manner, there is a great chance of success to deployment this model in the current game's market.
- Cost reduction: the content sharing by the users, provides a bandwidth reduction on content local servers. Thus, it is possible to attend more clients and to save infrastructure investments.
- Availability and Scalability: the content network infrastructures work on demand, thus it does not degrade if the number of player increases. Moreover, the availability also increases as the demand increases; more clients mean more available caches.

VI. Proposal Evaluation

To evaluate the NDNGame proposal was built two scenarios: (1) a legacy network with ten IP switches, showed in Figure 5, and (2) a hybrid NDN and IP network ten NDN switch with an IP routing table, showed in Figure 6. The scenario is a topology with ten switches in line. In each scenario, we consider one gaming server and ten players. The game traffic used was the Quake 3. The Quake 3 game application was chosen in order to compare against some related work, which uses it as reference game application. We evaluate the proposed NDNGame architecture against a legacy IP network.

The evaluation environment was based on a virtualized network using Mininet [7]. The Mininet system permits the specification of a network interconnecting "virtualized" devices. Each network device, hosts, switches and controller are virtualized and communicate via Mininet. A Python script is used to create the topology in Mininet, and the traffic flows controls are made by the OpenFlow controller. Therefore, the test environment implements and performs the actual protocol stacks that communicate with each other virtually. The Mininet environment allows the execution of real protocols in a virtual network.

The experiment was built over VMware Workstation 10. All tests run on a Dell PowerEdge R-620 with two Xeon processors and 64 GB RAM. To ensure the reliability of results, the workload in the server is maintained below 80% of processor capacity.

 ISBN: 978-1-61208-398-8

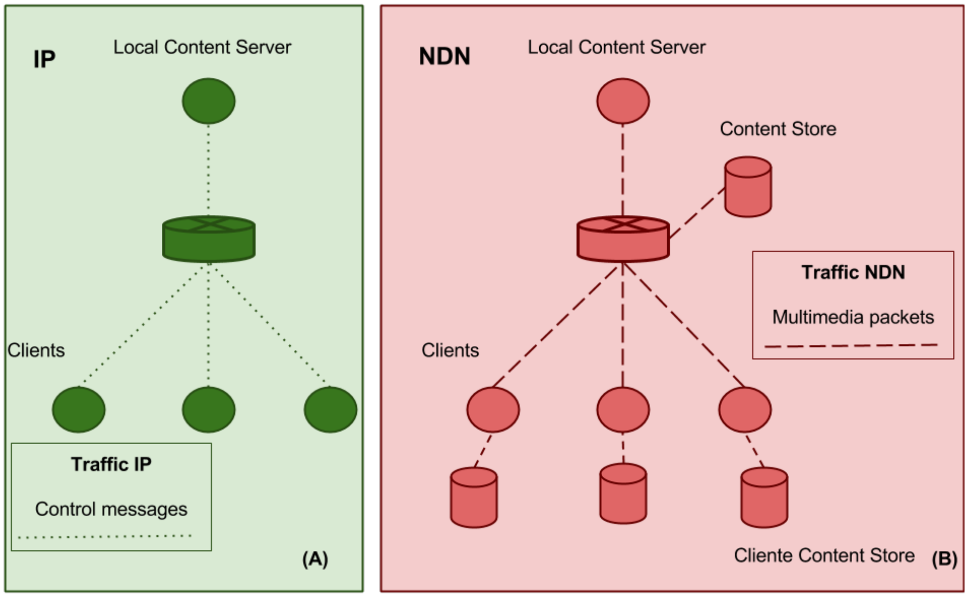

Figure 4. NDNGame architecture: (A) IP network and (B) NDN network.

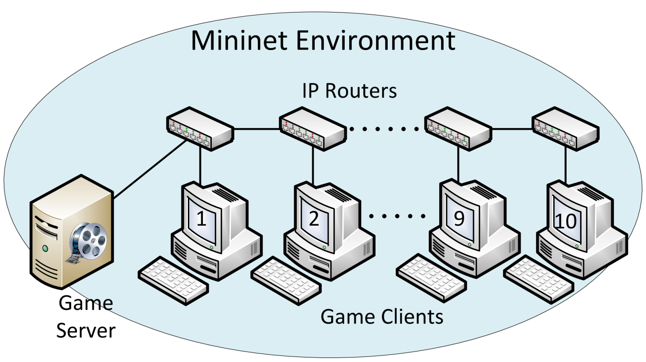

Figure 5. Legacy network evaluation scenario

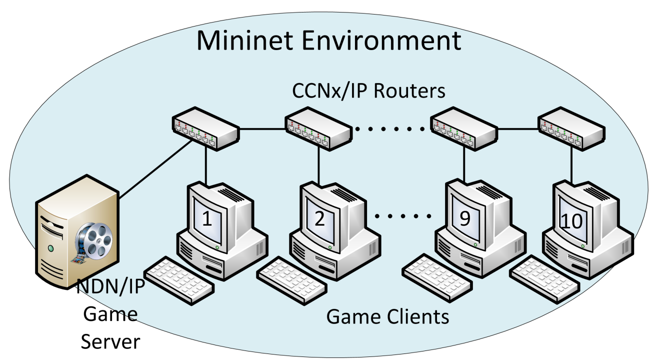

Figure 6. NDNGame evaluation scenario

The NDNGame prototype was built over Mini-CCNx environment [25], a CCNx implementation in Mininet. As CCNx is a CCN overlay over IP, it is possible to implement a hybrid network. The Quake 3 server forwards all static content packets to UDP port 9695. So, all 9695 UDP packets are treated as NDN packet across the overlay network. The Quake 3 game traffic was generated by D-ITG software [26] and a proxy inside servers classify static and dynamic content.

In initial evaluation, the results show an improvement in the delay of game content distribution when the number of users increase and a reduction on network and server load.

VII. Conclusion and Future Work

The NDN model works well to distribute massive amounts of static content. However, point-to-point application is not suitable to this model, due to NDN retrieve content by name instead an address. This problem makes it difficult for a pure NDN network deployment.

To overcome this barrier, this paper proposed a new hybrid network game architecture where NDN networks could be used to improve online game's distribution network. It is done applying NDN for content dissemination and point-to-point IP communication to transmit control messages. Our approach is simple and provides scalability and cost reduction maintaining the user experience.

A prototype was deployed, and the initial results showed that the proposal could reduce the network delay and reduce the network and servers' load, providing a better user experience.

As future work, we intend to evaluate the proposal in more diverse scenario and traffic workload. Another important issue is to analyze the impact of dual protocol stack, IP and NDN, in a unique switch.

REFERENCES

[1] J. S. Turner and D. E. Taylor, "Diversifying the internet," in Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE, vol. 2. IEEE, 2005, pp. 760–766.

[2] Steam, "Steam download stats," 2014, URL: http://store.steampowered.com/stats/content [retrieved: March, 2015].

[3] Theesa, "Essential facts about the computer and video game industry," 2014, URL: http://www.theesa.com/facts/pdfs/esa_ef_2014.pdf [retrieved: March, 2015].

[4] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC iView: IDC Analyze the Future, vol. 2007, 2012, pp. 1–16.

[5] A. Feldmann, "Internet clean-slate design: what and why?" ACM SIGCOMM Computer Communication Review, vol. 37, no. 3, 2007, pp. 59–64.

[6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009, pp. 1–12.

[7] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, ser. Hotnets '10. New York, NY, USA: ACM, 2010, pp. 19:1–19:6.

[8] A. Bharambe, J. R. Douceur, J. R. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang, "Donnybrook: enabling large-scale, high-speed, peer-to-peer games," ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, 2008, pp. 389–400.

[9] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves, "Content-centric networking," Whitepaper, Palo Alto Research Center, 2007, pp. 2–4.

[10] V. Jacobson, D. K. Smetters, N. H. Briggs, and et al., "Voccn: voice-over content-centric networks," in Proceedings of the 2009 workshop on Re-architecting the internet. ACM, 2009, pp. 1–6.

[11] J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "G-copss: a content centric communication infrastructure for gaming applications," in Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on. IEEE, 2012, pp. 355–365.

[12] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, "Copss: An efficient content oriented publish/subscribe system," in Architectures for Networking and Communications Systems (ANCS), 2011 Seventh ACM/IEEE Symposium on. IEEE, 2011, pp. 99–110.

[13] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, and et al., "A data-oriented (and beyond) network architecture," ACM SIGCOMM Computer Communication Review, vol. 37, no. 4, 2007, pp. 181–192.

[14] "CCNx Project," 2014, URL: http://www.ccnx.org/ [retrieved: March, 2015].

[15] F. Caron, "Gaming expected to be a 68 billion business by 2012," 2014, URL: http://www.arstechnica.com/gaming [retrieved: March, 2015].

[16] Wikipedia, "Digital distribution in video games," 2014, URL: http://en.wikipedia.org/wiki/Digital_distribution_in_video_games [retrieved: March, 2015].

[17] ——, "Steam software," 2014, URL: http://en.wikipedia.org/wiki/Steam_28software29 [retrieved: March, 2015].

[18] G. Armitage, M. Claypool, and P. Branch, Networking and online games: understanding and engineering multiplayer Internet games. John Wiley & Sons, 2006.

[19] GafferonGames, "What every programmer needs to know about game networking," 2014, URL: http://gafferongames.com/networking-for-game-programmers/what-every-programmer-needs-to-know-about-game-networking/ [retrieved: March, 2015].

[20] F. Sanglard, "Quake 3 source code review: Network model," 2014, URL: http://fabiensanglard.net/quake3/network.php [retrieved: March, 2015].

[21] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of networking in multiplayer computer games," The Electronic Library, vol. 20, no. 2, 2002, pp. 87–97.

[22] A. Tanenbaum and M. Van Steen, Distributed systems. Pearson Prentice Hall, 2007.

[23] R. Appleton, "Titanfall's pc servers are overloaded," URL:http://www.gamerheadlines.com/2014/03 [retrieved: March, 2015].

[24] K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" Communications of the ACM, vol. 49, no. 11, 2006, pp. 34–38.

[25] C. Cabral, C. E. Rothenberg, and M. F. Magalhães, "Mini-CCNx: fast prototyping for named data networking," in Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking. ACM, 2013, pp. 33–34.

[26] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-itg distributed internet traffic generator," in Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings. First International Conference on the. IEEE, 2004, pp. 316–317.