2016 28th International Teletraffic Congress

Optimizing Time to Exhaustion in Service Providers Using Information-Centric Networking

Ali Shariat*†, Ali Tizghadam*, Alberto Leon-Garcia*
*University of Toronto, Canada
†ali.shariatmadari@utoronto.ca

***Abstract*—Exponential traffic growth due to the increasing popularity of Over-The-Top Video services has put service providers under much pressure. By promoting in-network caching, Information-Centric Networking (ICN) is a promising paradigm to answer current challenges in the service provider's domain. This paper reports on a cache placement strategy for service providers to delay the onset of congestion (time-to-exhaustion) to the extent possible in order to optimize their capital expenditure for their limited capacity planning budget. We show that even a limited deployment of ICN provides a substantial increase in the time-to-exhaustion of the network and a decrease in the number of links with high utilization.**

I. INTRODUCTION

Current Internet is a product of four decades of evolution. Today, Internet traffic is rapidly growing due to Over-The-Top (OTT) and Video-on-Demand (VoD) services such as Netflix and YouTube. Video traffic is now consuming most of the bandwidth on the Internet. A more detailed analysis shows that Netflix (31.6%) and YouTube (18.7%) combined, account for over 50% of downstream traffic in fixed access [1]. This growth is changing the architecture of the Internet. The content creators are exploiting the economies of scale and using *Content Delivery Networks* (CDN) to transfer this huge traffic, which exacerbates the change. CDNs were introduced to overcome the limitations of traditional Web caching systems by deploying several caches throughout the globe and populating these caches with the popular content during the off-peak traffic hours. Some content providers are very keen to work with Service Providers (SP) to provide these caches. For example, Netflix OpenConnect program is rapidly expanding its coverage by offering to install and maintain the caches in the SP's network.

The SPs usually place CDN caches at the Internet Exchange peering points that connect to the core of the SP's network. Therefore, the CDN architecture is optimized to deliver the content until it reaches the last mile of its path, but it does not solve the inefficient use of the SPs network infrastructure. Even when all of the users are requesting the same content, that content is transmitted over the network of the SP multiple times.

New networking paradigms such as *Information-Centric Networking* (ICN) provide solutions to this problem. The ICN model has *named-data* at the core of the networking, and names are *decoupled* from location, applications, storage or media of transport. This solution is not only agnostic about the source of the content but also gives us the capability of *in-network caching* for all contents. In-network caching will help place popular content near the consumer to lower the latency and results in a better utilization of the infrastructure and increased throughput [2]. Here we will focus on the Named Data Networking (NDN) implementation of ICN [3]. In this work, we use *time-to-exhaustion* (TTE) as our metric and formulate the problem to place the caches in the network in a way that TTE is maximized.

The rest of this paper is organized as follows. Section II reviews previous works on network design and cache placement. In Section III, the content delivery problem in a service provider is investigated. Further, details of our analytical model are discussed in Section IV. Simulation results and validation are provided in Section V.

II. RELATED WORKS

Content Delivery Networks provide multi-server to multi-client paradigm for everyone. CDNs forward the requests from clients towards the best server. This process heavily relies on DNS, which raises several issues [4]. To better serve the users, CDNs are putting their own caches inside the network of service providers. This architecture is a win-win situation for both the content providers and network operators. For example, Netflix Open Connect program is rapidly expanding its coverage, since they offer to install and maintain the caches for free in the SP's network. Deploying *transparent* cache servers directly in the network is another method that is employed by the SPs. This method has the benefit that SPs have full control over the configuration and placement of the caches. Placing a transparent web cache in different topologies is studied in [5]. The authors model line and ring networks and experimentally study a single web cache case.

Information-Centric Networking [3], [6] is a clean slate networking paradigm that tries to solve current networking problems by replacing the host-to-host communication model and introduces concepts such as *Naming*, *Name-based routing* and *In-network caching*. Named-Data Networking (NDN) [7] is a fully-fledged ICN architecture. In NDN, content is moved and cached between neighbors and when there is an interest for a content, the network will *strategically* find and forward the interest towards the content store. There is a wealth of literature on cache deployment in the context of ICN [8], [9], [10] some with contradictory results. The authors in [11] provide an analytical model of the cache miss probability

978-0-9883045-1-2/16 $31.00 © 2016 ITC

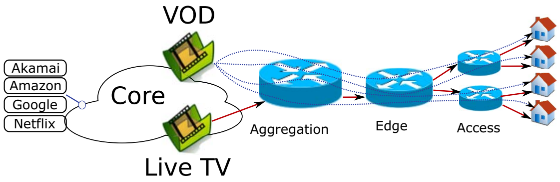

Fig. 1: Content distribution in Service Providers

of a single caching system and extend it to a network of caches. In [10], the authors use different centrality metrics for sizing storage in a content-centric networks, but couldn't find an incentive for heterogeneous caching. The authors in [12] solve a budget constrained caching problem in Content-centric networking context and note that topology has a significant impact on the optimal cache placement. They have considered hop counts as the base metric for optimizing cache placement. Reference [2] studies the evolution of CDN and its challenges and shows how ICN paradigm can help overcoming them.

III. PROBLEM DEFINITION

A. Network of a Service Provider

A service provider network (Fig. 1) usually consists of multiple layers. The core of the network transfers the highest volume of data from various aggregation sites between sources and destinations. Content servers are usually connected to the core through an Internet exchange peering point. The next level is the Aggregation level and is a concentration point of multiple distribution centers, which themselves may be connected to smaller distribution edge centers. The final layer is called Access layer and is directly connected to the consumers. For example, an access layer of a wireless service provider contains several cellular antennas.

Now consider the subscribers that request an OTT video content. For every request for a content, a new TCP connection is created between the content source and the consumer's machine. Even when all of the users are requesting for the same content, that content is transmitted over the network multiple times. Note that the source of this content may be either controlled by the operator itself or come from a VoD content server owned by a 3rd party CDN. If the content source is a live stream from outside of the network, operators are faced with an even bigger challenge than for VoD content. Live stream content is watched by many consumers at the same time and the operator does not have any control over the content if it is originated from outside of the network. As one can see, this structure is not scalable and is a clear waste of underlying resources.

B. Time-to-exhaustion

In a network with increasing demand, such as service providers, congestion is inevitable. For example, assume that the demand is increasing by 5% every month. Fig. 2 shows such a scenario. If the current demand in the network is 100Gb/s and the network can handle a maximum of 400Gb/s, the current infrastructure will keep up with the traffic for the next 27 months. To maintain a congestion-free network, while the demand is increasing, a service provider has to invest in its infrastructure.

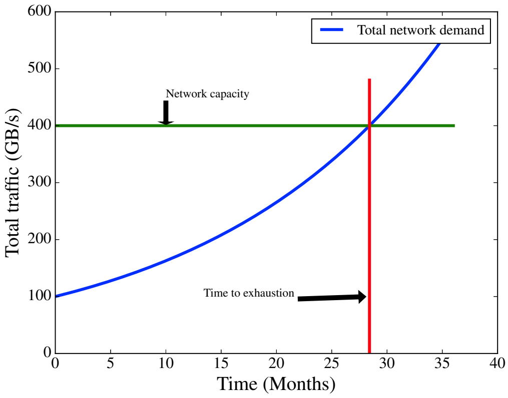

Fig. 2: Time-to-exhaustion. Traffic is increasing monthly until network is congested.

For a service provider, serving content from a local cache or peering point incurs different costs. In addition, increasing link capacity costs much more than increasing caching storage. At the same time, serving more content to users means more revenue. This demand increase will eventually exhaust the network at some point in the future (Fig. 2) unless the network onset of congestion threshold is increased. The network onset of congestion of is when the capacity of some link in the network is exceeded (congestion) and depends on different factors such as network topology, routing and link capacities.

The problem that SPs are facing is how to plan their future network to accommodate the constant increasing of the demand, to provide a congestion free network and to minimize the costs. Another challenge is that the SP already has an established network. This investment can keep the network congestion-free for a limited time. Therefore, TTE becomes crucial for network capacity planning since it affects the amount and the timing of investment in the infrastructure. With a limited budget, the SP must choose how to plan the additional capacity, where to put the caches and what content should to cached. We aim to maximize the time-to-exhaustion, considering a limited budget, by placing caches in the best locations. We will show that using NDN will outperform optimal cache placement and routing in CDN and will prolong time-to-exhaustion of the network.

IV. PROBLEM FORMULATION

We model our network as a directed graph $\mathbb{G}(\mathbb{V}, \mathbb{E})$ with the set of nodes $\mathbb{V}$ and links $\mathbb{E}$. $\mathbb{U}$ is the notation for the set of nodes that have a demand for contents. $\mathbb{P}$ indicates the set of nodes that can satisfy demands for contents, e.g Internet exchange points. The set of nodes that are candidates for caching contents is noted by $\mathbb{C}$.

A. Demands and Storage Budget

To find the TTE of the network, we will model the network for one time epoch. We assume that within this time epoch, the demands are known and fixed, but the location of the caches, the cached content and content routing are not. We also assume a limited storage budget, $\mathbb{B}$, is available for capacity planning of all the caches in the network. For each budget value, we solve a series of *feasibility* problems by increasing the demands following a pre-known pattern. A feasibility problem does not have any objective and will only find a feasible solution to the problem. By increasing the demand, the network will become congested, which means the problem will become infeasible. When the problem becomes infeasible, i.e. the network is saturated, we have found the TTE of the network. Final solution of the model provides a content caching policy and traffic routing that maximizes the TTE of the network.

1) Demands: We denote the demand at each node i for content k by α_i^k. The demand at each node also depends on whether the node caches any content or not (h_i^k). In other words, the traffic of populating a cache is also a demand. Therefore, total demand at node i can be written as:

$$\beta_i^k = \alpha_i^k + h_i^k \quad \forall i \in \mathbb{C} \cup \mathbb{U} \tag{1}$$

Note that β_i^k is the number of the requests for content k, not the size of the demand. The size of the demand is $L_k \beta_i^k$ where L_k is the size of content k.

2) Storage Budget: Each cache in the network is assigned a part of the storage budget, denoted by S_i. Therefore, the budget constraint can be written as Eq (2).

$$\sum_i S_i \leq \mathbb{B} \quad \forall i \in \mathbb{C} \tag{2}$$

Let p_i be the binary variable that decides if node i is a cache and h_i^k the binary variable that shows if content k is cached at node i. For each cache, total cached objects can not exceed the size of the storage of that cache as written in Eq (3).

$$\sum_k L_k h_i^k \leq p_i S_i \quad \forall i \in \mathbb{C} \tag{3}$$

Also, total number of caches placed in the network can be limited by an upper limit, M, as written in Eq (4).

$$\sum_i p_i \leq M \quad \forall i \in \mathbb{C} \tag{4}$$

To have homogeneous caching, we may also add a constraint that enforces all S_i to be equal.

3) Cache Replacement Policy and Routing: Caching policy provided by the solution will maximize the TTE of the network. As mentioned earlier, h_i^k is the binary variable that shows if content k is cached at node i. Solving the model for two different time epochs with different demands will result in different h_i^k. The difference between h_i^k for different demands will be the cache replacement policy of node i. Adopting a certain caching replacement policy, such as Least Recently Used (LRU) or Least Frequently Used (LFU), will reduce the TTE of the network.

Constants	
$\mathbb{V}$	Set of nodes
$\mathbb{E}$	Set of directional links
$\mathbb{G}(\mathbb{V}, \mathbb{E})$	Graph of the network
$\mathbb{P}$	Subset of nodes that are connected to peering points
$\mathbb{U}$	Subset of nodes that have a demand for contents
$\mathbb{C}$	Subset of nodes that can cache contents
L_k	Size of the content k
α_i^k	Demand for content k at node i
Γ_i^+, Γ_i^-	Set of ingress and egress neighbors of node i
r_i^k	Maximum rate node i can read content k from its cache
$\mathbb{B}$	Total storage available for all caches
$c_{i,j}$	Capacity of link (i,j)
$\mathbb{I}(.)$	Indicator function, 1 if the condition is true, 0 o.w.
M	Maximum number of caches in the network
$\phi_{i,j}^{sd}$	Shortest-path betweenness of link (i,j) from node s to d
Common Variables	
S_i	Storage at node i
p_i	Decision variable for cache placement at node i
h_i^k	Decision variable for caching content k at node i
β_i^k	Total demand by node i for content k
CDN Specific Variables	
$f_{i,j}^{kd}$	Flow for content k on link (i,j) going to node d
γ_s^{kd}	Traffic flow from node s to node d for content k
NDN Specific Variables	
$f_{i,j}^k$	The rate interests for content k is sent on link (i,j)

TABLE I: Notations

The solution also provides the content routing policy for the network. Adopting a routing protocol such as shortest-path will reduce the TTE of the network. We study this effect in the result section.

B. Content Delivery Networks

In service providers, transparent caching is done by putting one or more caches in the network and re-routing the requests towards them. SPs may also host the content sources of their own or from third parties. To model this, we define a multi-commodity flow problem.

1) Flow Conservation: The flow conservation at node s for content k can be written as Eq (5).

$$\sum_{j \in \Gamma_s^-} f_{s,j}^{kd} - \sum_{j \in \Gamma_s^+} f_{j,s}^{kd} = \gamma_s^{kd} - L_k \beta_s^k \delta(s-d) \quad \forall s, d \in \mathbb{V} \tag{5}$$

We denote $f_{i,j}^{kd}$ as the flow for content k on link (i,j) going to node d and γ_s^{kd} for the flow for content k from node s to node d. The left-hand side of Eq (5) is the difference between total egress (Γ_s^-) and ingress (Γ_s^+) flows for content k at node s that is destined for node d.

The right-hand side of Eq (5) is the total flow that is originated at node s towards node d for content k minus the demand at node s for content k. $\delta(i)$ is the Kronecker delta function, it is equal to 1 when i is zero, otherwise it is zero. Therefore, $L_k \beta_s^k$ in Eq (5) will only have any effect when s and d are the same node. In other words, the ingress and egress flow destined to node d at any node other than d is equal to the traffic produced at that node for node d. When s and d are equal all ingress traffic into node d will be equal to the demand at node d. Therefore, considering the fact that node d does not sent traffic to itself (i.e. $f_{d,j}^{kd} = 0, \forall j$ and $\gamma_d^{kd} = 0$), Eq (5) will be reduced to

$$\sum_{j \in \Gamma_d^+} f_{j,d}^{kd} = L_k \beta_d^k$$

2) Cache Population Traffic: The cache population traffic is satisfied by peering points. Therefore, the total demand originated at the core ($\mathbb{P}$) of the network, must be bigger than the size of the cached content (Eq (6)).

$$\sum_{s \in \mathbb{P}} \gamma_s^{ki} \geq L_k h_i^k \quad \forall i \in \mathbb{C} \tag{6}$$

3) I/O and Link Capacity Limits: A node can only become a source of the flow for a content request when it is a cache and it has the content cached. $\mathbb{I}(i \in \mathbb{C})$ in Eq (7) is equal to 1, if only node i is a cache candidate. h_i^k will be equal to 1 when the content k is cached at node i. r_i^k is the rate that each node can read contents from its cache storage and put them on the wire. It is the limitation of the node's hardware, e.g. I/O limit of the node's hard disks.

$$\sum_d \gamma_i^{kd} \leq \mathbb{I}(i \in \mathbb{C}) r_i^k L_k h_i^k \tag{7}$$

We also write the link capacity constraint as Eq (8).

$$\sum_{k,d} f_{i,j}^{kd} \leq c_{i,j} \tag{8}$$

The complete feasibility problem that models a CDN in the network of a service provider is:

$$\begin{gathered}
\text{solve} \\
\text{subject to} \\
\sum_{j \in \Gamma_s^-} f_{s,j}^{kd} - \sum_{j \in \Gamma_s^+} f_{j,s}^{kd} = \gamma_s^{kd} - L_k \beta_s^k \delta(s-d) \quad \forall s, d \\
\sum_{s \in \mathbb{P}} \gamma_s^{ki} \geq L_k h_i^k \quad \forall i \in \mathbb{C} \\
\sum_d \gamma_i^{kd} \leq \mathbb{I}(i \in \mathbb{C}) r_i^k L_k h_i^k \quad \forall i \in \mathbb{V} \setminus \mathbb{P} \\
\sum_{k,d} f_{i,j}^{kd} \leq c_{i,j} \\
\beta_i^k = \alpha_i^k + h_i^k \quad \forall i \in \mathbb{V} \setminus \mathbb{P} \\
\sum_i S_i p_i \leq \mathbb{B} \\
\sum_k L_k h_i^k \leq p_i S_i
\end{gathered}$$

4) Shortest-path routing: Routing in the network of the service providers is usually based on shortest-path routing. To study the effects of shortest-path routing, we add a routing constraint to our model. Shortest-path routing is modeled using the shortest-path *betweenness centrality* of each link.

Betweenness centrality (BC) is one of the centrality metrics in graphs [13]. Betweenness centrality measures the degree to which a node or a link is needed when connecting other nodes along paths. Shortest-path betweenness centrality of the link (i,j) with respect to the source node s and the destination node d, denoted as $\phi_{i,j}^{sd}$, is defined as the proportion of the number of shortest paths from node s to d that passes through link (i,j). Therefore, the average traffic for content k that passes through link (i,j) from source s to destination d can be written as $\phi_{i,j}^{sd} \gamma_s^{kd}$. To model shortest path routing we can add Eq (9) to the model. Eq (9) will have the link (i,j) to not transfer any traffic more than its share, if the routing is done using shortest-path.

$$f_{i,j}^{kd} \leq \sum_s \phi_{i,j}^{sd} \gamma_s^{kd} \quad s \in \mathbb{V} \tag{9}$$

C. Named-Data Networking

1) Interest Forwarding: To model NDN, we will find the locations that potentially can satisfy more *interest* in contents. This notion of interest here is more of the nature of content popularity in a node, similar to the virtual interest packets studied in [14], and is different from the Interest packet in NDN paradigm. We denote $f_{i,j}^k$ as the rate that interest for content k is forwarded on link (i,j). Since NDN is a point-to-point protocol we do not have flows from sources to destinations, but potential interests that move around the network until they are satisfied.

Suppose node s has some interest in content k (β_s^k). Therefore, the egress interests ($\sum_{j \in \Gamma_s^-} f_{s,j}^k$) from node s is increased by β_s^k. This is written as an inequality in Eq (10).

$$\sum_{j \in \Gamma_s^-} f_{s,j}^k - \sum_{j \in \Gamma_s^+} f_{j,s}^k \leq \beta_s^k \tag{10}$$

Now consider a node that has a content cached in its content store and can satisfy interest for that content and remove the interest from the network. Each node also has an I/O limit for reading its content store that limits the rate interests are satisfied. Otherwise, the interest will be forwarded towards other nodes in the network. Therefore, a node can at *most* satisfy the interests by the rate that is bounded by its I/O limit, as written in Eq (11).

$$\sum_{j \in \Gamma_s^-} f_{s,j}^k - \sum_{j \in \Gamma_s^+} f_{j,s}^k + \mathbb{I}(i \in \mathbb{C}) r_s^k h_s^k \geq \beta_s^k \tag{11}$$

Consider the scenario that node s is not caching content k. Therefore, Eq (10) and Eq (11) will be reduced to an equality and will enforce that node s forwards all of its ingress and local interests. However, if node s caches content k, the ingress interests can be satisfied by an amount bounded by the hardware limitations of node s. Finding the movement of this potential interest in the network can be used to find the best place to cache the content.

2) Link capacity limit: The next step is to model the link capacity constraint. In NDN, Data packets follow the reverse path of the Interest packet to reach the destination. Therefore,

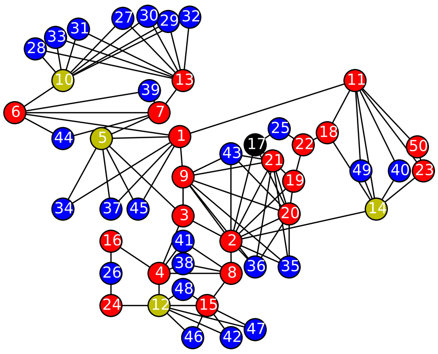

Fig. 3: Rocketfuel network

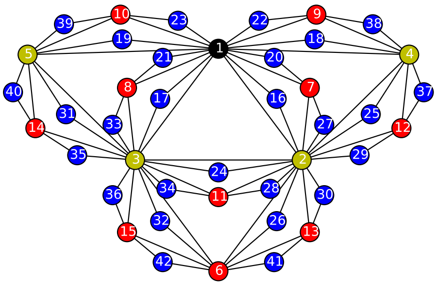

Fig. 4: DGM network

sending an interest over the link (i, j) will result in the data sent back over the link (j, i). We can use this to write link capacity constraint as Eq (12).

$$\sum_k L_k f^k_{i,j} \leq c_{j,i} \tag{12}$$

Including Eq (1), Eq (2), Eq (3), the complete feasibility problem for NDN is:

$$\begin{gathered}
\text{solve} \\
\text{subject to} \\
\sum_{j \in \Gamma^-_s} f^k_{s,j} - \sum_{j \in \Gamma^+_s} f^k_{j,s} \leq \beta^k_s \\
\sum_{j \in \Gamma^-_s} f^k_{s,j} - \sum_{j \in \Gamma^+_s} f^k_{j,s} + \mathrm{I}(i \in \mathbb{C}) r^k_s h^k_s \geq \beta^k_s \\
\sum_k L_k f^k_{i,j} \leq c_{j,i} \\
\beta^k_i = \alpha^k_i + h^k_i \quad \forall i \in \mathbb{V} \setminus \mathbb{P} \\
\sum_i S_i p_i \leq \mathbb{B} \\
\sum_k L_k h^k_i \leq p_i S_i
\end{gathered}$$

V. RESULTS

We evaluated the numerical result of our model using multiple network topologies. Fig. 3 is one of the Rocketfuel networks [15]. Fig. 4 is a Dorogovtsev-Goltsev-Mendes (DGM) topology and Fig. 5 is a tree network. The Rocketfuel topology is simplified by removing the leaf nodes from the original network and consolidating the demands from the removed nodes into their parent nodes [16]. The simplified network has 50 nodes and 194 directed links. These three topologies are comparable in size. The number of users is 25 nodes in Rocketfuel and 27 nodes in DGM and tree topologies. We consider one peering point for each network, and the rest of nodes are cache candidates nodes. At each node, the demand for each content follows a Zipf distribution with $\alpha = 2$. We assumed all the users have the same demand, and it is increasing by 5% every month every where. This increase is based on the current observation of OTT demand increase. As mentioned earlier, for each budget point we solve a series of feasibility problem and increase the demand until the network is saturated. We also simulated the back-pressure algorithm in [14] to compare with the performance of our model.

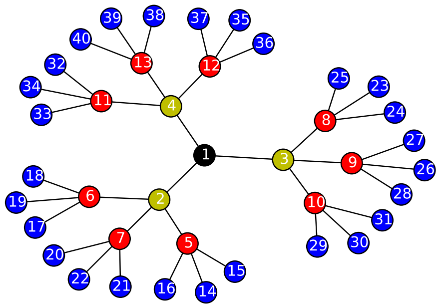

Fig. 5: Tree network

A. *Time-to-Exhaustion of different topologies*

To evaluate the performance of the CDN method, we find the TTE of the network by putting at most four caches. The assigned storage budget is equally divided between these nodes, assuming they are all using similar hardware. In other words, we use homogeneous caching. For example, in Rocketfuel network (Fig. 3), Nodes 5, 10, 12 and 14, are selected for caching, and respectively, in DGM topology, Nodes 2, 3, 4, 5 and tree topology, Nodes 2, 3 and 4 are selected for caching. It is also worth noting that in the tree topology only three nodes are selected for caching, since adding more caches has not increased the TTE further. To evaluate the performance of NDN, we will enable caching in all of the candidate nodes. Therefore, storage budget will be equally divided between more nodes, and each node can cache less number of objects.

Fig. 6, Fig. 7 and Fig. 8 show the TTE in different topologies while using CDN model, NDN model and NDN simulation using back pressure algorithm. We assumed that there is demand for 2000 objects, divided into 100 popularity groups, each with the size of 1Mb and all of the links in the network have the capacity of 1Gb/s. As mentioned above, we had placed at most four caches in CDN while all of the nodes can cache in NDN scenarios. Note that in all of the topologies,

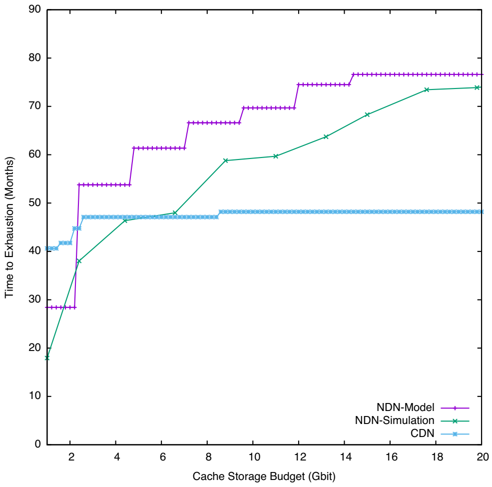

Fig. 6: Time-to-exhaustion in Rocketfuel network

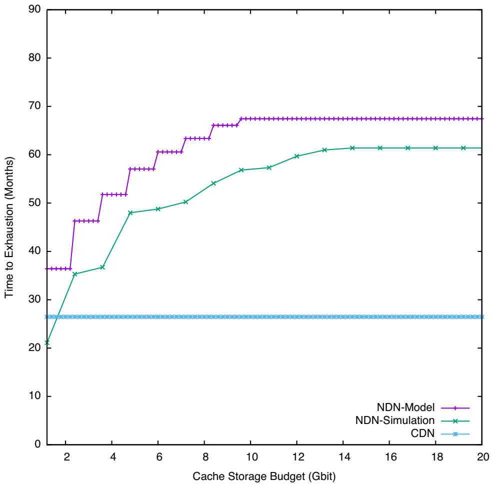

Fig. 8: Time-to-exhaustion in Tree network

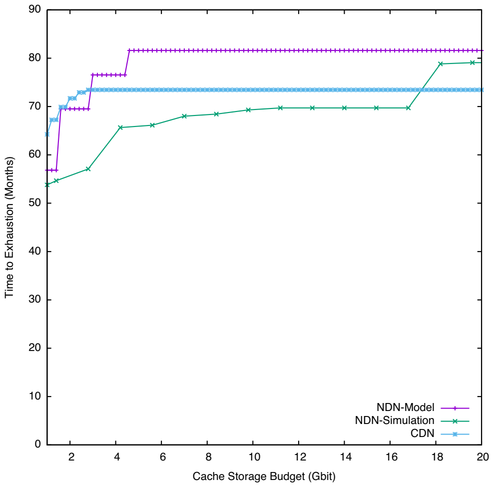

Fig. 7: Time-to-exhaustion in DGM network

NDN-Simulation using back-pressure closly follows the our NDN-model.

At very low storage budget, CDN and NDN had a similar TTE, because most of the content is provieded by the peering point, and that will become the bottleneck of the network. This means network onset of congestion will be similar for both NDN and CDN scenarios. Different topologies have different TTE for very low storage budget. The TTE depends on the onset of congestion, and the onset of congestion depends on the topology of the network. TTE is lowest for the tree topology and the highest for the DGM topology. This observation is also in agreement with the reciprocal of *network criticality* of each topology [17].

By increasing the caching storage, TTE is also increased. As mentioned earlier, storage budget is equally divided between all caches. Therefore, the increase in the total storage budget will increase the TTE. As the number of caches increases, each cache will receive a smaller portion of the budget. Therefore, when there is not enough additional storage available to each cache, there will be no change in the number of cached contents, and the TTE will not change either. This minimum increase in storage depends on the number of caches in the network. In NDN, the steps are larger since there are more caches and a greater increase in the total storage budget is required to cache more contents. In CDN, the steps are smaller since there are only four caches and a smaller amount of increase in storage budget, compared to NDN, will result in more cached contents. However, the height of the steps decreases with increase of the budget, because caching begins losing its effect. There is also a limit on the maximum TTE of each topology, after which even caching does not help anymore. This TTE is the maximum that a network can reach with the help of caching. Similar to the low budget TTE, the maximum TTE also depends on the topology of network.

Furthermore, in low storage budget, there is little difference in TTE between using CDN and NDN. Because of homogeneous caching, sometimes CDN even performs better. However, in all of the topologies, the network that uses CDN is saturated in much lower storage budget compared to NDN. This better performance is the direct result of the NDN paradigm. In NDN, due to its in-network caching and point-to-point nature, each cached content is sent over the links only once. However, in CDN each content is sent multiple times. This waste of link capacity shows itself by having the network saturated much sooner. There is a huge difference in maximum TTE between using CDN or NDN in each topology. In Rocketfuel, using CDN will saturate the network after 47 months. But using NDN, the network can be operational until 77 months. Similarly, DGM with CDN is operational for 74 months and with NDN for 82 months. Tree topology with CDN is operational for 27 months and with NDN for 67

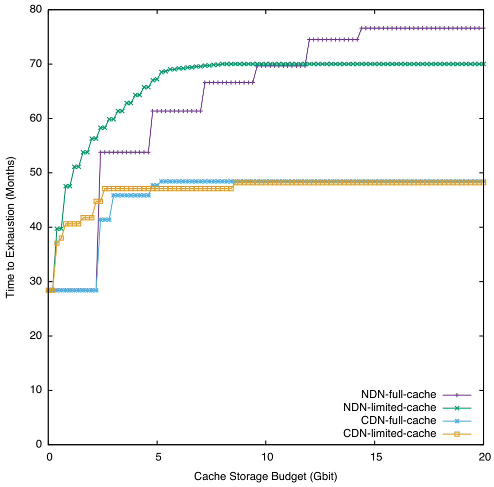

Fig. 9: Changes in TTE of Rocketfuel topology with number of caches

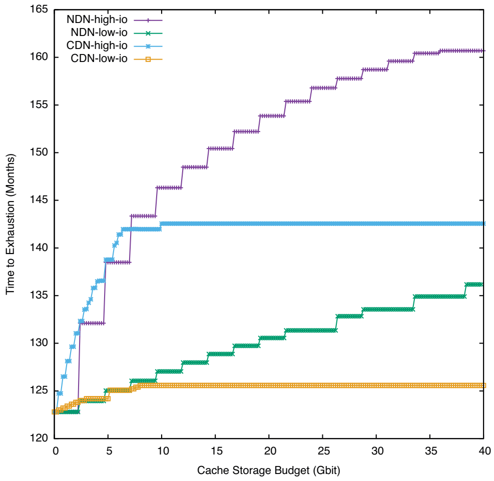

Fig. 10: Changes in TTE of Rocketfuel topology with I/O limit

months. This huge difference in tree topology is because of the fact that in NDN caches are placed throughout the network. As in Fig. 5, NDN places caches in Nodes 2 to 13. But CDN only places caches in Nodes 2, 3 and 4. For example, having a cache in Node 5 will reserve bandwidth in all of the up-links and will make more capacity available to deliver more content.

B. Limited NDN Deployment

To see how much of the difference in TTE between CDN and NDN comes from the number of caches in the network, we can limit the number of caches in NDN to four as well. Fig. 9 shows that even with four caches, content delivery using NDN outperforms the CDN design. We have also considered a non-practical case that every node in the CDN can also cache

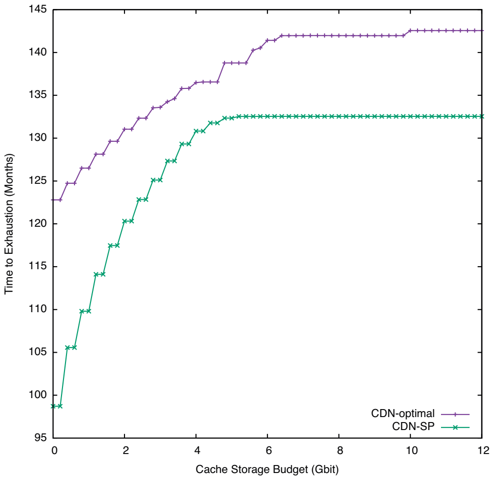

Fig. 11: Changes in TTE of Rocketfuel topology with Routing algorithm

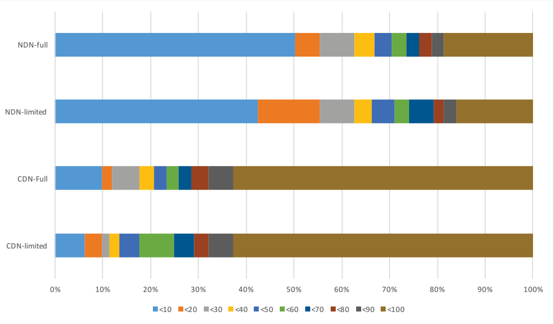

Fig. 12: Link utilization of NDN vs CDN

contents. This case is just for the comparison and in practice cannot be implemented due to the nature of CDN. One could say that one of the reasons behind the NDN proposal is the impossibility of in-network caching in TCP/IP. However, even if all of the nodes in the CDN had the caching capability, the network will saturate similar to the case that there are four caches in the network. In addition, limited NDN deployment has better TTE for low budget than full NDN deployment. This suggests limiting the number of NDN caches when the storage budget is low.

We can also look at link utilization in the network. Fig. 12 shows the percentage of links with various percentage of utilization during network congestion. Using CDN, more than 60% of the links will have a link utilization of more than 90%. In contrast, NDN scenario, even with limited deployment, has less than 20% of the links with high utilization. Using NDN, has resulted in a network that more than 40% of the links have link utilization of less than 10%. This difference in link utilization means that if CDN is used to increase the TTE of the network, we have to increase the capacity of most of the

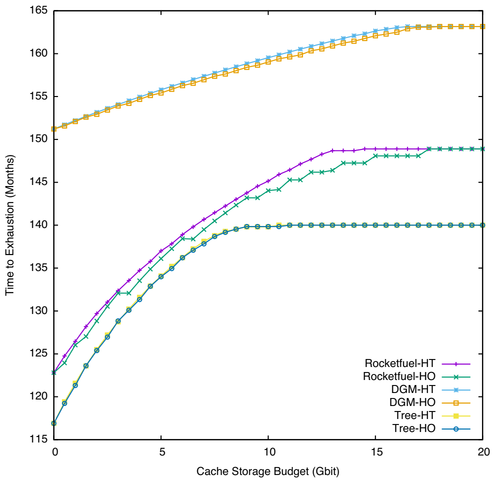

Fig. 13: Heterogeneous vs Homogeneous caching storage in NDN

links. But using NDN will result in a much fewer bottlenecks, which makes capacity planning much easier and cheaper.

C. I/O Speed Effect

One of the parameters we have considered in our modeling is the I/O limit of each cache. The I/O limit depends on hardware design of the cache. Fig. 10 shows the effect of this parameter. To better see the difference the I/O speed makes, we have increased the capacity of all of the links to limit the effect of congestion. As shown in Fig. 10, as the I/O limit increases from 10Gb/s to 100Gb/s, TTE also increases. But it must be said that having low link capacity will greatly diminish the improvement gained by having a hardware with higher I/O limit.

D. Routing Protocol Effect in CDN

As mention above, our modeling tries to maximize the TTE and therefore optimizes the routing of data. However, in practice routing is not optimal. As shown in Fig. 11, by enforcing shortest-path routing for CDN in the Rocketfuel network the TTE will be reduced by more than 10 month. The NDN does not have this problem since its strategy layer can employ an optimal routing algorithm.

E. Heterogeneous Caching

Fig. 13 shows the effect of heterogeneous caching on the TTE when NDN is used. In using heterogeneous caching, the model will assign each cache a different storage capacity while satisfying total storage budget constraint. It is expected that the symmetry in tree and DGM topologies would imply little benefit to the heterogeneity. However, there is some difference in Rocketfuel which is less symmetric. If heterogeneous caching is employed, TTE in the Rocketfuel network is increase at most by 3 months.

VI. CONCLUSION

In this paper, we propose in-network caching strategy for service providers in order to control the time-to-exhaustion for their backbone capacities. We propose that the service provider uses Named-Data Networking (NDN) for video delivery in its network. Even a limited deployment of NDN provides a substantial increase in time-to-exhaustion of the network and lower number of links with high utilization. We studied different parameters that affect the performance of content delivery in the service provider's network and validated our model by simulation. We also demonstrated that heterogeneous caching does not provide a substantial performance benefit over homogeneous caching.

REFERENCES

[1] "Global Internet Phenomena," Sandvine, Tech. Rep., 2013. [Online]. Available: https://www.sandvine.com/trends/global-internet-phenomena/

[2] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello, "From Content Delivery Today to Information Centric Networking," *Comput. Netw.*, vol. 57, no. 16, pp. 3116–3127, Nov. 2013.

[3] 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*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[4] J. Pang, A. Akella, A. Shaikh, B. Krishnamurthy, and S. Seshan, "On the responsiveness of DNS-based network control," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004, pp. 21–26.

[5] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 568–582, Oct. 2000.

[6] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and beyond) Network Architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, Aug. 2007.

[7] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[8] G. Tyson, S. Kaune, S. Miles, Y. El-khatib, A. Mauthe, and A. Taweel, "A trace-driven analysis of caching in content-centric networks," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, July 2012, pp. 1–7.

[9] P. Agyapong and M. Sirbu, "Economic incentives in information-centric networking: implications for protocol design and public policy," *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 18–26, December 2012.

[10] D. Rossi and G. Rossini, "On Sizing CCN Content Stores by Exploiting Topological Information," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, March 2012, pp. 280–285.

[11] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Teletraffic Congress (ITC), 2011 23rd International*, Sept 2011, pp. 111–118.

[12] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal cache allocation for content-centric networking," in *Network Protocols (ICNP), 2013 21st IEEE International Conference on*, Oct 2013, pp. 1–10.

[13] A. Tizghadam and A. Leon-Garcia, "Betweenness centrality and resistance distance in communication networks," *Network, IEEE*, vol. 24, no. 6, pp. 10–16, November 2010.

[14] E. M. Yeh, T. Ho, M. Burd, Y. Cui, and D. Leong, "Vip: A framework for joint dynamic forwarding and caching in named data networks," *CoRR*, vol. abs/1310.5569, 2013.

[15] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, ser. IMW '02. New York, NY, USA: ACM, 2002, pp. 231–236.

[16] D. Applegate and E. Cohen, “Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs,” in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM ’03. New York, NY, USA: ACM, 2003, pp. 313–324.

[17] A. Tizghadam and A. Leon-Garcia, “Robust network planning in nonuniform traffic scenarios,” *Computer Communications*, vol. 34, no. 12, pp. 1436 – 1449, 2011.