A Trace-Driven Analysis of Caching in Content-Centric Networks

Gareth Tyson*, Sebastian Kaune†, Simon Miles*, Yehia El-khatib‡, Andreas Mauthe‡ and Adel Taweel*

*Department of Informatics, King's College London, UK
†Multimedia Communications Lab, Technical University of Darmstadt, Germany
‡School of Computing and Communications, Lancaster University, UK

Abstract—A content-centric network is one which supports host-to-content routing, rather than the host-to-host routing of the existing Internet. This paper investigates the potential of caching data at the router-level in content-centric networks. To achieve this, two measurement sets are combined to gain an understanding of the potential caching benefits of deploying content-centric protocols over the current Internet topology. The first set of measurements is a study of the BitTorrent network, which provides detailed traces of content request patterns. This is then combined with CAIDA's ITDK Internet traces to replay the content requests over a real-world topology. Using this data, simulations are performed to measure how effective content-centric networking would have been if it were available to these consumers/providers. We find that larger cache sizes (10,000 packets) can create significant reductions in packet path lengths. On average, 2.02 hops are saved through caching (a 20% reduction), whilst also allowing 11% of data requests to be maintained within the requester's AS. Importantly, we also show that these benefits extend significantly beyond that of edge caching by allowing transit ASes to also reduce traffic.

I. Introduction

Traditionally, the Internet has been a host-centric environment in which hosts generate packets that are subsequently routed to other hosts. Throughout the majority of its evolution, this has been viewed as its primary function due to the requirements of prominent higher-level applications such as telnet, e-mail and FTP. However, as bandwidth availability has improved and Internet access has become ubiquitous, content delivery has overtaken such applications to become the predominant traffic generator within the Internet [1].

In response to this, researchers have proposed re-architecting the Internet to be *content-centric* [2]. This involves replacing or augmenting the existing host-to-host routing infrastructure with a host-to-content routing substrate. Thus, applications generate content 'Interest' packets, which are routed to content sources that reply with 'Data' packets using globally unique content identifiers. This exploits the observation that applications rarely have a vested interest in where the content comes from as long as it is verifiable and conducive with their underlying delivery requirements. Potential benefits include superior resource utilisation through in-network caching, better mobility support and more reliable data security.

As of yet, many designs have been proposed, however, little in-depth evaluative work has been performed validating the *real-world* applicability of the above benefits. This paper therefore seeks to perform an evaluative study of one of the above benefits: *content-centric caching*. By uniquely identifying each content chunk at the network-layer, it becomes possible to cache content within routers, thus scalably enabling the potential of redundant traffic removal. To study this topic, we have utilised two measurement sets to explore how real-world request patterns would perform in a content-centric Internet.

To acquire these request patterns, we chose to measure one of the most significant content traffic generators [1] in the Internet: BitTorrent. This is because BitTorrent's open nature allows us to follow the request patterns of a large number of hosts (≈35k) across the entire Internet. This is in contrast to prior studies that either use synthetic workloads (e.g. [3][4]) or solely use HTTP traces taken from individual observation points (e.g. [5]). Unfortunately, the former does not offer real-world insights, whilst the latter can generally result in biased traces (e.g. a university campus will generally have more skewed content request patterns than typical).

Using these real request patterns, we constructed NS2 simulations using router and Autonomous System (AS) level Internet topology traces, provided by CAIDA [6]. This allowed us to replay the monitored content requests over a real-world topology, whilst introducing content-centric caching in the routers. We find that content-centric caching can, indeed, significantly reduce path lengths for retrieving content. This is manifested in two ways. At the router-level, we show that three or more hops can be saved for 30% of packets, using feasible caches sizes, with 2.02 hops being saved on average. On an intra-domain level, we also find that a significant proportion of Interest packets can be maintained within a requester's AS (11%) through caching. More importantly, however, in-line with some of the key research questions[1] raised in [7], we show that the benefits of content-centric caching extend well beyond that of traditional edge caching. Specifically, we find that a further 21% of Interest packets can be served within only a single AS hop by enabling caching. Through this, we also explore some of the potential financial incentives for content-centric deployment, highlighting that many benefits may be available for transit networks as well as stub ISPs.

The rest of the paper is structured as follows. First, the background to the topic of content-centric networking is presented alongside related work. Next, the methodology of

---

[1]The authors questioned whether ubiquitous router caching would create much improvement over edge caching.

our evaluation is discussed before presenting the results of our simulations. Finally, the paper presents conclusions and future work.

II. Background and Related Work

A. Content-Centric Networking

Within a content-centric network, nodes can both publish and consume a content packet to/from the network using a globally unique content identifier. Publication results in routing information being propagated within the network to allow subsequent requests to be routed towards a source. Importantly, the same content can be published at multiple locations (e.g. content mirrors, caches etc.) whilst leaving the necessary source selection to the underlying routing substrate.

Prominent examples of these systems are DONA [8], LIPSIN [9] and CCNx [2]. CCNx is probably the most well known; it performs content-centric routing using similar algorithms to current IP infrastructure (e.g. OSPF, BGP), utilising longest prefix matching with hierarchical aggregation to ensure scalability. In CCNx, a content request is issued by sending an Interest packet, which is routed through the network to the 'closest' instance of the content. Subsequently, if available, the source responds with a Data packet, which then follows the reverse path back to the requester (using breadcrumbs). Alternatively, approaches such as [10], continue to use location-oriented routing but with layer 4 packet labelling for identifiers. Importantly, as content is immutable, a given Data packet will always be able to serve an Interest packet with a matching identifier. Due to its prominence, we will use CCNx terminology in the rest of the paper.

B. Packet Caching

Recent measurements have observed the redundant nature of many data transmissions over the Internet [11]. This is best highlighted by studies showing that caching efficiencies of up to 90% can be achieved for certain traffic types, e.g. distributing software updates [12]. Consider, for instance, the transmission of the Google logo, which will happen millions of time per-day. Currently, routers simply see this as a packet, which must be forwarded to a host, regardless of the packet's payload. Due to this, identical payloads will frequently pass through the same router, consuming unnecessary resources. Systems such as CacheCast [13] exploit this observation to enable scalable multicast by stripping the payload from redundant packets and reconstructing them downstream. Alternatively, other systems [14] try to optimise bandwidth usage by stripping payloads from any packets that are cached downstream. These approaches can be contrasted with traditional edge caching as recently discussed in [12] and forward caching, which involves placing caches strategically throughout the entire network [15]. Similarly, peer-to-peer alternatives such as Corelli [16] are also available.

The above approaches, however, are built using the existing network infrastructure with its host-to-host routing substrate. Consequently, redundant packet/content identification must be performed using techniques such as layer 3/4 fingerprinting or layer 7 content identifiers. In contrast, content-centric networks make each packet identifiable as a unique item of content. Through this, in-router caching becomes a trivial task that can easily be performed because a cached packet will be immutable, i.e. any other packets containing the same identifier will be identical (unlike in [14]). Therefore, whenever a router receives an Interest packet, before forwarding, it can check its cache to see if a corresponding Data packet is locally stored. This can not only reduce the load on end host providers but also can minimise resource utilisation and remove the need to use costly inter-AS links. As of yet, however, we have a poor understanding of how effective content-centric caching would be in the wild; in fact, some have suggested it might not even be worthwhile [7]. Up until now, most studies (e.g. [3][4]) have used synthetic workloads to study performance. Although, this allows a number of insights to be gained, it does not allow a real-world study to be performed. Similarly, early trace-based studies [5] utilise HTTP traces taken from individual vantage points, preventing an Internet-wide macroscopic understanding being gained. The purpose of this paper is therefore to build upon this previous work to measure how caching would perform in a more realistic setup (in terms of both topology and request patterns). The work is therefore presented as a form of 'case-study' in this domain.

C. Traffic Localisation

An alternative to the above forms of caching is the concept of traffic localisation, in which consumers attempt to utilise content sources within their own domain (e.g. AS). In-line with the traces used in this paper, BitTorrent has been under particular focus due to its heavy traffic profile. Approaches such as [17][18] have proposed effective mechanisms to localise a large proportion of content requests with only a limited impact on system performance. These approaches, however, clearly differ greatly from the packet-level approach of content-centric networking, and take on many protocol-specific characteristics of BitTorrent. They also do not support the extension of localisation beyond the requester's AS. Similarly, traffic localisation does not utilise router-level caching; instead, relying on content replicas being available on peers. This work does, however, highlight the strong (topological) overlap of content requests in the Internet today, offering strong motivation for pursuing a content-centric approach.

III. Methodology

To evaluate the potential of content-centric router caching, it is necessary to utilise two data sets: (i) realistic router-level and AS-level traces, and (ii) realistic content request patterns. Assuming that a content-centric topology would be similar to the existing Internet topology, these data sets would allow us to measure the real performance benefits of using content-centric networking. To acquire the former, we have utilised CAIDA Internet traces [6] to generate the real-world router-level and AS-level topology. To model content request patterns, we have then used a set of BitTorrent traces that we have gathered to replay requests consisting of 35,240 nodes. Using these, we have performed NS2 simulations to calculate the potential of content-centric caching in this 'case-study' environment.

A. Topology Traces

To perform a detailed analysis, we use both router-level and AS-level Internet traces. We have chosen this because the router-level topology allows us to study resource consumption on an intra-domain level by indicating how many resources (routers and links) are used by content-centric and location-oriented networking. The AS-level traces then allow us to study routing on an inter-domain level, which, consequently, offers many insights into the financial incentives that might be available for transit and stub networks when deploying content-centric routing (e.g. relating to peering agreements).

In the simulations, CAIDA's ITDK [6] traces are used, which provide a router-level and AS-level topology of the Internet. The ITDK traces were performed by executing traceroutes from 54 monitor nodes (in 29 countries) to randomly chosen destinations in each /24 BGP prefix (between 11/07/10 - 26/07/10). Bandwidth capacities were then allocated to end nodes by performing longest-match lookups on the iPlane service [19]. These traces therefore allow us to compute the path length and arrival times of any Data packets on a router-level and AS-level granularity.

B. Content Request Traces

To be able to accurately model request patterns, we chose to perform BitTorrent measurements. Importantly, this allowed us to gain large-scale macroscopic traces across many different ASes and not just single vantage point. To achieve this, we implemented a BitTorrent crawler that connected to Mininova.org for 38 days. This crawler periodically requested, from multiple sites in Europe, tracker information regarding each torrent's number of seeders and leechers alongside the members' IP addresses. This study allowed us to gain a large number of measurements regarding details such as peer arrival patterns, seeder:leecher ratios and torrent sizes. Importantly, it also allowed us to trace all requests for a given content item, alongside the IP address of requesters. Each monitored torrent was contacted every 10 minutes to refresh this information (more frequent requests resulting in our IP addresses being blocked); further information can be found in [20]. From this, 35,240 nodes were taken, by randomly selecting 71 items of content. These files ranged from 1 MB to 1.7 GB with an average swarm population of 391 nodes.

It is important to also note that we only utilise the request patterns from these traces; we do not take on other aspects of BitTorrent, e.g. the protocol, tit-for-tat, source selection, naming etc. The only information taken from the traces is therefore content size, request arrival rates and topological location of requests. By removing the more BitTorrent-specific aspects, it thus becomes possible to generalise these request patterns for any application downloading stored content.

C. Simulations and Analysis

The above data sets allow us to replay a number of real-world content requests over a real-world Internet topology. Importantly, this is on a macroscopic level, unlike studies that have solely looked at the traffic observed at a single point in the network (e.g. the edge of a university network). To achieve this, both sets of traces were parsed to generate a set of NS2 simulation scripts. Packet-level traces were then post-processed to calculate the effects of introducing router-level caching.

1) NS2 Simulations: The simulated torrents consisted of a variety of node populations and were distributing content of a range of sizes. Each peer in the torrent was attached to the ITDK trace topology by performing a longest prefix lookup on all IP addresses within the topology. If multiple potential candidates were found, a random one was selected. To make the simulations tractable, the ITDK traces were then reduced by computing the shortest path between all nodes in each torrent. Consequently, routing was solely based on the shortest path as the traces did not contain any record of any actual (possibly fluctuating) routing decisions made in the Internet. One limitation of this is that BGP information cannot be taken into account (which is further based on a number of characteristics, e.g. monetary costs, Quality of Service (QoS), load balancing). We do not believe this, however, to be a significant problem as routing in a content-centric network is likely to differ significantly from existing location-oriented BGP [21]. Therefore, shortest path routing is an appropriate assumption in this setting.

Once the shortest paths had been computed, each peer in the torrent was allocated as either a provider or consumer, based on the seeder:leecher ratio recorded. Subsequently, as consumers arrived in the torrent, a random provider was selected to connect to. This is representative of a content-centric network implementation as described in [10]. Data transfers were performed from *consumers* to *providers* using a TCP stream. This then allowed us to trace the paths taken by Interest packets in the network (Data packets always take the reverse path [2]).

2) Post-Processing: The above simulations produced packet-level traces for all content exchanges. No content-centric caching was built into these simulations and, consequently, the traces represent the traditional location-based networking paradigm. To augment this data, the packet-level traces were therefore post-processed to calculate the potential of introducing global content-centric caching. This was done by parsing the packet-level traces to elicit the arrival times of each packet at each router. As each torrent (and therefore each content item) was simulated independently, the TCP sequence number was used as the content identifier. Following this, we calculated a number of statistics based on introducing caching at each router; the replacement policy used was FIFO. We selected a number of per-router cache sizes: 10, 100, 1000 and 10,000 packets (default packet size is 1000 bytes). We also performed simulations with a 100,000 packet cache size; this, however, yielded near identical results to 10,000. This is because the benefits of increasing cache sizes exponentially decrease due to the skewed nature of content demand. Importantly, these cache sizes represent the proportion of a router's cache allocated to storing a given item of content. Such a scenario would be likely if QoS contracts had been negotiated between content providers and network infrastructure.

IV. RESULTS

This section presents the results of the simulation studies. First, we investigate the impact that caching has on Internet path distances before inspecting how content and popularity characteristics can affect this. As our core metric, we use the number of network hops. This is because it offers a range of insights that are extensible for modelling such things as delay and performance. Importantly, it also allows the number of inter-domain interactions to be monitored (this is particularly important when considering financial peering agreements).

A. Reductions in Path Distance

The aim of this section is to quantify the change, if any, in path length that content packets have to traverse using content-centric caching.

1) Router-Level Savings: An important benefit of using content-centric networking is the ability to reduce the number of router-level hops taken by packets through the network. This means that packets have less distance to travel, reducing network loading as well as optimising protocols that are heavily dependent on delay. Fig. 1 presents a cumulative distribution function (CDF) of the number of hops a packet takes through the network using various cache sizes. The benefits when using small cache sizes (e.g. 100, 1000 packets) are insignificant. When using a cache of 1000 packets, 90% of packets have no reduction in the number of hops. In contrast, when increasing the cache size to 10,000 packets, large reductions can be gained (up to 10 hops for 2% of packets, with at least 3 hops being saved for 30% of packets). Despite this, 61% of packets have no reduction whatsoever, even when using this large cache size, as shown in Fig. 2. As shown in Table I, this leads to an average reduction of 2.02 hops per Interest packet (with a high standard deviation of 3.2). This high deviation indicates that relatively high caching hit rates will be achieved for some data, whilst other data will benefit little, if at all. We also find that, probabilistically, most savings will occur within a few hops of the provider. Despite these low numbers, significant quantifiable benefits can be observed: when calculating the aggregated amount of per-hop data that traverses the network, we find that 294 GB of data is removed (from an original total of 1462 GB) when using caches of 10,000 packets, representing a significant reduction.

Cache Size	0	10	100	1k	10k
Router-level	10.03	9.82	9.72	9.48	8.01
AS-level	3.38	3.37	3.36	3.31	2.98

TABLE I: Mean Number of Hops

2) AS-Level Savings: The above analysis has looked at the savings on a per-router granularity. This, however, focusses on resource utilisation; of more importance to most ASes is the amount of inter-AS traffic that can be reduced. This is due to the high cost of transporting data across domain boundaries; for example, it is predicted that prices will reach ≈$2.34 per Mbps in 2012 [22]. Transit reduction, from the perspective of stub networks, is therefore highly beneficial. In contrast,
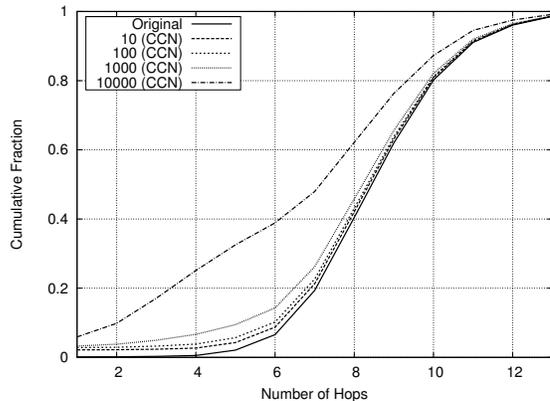

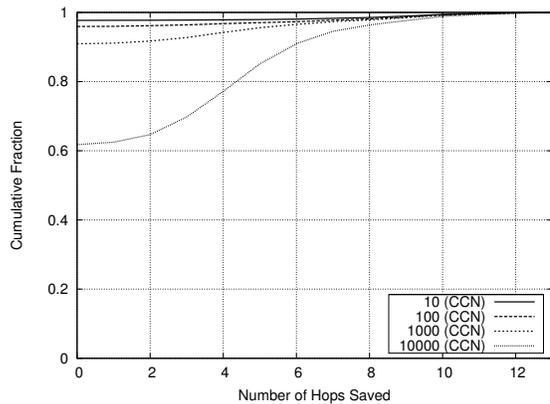Fig. 1: CDF of Router-Level Hops in Original and CCN Traces

Fig. 2: CDF of Router-Level Hops Savings using CCN

transit networks, which make their money from transporting data, have a direct incentive to oppose using content-centric networking. It is therefore important to understand the possible reductions in inter-AS traffic. Fig. 3 presents a CDF of the number of ASes traversed for each packet with various cache sizes. For small cache sizes (e.g. 100 and 1000 packets), there is almost no performance advantage: caching 1000 packets per-router maintains only 3% of Interest packets within the requester's AS. Consequently, the use of smaller sized router buffers [23] would have little effect on reducing inter-AS traffic. Even when using caches of 10,000 packets, 74% of all packets achieve no reduction in AS traversals, as shown in Fig. 4. Despite this, in the traces, up to 6 AS-level hops can be reduced, with 13% of packets achieving at least a reduction of 2 AS traversals. Consequently, we discern significant deployment incentives from the results. This is highlighted in Fig. 5, which presents the aggregated inter-AS per-hop data transferred over the network: it can be seen that 19% of data can actually be removed from the network through caching, creating a notable reduction in traffic.

An important question is therefore who would benefit from content-centric deployment? Intuitively, networks without settlement-free agreements would benefit most due to the expensive cost of transiting data through other ASes. However, despite the large overall reduction in data transferred, it can be seen in Fig. 4 that relatively few packets are actually maintained within the requester's AS (11% for 10,000 packet
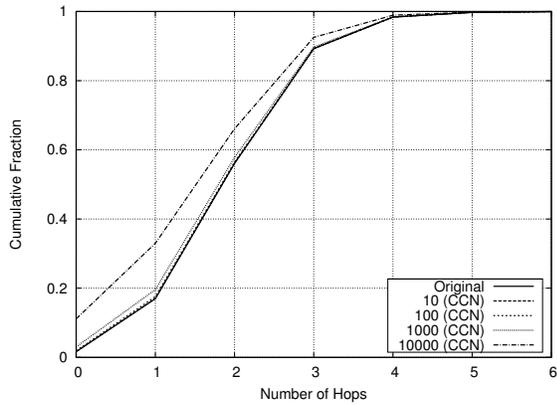
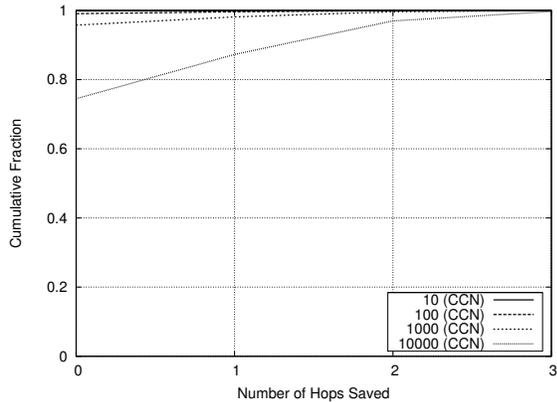Fig. 3: CDF of AS-Level Hops in Original and CCN Traces

Fig. 6: CDF of Router-Level Hops for Small Content Items

Fig. 4: CDF of AS-Level Hops Saved in CCN Traces

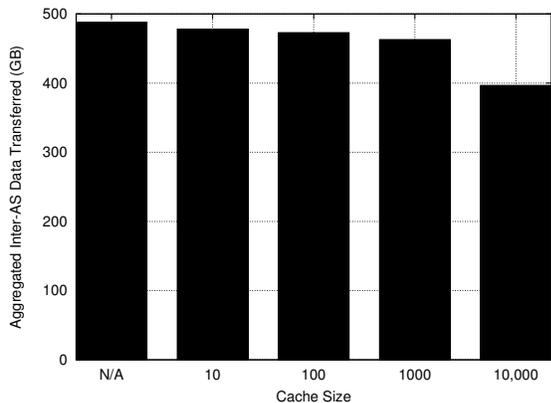Fig. 7: CDF of Router-Level Hops for Large Content Items

Fig. 5: Aggregated Inter-AS Data Transferred

cache). Consequently, benefits are likely to be greatest for Tier-2 ASes that can utilise cached data from many dependent ASes, rather than stub ASes containing smaller sets of consumers. Similarly, in-line with the previous results, we find that the greatest hit rates occur closer to the provider than the consumer due to the increased probability of route overlap.

B. Content Characteristics

The above sections have studied the amount of redundant traffic that can be removed from a content-centric network using caching. However, it is also important to study how
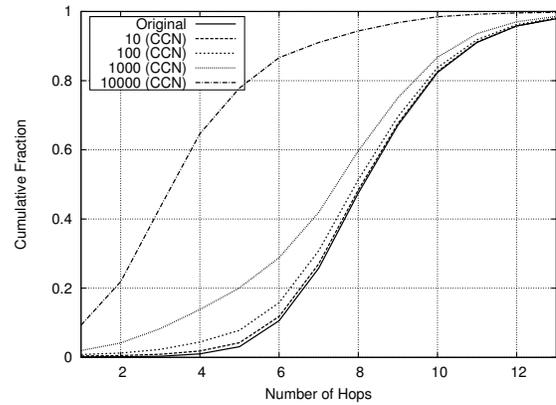
different properties of the content and consumers can impact this. This section looks at how items of different sizes and popularities will affect performance.

1) Content Size: A variety of content items are transported over the Internet on a daily basis; these range from high definition films to small music files. It is therefore important to understand how content size will impact the performance of content-centric networking. To study this, we split the results into two groups, (i) small items (less than or equal to 5 MB), and (ii) large items (greater than 5 MB). This is to differentiate web sites and small media files from larger files (e.g. films, software). Fig. 6 and Fig. 7 present CDFs of the number of router-level hops for both small and large items.

It is evident that small content items achieve a far greater reduction in hops than larger items. This is due to the finite nature of the cache; for instance a 1 MB file could be entirely cached in a 1 MB cache, whilst a 10 MB file clearly could not. In contrast, large files gain marginally lower cache performance than the overall averages presented in Fig. 1. For instance, with a 10,000 packet cache, the overall average of hops taken is 8.01, compared to 5.12 for small files and 8.41 for large files. It is important to note that this will therefore vary heavily based on the link's traffic profile and caching policy; clearly, if a high capacity link frequently flushes content out of the cache, performance will degrade.

2) Content Popularity: A content item's popularity can be defined by the number of consumers who request it in a given
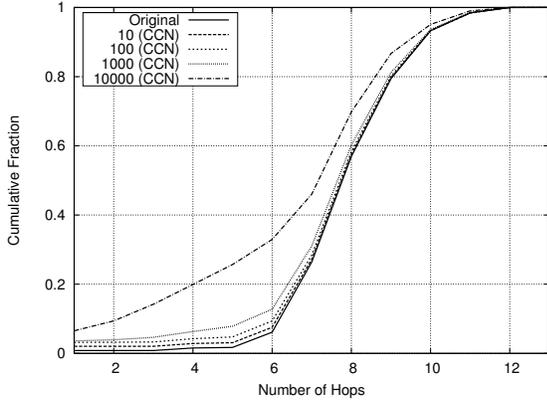
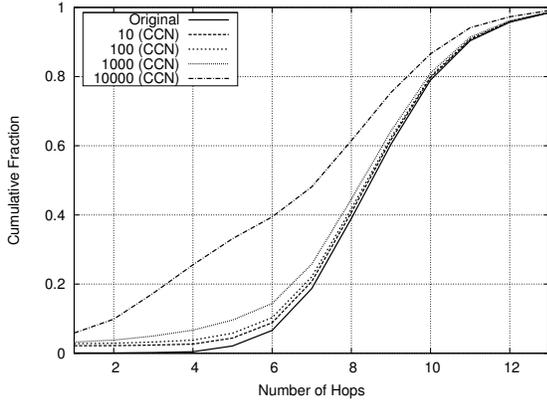Fig. 8: CDF of Router-Level Hops for Unpopular Content

Fig. 9: CDF of Router-Level Hops for Popular Content

period. Clearly, content-centric networks are more suited to highly popular content with short peak periods as this creates a greater probability of route overlap. To study this, we split the results into two groups, (i) unpopular content (less than 50 nodes), and popular content (between 51 and 2952 nodes). Fig. 8 and Fig. 9 present CDFs of the number of router-level hops for these content types.

Broadly speaking, varying content popularity does not have a significant effect on the caching performance. This is because of the nature of BitTorrent-like traffic, which is usually based on relatively small consumer groups. We categorise unpopular content as having fewer than 50 consumers. Consequently, the difference between that and the most popular content (2952 nodes) is not significant considering the scale of the topology traces ($\approx$3 million routers). Despite this, popular content gains a marginally better performance (i.e. fewer hops) than unpopular content due to the increased probability of route overlap. It is therefore evident that very high caching performance will be limited to either extremely popular content (e.g. the Google logo) or content that is popular in a topological locality. Consequently, performance will vary greatly between different content request patterns (occurring in different domains, e.g. video streaming, software update deployment etc.).

C. Content-Centric vs Edge Caching

In Ghodsi et. al. [7], the authors suggest that the benefits of content-centric caching might not exceed far beyond that of

traditional edge caching. Consequently, it is vitally important to answer this question at this early stage in the research field. A typical form of edge caching involves utilising a proxy that also caches data. Through the simulations, we can therefore place an edge cache at every Tier-3 AS and compare the performance against that of global content-centric caching.

Our simulations show that using a simple edge cache of 10,000 packets would allow 11% of Interest packets to be served from within the requester's AS. This results in a 10% reduction in egree link utilisation of edge ASes. Assuming that an edge cache could be placed in each stub AS, this means that such benefits could immediately be gained without using content-centric networking. At first, it might seem that edge caching could therefore negate the need for content-centric caching. However, it can also be observed that global content-centric caching can offer benefits far beyond 11% by allowing caching across multiple ASes; for instance, $\approx$16% more requests can be limited to 1 AS traversal, whilst $\approx$10% more can be limited to 2 AS traversals through CCN (i.e. through caching in transit ASes). Specifically, overall, with a cache of 10,000 packets, 32% of requests can be limited to a maximum of 1 AS traversal, whilst 34% more can be limited to 2 AS traversals. Interestingly, this means that content-centric caching benefits are not restricted to stub ASes wishing to maintain traffic within their own domain. In fact, direct benefits can be gained by these intermediate Tier-2 transit ASes by reducing their need to utilise (non settlement-free) egress links whilst still serving the needs of their dependent stub networks. This suggests that many Tier-2 networks may want to locally implement content-centric caching but deter any dependent Tier-3 networks from doing so, to maintain their income stream. Importantly, this also shows that global content-centric caching offers benefits well beyond simple edge caching in stub networks.

D. Discussion and Limitations

It has been shown that introducing larger caches can, indeed, offer significant benefits when faced with 'BitTorrent-like' request patterns. These benefits, however, require caches in the order of at least 10,000 packets per content stream; although, this is a feasible value, it will grow dramatically as the number of data streams scale up. Importantly, these benefits have also been shown to extend well beyond that of traditional edge caching by also reducing Tier-2 AS traffic. A number of key points can be derived; specifically, these are:

- Caching traffic in a content-centric network can lead to significant router-level path length reductions. Up to 10 hops can be reduced for 2% of packet, with 2.02 hops being saved on average. Probabilistically, however, these hops are likely to be saved nearer to the provider, rather than the consumer.
- Caching traffic can lead to a noticeable reduction in inter-domain traffic. Up to 11% of Interest packets can be maintained within requesters' ASes.
- Notable benefits can be gained by placing caching solely at the edge of ASes. However, content-centric networking offers improved performance well beyond this by allow-

ing caching to take place globally (e.g. 32% of Interest packets can be maintained within one AS traversal).

- Caching traffic for small items of content will lead to far higher performance levels than large content on links that do not have their caches frequently flushed by rival data, e.g. in the traces, requests had an average of 5.12 hops for small files compared to 8.01 overall.

It is now, however, vital to detail some key limitations of the study. One limitation is the lack of realistic routing information for content-centric networking; this is because routing is usually based on a range of characteristics including load balancing, QoS issues and inter-domain business arrangements. However, as content-centric networking has not been deployed, there is no information regarding this available (as it will change significantly from BGP [21]). Consequently, routing has been based on fixed shortest paths, thereby removing the presence of unpredictable multi-path routing and route variations. Another limitation is the lack of 'rival' data streams that might fill a cache; instead, currently, a proportion of each router cache is dedicated to the content being studied. This allows a better understanding of caching without an explosion in the number of parameters (thereby improving casual traceability). However, on the other hand, it does not represent a true deployment. Despite this, we believe the results to offer a valuable insight into how a content-centric network might perform under realistic macroscopic traces.

V. Conclusion and Future Work

This paper has presented a performance study of the potential of content-centric caching. To achieve this, two data sets have been combined: a detailed Internet router-level topology trace and a BitTorrent content request pattern trace. We chose the use of BitTorrent request patterns due to its significance in current traffic studies [1], but also because all prior studies have utilised either synthetic workloads or HTTP traces, which portray different characteristics. Using these data sets, simulations have been performed to replay real content requests in a content-centric environment. This paper constitutes the first step in a long-term research scheme, looking at the effectiveness of content-centric caching. We feel it is therefore vital to extend this study to include many different set-ups, e.g. looking at content requests that have greater geographic correlation. It is also important to contrast these results against alternate forms of request patterns that exhibit different behaviour. Web delivery, for instance, often displays a more skewed popularity distribution than peer-to-peer traffic, leading to superior cache performance [24]. Clearly, an important future step is therefore applying alternate macroscopic request traces to our content-centric setup. Similarly, different content types (e.g. videos), which have different viewing characteristics, such as skipping, are important to study. Further, a wider range of metrics should be investigated (e.g. delay, bandwidth) to gain a true understanding of content-centric caching. Last, novel replacement strategies should also be explored to discover optimal mechanisms for different scenarios.

References

[1] H. Schulze and K.Mochalski, "Ipoque internet study," tech. rep., ipoque GmbH, 2008/2009.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *In Proc. 5th ACM Intl. Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2009.

[3] D. R. G. Rossini, "A dive into the caching performance of content centric networking," *Technical report, Telecom ParisTech*, 2011.

[4] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I*, 2011.

[5] Z. Ming, M. Xu, and D. Wang, "Age-based cooperative caching in information-centric networks," in *Workshop on Emerging Design Choices in Name-Oriented Networking*, 2012.

[6] "The internet topology data kit - 2011-04." http://www.caida.org/data/active/internet-topology-data-kit.

[7] A. Ghodsi, T. Koponen, B. Raghavan, S. S. abnd Ankit Singla, and J. Wilcox, "Information-centric networking: Seeing the forest for the trees," in *Proc. of 10th ACM Workshop Hot Topics in Networks*, 2011.

[8] 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, 2007.

[9] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 195–206, 2009.

[10] J. Ott, K. Budigere, P. Sarolahti, and C. Perkins, "Poor man's content centric networking (with TCP)," Tech. Rep. Aalto-ST 5/2011, Aalto University, 2011.

[11] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: findings and implications," in *Proc. of the Intl. joint conference on Measurement and modeling of computer systems (SIGMETRICS)*, 2009.

[12] B. Ager, F. Schneider, J. Kim, and A. Feldmann, "Revisiting cacheability in times of user generated content," in *IEEE INFOCOM Conference on Computer Communications Workshops*, 2010.

[13] P. Srebrny, T. Plagemann, V. Goebel, and A. Mauthe, "Cachecast: Eliminating redundant link traffic for single source multiple destination transfers," in *Proc. of the Intl. Conference on Distributed Computing Systems (ICDCS)*, 2010.

[14] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: the implications of universal redundant traffic elimination," in *Proc. of SIGCOMM Conference*, 2008.

[15] J. Erman, A. Gerber, M. T. Hajiaghayi, D. Pei, and O. Spatscheck, "Network-aware forward caching," in *Proc. of the 18th Intl. conference on World wide web (WWW)*, 2009.

[16] G. Tyson, A. Mauthe, S. Kaune, M. Mu, and T. Plagemann, "Corelli: A dynamic replication service for supporting latency-dependent content in community networks," in *Proc. 16th ACM/SPIE Multimedia Computing and Networking Conference (MMCN)*, 2009.

[17] R. Rumin, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez, "Deep diving into bittorrent locality," in *Proc. IEEE INFOCOM*, 2011.

[18] S. L. Blond, A. Legout, and W. Dabbous, "Pushing bittorrent locality to the limit," *Computer Networks*, vol. 55, no. 3, pp. 541–557, 2011.

[19] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane nano: path prediction for peer-to-peer applications," in *Proc. of the USENIX symposium on Networked systems design and implementation (NSDI)*, 2009.

[20] S. Kaune, G. Tyson, K. Pussep, A. Mauthe, and R. Steinmetz, "The seeder promotion problem: Measurements, analysis and solution space," in *Proc. of the Intl. Conference on Computer Communications and Networks (ICCCN)*, 2010.

[21] S. DiBenedetto, C. Papadopoulos, and D. Massey, "Routing policies in named data networking," in *Proc. of the ACM SIGCOMM workshop on Information-centric networking (ICN)*, 2011.

[22] W. B. Norton, "Internet transit prices - historical and projected," tech. rep., Dr Peering International, 2010.

[23] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proc. of SIGCOMM Conference*, 2004.

[24] O. Saleh and M. Hefeeda, "Modeling and caching of peer-to-peer traffic," in *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, 2006.