Content Oriented Networking: A Promising Alternative for Home Networks

Lisiane Maria Bannwart Ambiel
University of Campinas (UNICAMP)
Av. Albert Einstein, 200 – 13083-852
Campinas, SP – Brazil
lisiane@dca.fee.unicamp.br

Christian Esteve Rothenberg
CPqD – R&D Center in Telecomm.
Rod. SP 340, Km 118, 5 – 13086-902
Campinas, SP – Brazil
esteve@cpqd.com.br

Maurício Ferreira Magalhães
University of Campinas (UNICAMP)
Av. Albert Einstein, 200 – 13083-852
Campinas, SP – Brazil
mauricio@dca.fee.unicamp.br

Abstract— **This paper proposes an architecture for content oriented networking at the link layer (Ethernet) without the use of network addressing schemes. Content Routers (CR) are the basis for this architecture and perform packet caching and routing directly on content names. Different from IP-based approaches where the destination address of the content source is known and carried in the packet header, the proposed link-level architecture requests content by controlled message flooding. Questions arise concerning the introduced overhead and the overall scalability. The paper describes design choices to contain the impacts of the content-oriented flooding approach and validates the prototype implementation in a Internet-like topology with scenarios comparing the Link Layer to an IP approach; a Home Network topology is also used for validation purposes. Results suggest that content oriented IP-less architectures may be interesting for small networks such as home networks that would largely benefit from application-independent plug-and-play behavior that avoids the hazards of network configuration and management as required with IP.**

Index Terms— **Content Oriented Networking, Home Network, Caching.**

I. Introduction

Content Oriented Networking (CON), also commonly referred as Information Centric Network (ICN) is a new way to think about networking by changing the communication paradigm to an approach where content becomes the basis in replace of network location identifiers. The Internet was designed for host-centric services where the popular applications where remote terminals, file transfers or basic mail exchange. Nowadays, the vast majority of bytes filling the Internet pipes result from the request and dissemination of named pieces of content, primarily video (streaming and fully downloaded), but also music, images, and any user-generated files that find its way to Internet cloud archival and publication services. Users are not interested to know where these information came from and only need to verify its integrity and authenticity. [1]

This usage shift and the efforts to reconcile network architecture deficits in the application layer have lead to an ICN research line that pursues a better suited content distribution approach based on fundamental concepts such as, named content, name-based routing, in-network caching, content-based security. Remarkable architectures include DONA [2], CCN [3], PSIRP [4] and NetInf [5]. Common to these architectures are content names (hierarchical or flat) which are independent of their location and a security model in which content is signed by the original provider, so network elements and consumers can verify the validity of the content by verifying the signature. Challenges that need to be addressed for ICN to become deployed and used in a wider scale are related to scalability since the number of content is vastly larger than the number of hosts in the current Internet causing impact on routing and name resolution systems [5].

Some studies indicate a shifting in the address space from one billion IPs to at least one trillion content names [6]. Many concerns go into keeping down the costs related to forwarding state, content storage, and processing. Open questions include how to reduce the size of pending tables so that they can be stored efficiently on the routers, or how to discover content paths in the network and new forwarding strategies with different trade-off between the overhead and delay to retrieve data from multiple interfaces [7]. As one way to turn any new network architecture viable, overlaying over the existing IP infrastructure is the common approach to start deploying novel technologies.

In our work, we step back from looking at the global Internet and the need of common global name spaces, whether network- or content-centric, and motivate our research focus on the needs of localized networks such as home networks or campus. We expect such networks to have not only lower scalability requirements but also higher affinity in terms of requested contents, which makes the use of native content routing and caches even more attractive. Particularly, we believe that a content-oriented network architecture would fit well the networking needs of home networks. Home networks [8] differ from other Internet-style networks in significant ways, a few of them being: (*i*) lack of professional administrator or technical training, so users need help from others to set up or install new devices; (*ii*) deep heterogeneity, as new applications and devices are installed constantly increase the number of network faults and difficult the isolation problems; and (*iii*) the expectation of privacy. Besides other problems related to human needs and expectations, some technical difficulties are related to configuration and interaction between applications and the network itself.

This paper presents and evaluates an architecture for content-oriented networking at the link layer (Ethernet) based on Content Routers (CRs) without network addressing schemes. Content becomes the main attribute for all mechanisms: caching, routing, data delivery, and avoids network parameter configuration as requested by IP. The original CR architecture [9] was initially defined as an IP overlay network augmented with content routing capabilities, opportunistic caching, and a flat naming scheme providing interesting security properties. Previous work on CRs focused on the in-caching architecture and the prototype did not implement all ideas presented in [9], especially those related to routing and forwarding such as loop control when routing at the content-level. Moreover, CRs assume the existence of a fully-functional IP network infrastructure and includes the IP destination address of the content source.

The work presented in this paper revisits the CR architecture an explores a link-level approach where content requests are based on controlled message flooding and the (content opportunistic learning) routing table is used to alleviate the flooding effects. Questions arise concerning the introduced overhead related to number of messages and the overall scalability. Our design choices to contain the impacts of the content-oriented flooding approach by introducing (*i*) content announcing, (*ii*) compact data structure for pending request (Bloom filter), (*iii*) a specialized (but optional) CR with a better condition for data storage/caching to avoid message flooding and which can function as a gateway to link the network to IP environment. The CR prototype was improved to support unimplemented functionalities from the original CR proposal. Furthermore, the prototype was extended with the newly developed modules that allow for direct link layer operations without any reliance on IP.

We verify the experimental behavior of both the IP/overlay approach and the proposed link-level alternative for different conditions for caching and flooding-based search mechanisms in two different networking scenarios, an Internet-like topology and a wireless home network. In addition to the expected gains of native caching capabilities, the obtained results suggest a worth to consider trade-off between message traffic and the plug-and-play simplicity of Ethernet.

The rest of this paper is organized as follows. Section 2 presents some background and definitions related to ICN. Section 3 describes the proposed link layer content router architecture. Section 4 presents the experimental evaluation work. Finally, Section 5 concludes the paper.

II. Background

The following key concepts have been proposed in the context of ICN [5]:

Named Data Objects: represents the different contents such as WEB pages, documents, movies, photos, songs; all types of objects that can be stored and accessed via computing devices. The content name is kept independently of its location, storage method, or application.

Naming: names are used for identifying objects and is the bases for request, distribution and authentication process. Two main naming schemes are commonly considered: *hierarchical* and *flat* namespaces.

Routing and Forwarding: name-based routing approaches directly route the request message from the content requester to one or multiple sources in the network based on the requested object name. The routing algorithm heavily depends on the properties of the namespace. After the source has received the request message, the data is routed back to the requester. Another approach is using a name resolution service that allows routing and forwarding to take place in a new namespace that can be content- or network-based.

Caching: all nodes potentially have caching capabilities, including infrastructure nodes and end-user nodes like computers, home devices, and mobile terminals. Content requests can be satisfied by any node holding a copy in its cache. The caching function is generic, i.e., it is application-independent and applies to all forms of content, including user-generated content.

Caching may help reducing the transport cost for network providers and improving end-user delivery performance. The availability of different replicas depends on several factors, like content popularity, cache replacement policies, and so on, and is clearly impacted by the request forwarding policy. Study comparing deterministic exploitation of forwarding information towards an ”known” copy and a random network exploration towards an ”unknown” copy, via request flooding, shows the possibility of using these strategies besides a hybrid one for an intra-domain point of view [10].

Naming schemes can be classified as hierarchical, flat, and attribute-based [11]. Each approach partially attends fundamental naming scheme requirements such as persistence, scalability, and user intelligibility [12]. There are trade-offs related to naming that affect routing and security as well. Other fundamental aspects of ICN proposals relate to basic primitives and security model [13]. More ore less, all ICN designs are built around two basic interface primitives that resemble the notions of `PUBLISH` and `SUBSCRIBE` but acting on the content names and offering both “fetch” operation (retrieving content previously published under that name) and ongoing “subscribe” operation (retrieving all future content published under that name). As for security, ICN designs commonly adopt a content-centric security model in which content is signed by the original content provider, so network elements and consumers are able to verify the validity of the content merely by verifying the signature for a given piece of content.

ICN represents a challenge for today’s router technologies in order to support wire speed name-based forwarding and packet-level caching [6]. Shifting the address space from one billion IPs to at least one trillion content names would cause a clear increase of the required routing state. On the positive side, allowing routers to serve content from local caches can potentially alleviate the frequency of forwarding operations. In addition, the use of probabilistic data structures such as Bloom filters appears as one promising approach to implement the required cost-effective packet forwarding methods. Based on a systematic evaluation of the suitability of existing software and hardware in commercial routers, related work [6] concludes that today's routers could only support a fraction of the state required to operate an ICN at Internet scale. Nevertheless, by reducing the scope of a network deployment, i.e., from Internet scale to Content Delivery Network (CDN) or Internet Service Provider (ISP) scale, today's routers could be extended to become actual content routers. Our work goes further down the scaling avenue and looks initially at LAN scales motivated by the networking needs of home environments. We belief that similarly how the Internet grew up from a lab-scale environment, ICN shall be first pursued in smaller scales letting practical experience drive the paths ahead.

III. Link Level Content Router Architecture

The main elements in the proposed architecture are the so-called *Content Routers* (CR), which in their role of routers are responsible for content request and response forwarding, and opportunistic content caching. The original proposal [9] is based on an IP overlay augmented with in-network caching and a flat naming scheme. Two levels of request routing are defined: one directly on the content identifiers, and a second one based on the IP of content sources (original providers). Data delivery is based on network-level routing using the IP address of content requesters.

In our work, we propose CRs working at the link layer which requires a quite different approach to the problem of routing. For request forwarding, we introduce flooding via all available interfaces whenever the next hop of a given content identifier is unknown. State information related to pending requests is kept in compact data structure (Bloom filter) to make route back to content requesters for data delivery. Also, content announcement and registration are introduced and a specialized CR is proposed for better data storage/caching avoiding message flooding and also to work as a gateway to link the network to IP environment. Other elements of the network are *clients*, which request for content, and *servers*, which hold the content and attend the requests. These architectural elements apply to both environments (IP and link layer). Naming, caching, and security are kept unmodified and will be later introduced for completeness. The fundamental differences between the original IP/overlay approach and our link level (Ethernet) proposal will be further detailed after introducing some basic definitions:

***Data Chunk*:** the basic unit of communication used in the architecture. Data chunk or just chunk is a piece of data that is identified by a *CryptoId*, or *ChunkId*.

***Cryptographic Identifier* (CryptoId):** Content identifier generated by content providers based on data chunks using a cryptographic hash function.

***Metadata*:** A meta information structure where the *chunkIds* related to a content are aggregated. It also contains some additional information about the content such as version and validity. Clients need to retrieve the content metadata prior to being able to request and receive the data chunks.

***Caching Threshold*:** Determines the probability of a CR to opportunistically store a content it forwards.

***Neighbor Zones* (NZones):** allow CR to divert regular chunk requests up to n hops away to CRs that may have the content before going directly to the server (IP). For the link layer approach it helps to control message flooding by exploiting the information in routing tables before flooding the request (Ethernet) in an exploration mode.

***Visited Neighbors*:** In-packet Bloom filter data structure (i.e carried in the message header) containing a mark of each CR visited by a message to prevent loops. It applies to all message types.

A. Main architectural characteristics

Naming: a flat identifier *chunkId* is used to name pieces of content. This kind of identification meets the persistence requirements because location information is not coupled and the unique identification for the same collection of bits is provided. These identifiers support a content-oriented security model that uniquely binds the *ChunkId* to the content bits and the content provider.

Caching: the CR inspects a CR message header of every data packet in transit and stores a copy of it with a certain caching probability (*caching threshold*). Further requests can be served by the cache data in the CR increasing the content availability and reducing the content retrieval time.

Security: The *chunkIds* are auto certified identifiers. The CR architecture uses Merkle Tree to provide partitioned content authentication in multi-source content retrieval scenarios. It is a binary tree constructed over the data blocks with a root hash on the top of tree. This root hash is a digital signature securely retrieved by clients used to authenticate the received chunks (cf. Merkle trees [9]).

Routing: only `REQUEST` messages are actually routed with *chunkIds* being the forwarding identifiers. A `REQUEST` hitting a CR with a cached copy of the requested data chunk results in the CR generating a `RESPONSE`. In the proposed link level approach, when a content entry is not in the routing table, the `REQUEST` message is flooded to all available interfaces (except the incoming one). CRs keep the information on pending *chunkIds* per incoming interface, similar to the CCN Pending Interest Table (PIT) [3]. This state is compressed using a Bloom filter to reduce the memory requirements. In the IP proposal, when a content-based routing entry is not present, the `REQUEST` is forwarded to the server (destination IP address) based on IP routing protocols. Hence, in the IP approach, there are two 'levels' of routing: an initial content-based, and then IP-based towards the server when content routing is unknown or the *NZones* limit is reached.

Data delivery: For the link level scenario, the `RESPONSE` is sent back using the reverse path and free of loops, like data packet forwarding in CCN [3]. CRs check the pending list for each interface and send `RESPONSE` packets back via the matching ones. The *chunkId* is removed from the Bloom filter data structure. Duplicated messages generated by other content sources due to flooding of `REQUEST` are consumed. In the IP environment, data is routed to the customer using the

REQUEST source IP address which becomes the destination address of RESPONSE packets.

Content registration: CRs can opportunistically learn about routes when handling RESPONSE messages, a process similar to MAC learning in Ethernet. Only the best route in terms of hops is kept in the routing table. Optionally, servers may announce their content by using ANNOUNCE_CONTENT messages containing *chunkIds*. Both processes allow CRs to fill their routing tables with *chunkId* entries and avoid *flooding* by forwarding requests only in the direction of the neighboring CRs which have announced that information.

B. *Content Router (CR) Operations*

The Content Router (CR) is a network node providing content routing mechanisms and operates as follows:

REQUEST messages are directly responded by CRs when the *chunkId* and related data is available in its cache (*cache-hit*). Otherwise (*cache-miss*), it will look for an entry in the neighbor table and will proceed with the forwarding process as explained before. So, the forwarding process will take into account the information in the neighbor table and the neighbor zones (NZones) parameter.

RESPONSE messages are routed back to the content requesters (clients) based on plain IP routing (in case of IP environments) or based on the request pending information (in case of the proposed Ethernet environment). The data might be cached by CR when handling RESPONSE messages based on the *caching threshold* parameter which is applied in a probability function in order to no overload the local cache since the capacity is limited.

There is no signaling among CRs, it means that they do not work in a cooperative mode related to *caching*, so a piece of content can be stored in all CRs handling a RESPONSE message. We introduce the Super CR as a specialized CR to deal with caching of routes and contents in a special way, and to increase scalability by avoiding flooding where possible. This kind of CR is optional and allows to start introducing a hierarchy in the network using a simple control plane to inform CRs about its presence (ANNOUNCE_SCR). From the point of view of normal CRs, when Super CRs are present and the requested content is not in the routing table, the routing process will try first the Super CR instead to start flooding. Any CR can be set as a Super CR and may be a convenient way to take advantage of its network position for a given topology and resource availability constraints.

IV. Experimental Evaluation

A. *Implementation*

The CR prototype is implemented in C as a background Linux service. It gives us full control on operations and is easier to add and customize parameters. The prototype code was improved to support unimplemented functionalities from the original CR proposal. Furthermore, the prototype was extended with the newly developed modules that allow for direct link layer operations without any reliance on IP. The implementation is basically based on two main modules and

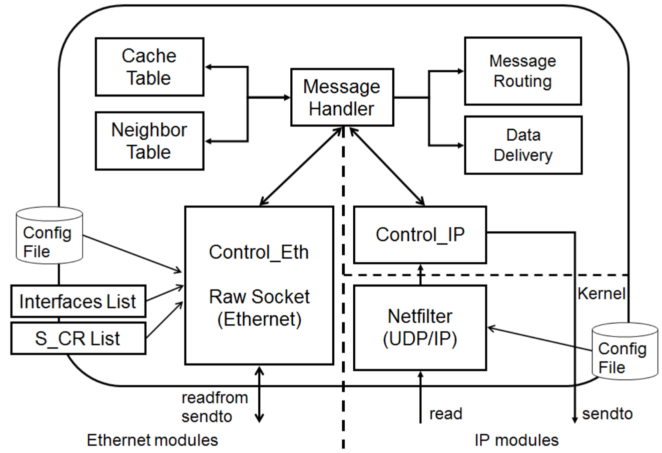


Fig. 1. CR Internal Modules

a number of data structures, as illustrated in Figure 1. The Control module handles messages received from the network interfaces and serves Ethernet using Raw Sockets and Netfilter to intercept messages in the case of the IP environment. The Message Handler module processes each different message type and interacts with the two main data structures: the *Cache Table* and the *Neighbor Table*, both of them indexed by the content identifier (*chunkId*) and with limited capacity.

For the Ethernet environment, the CR keeps a list of Super CRs (S_CR List) and the list of available interfaces (Interfaces List). Each interface keeps the pending requests by using a counting Bloom filter with 4-bit cells [14]. RESPONSE messages are forwarded to all interfaces which have the related *chunkId* appearing as a pending request in the Bloom filter. After forwarding, the corresponding *chunkId* entries are removed from the Bloom filter.

B. *Experimental Setup*

In order to evaluate the prototype, two topologies were considered: (1) Internet-like CR arrangement over high capacity wired links, and (2) home network with a single CR acting as wireless access point and Internet gateway.

The Internet-like topology shown in Figure 2 includes twelve CRs, four servers and eight clients. The CR network elements are a set of virtual machines created using XEN [1] to allow the CR act as Debian GNU/Linux routers. When running in *overlay*/IP mode, the routers also execute the open-source routing stack Quagga[2] with OSPF protocol in a single zone configuration. The CRs follow a simple access/aggregation and core organization in two domains with two "peering" links. This topology puts the link layer approach in disadvantage compared to IP because it requires more hops which introduces more potential flooding points. The topology serves well however to compare both approaches and resembles a "condo-networking" scenario where multiple home networks become interconnected to let users exchange content directly. The test procedure considers a sequence of 30 requests for

[1]http://www.xensource.com
[2]http://www.quagga.org

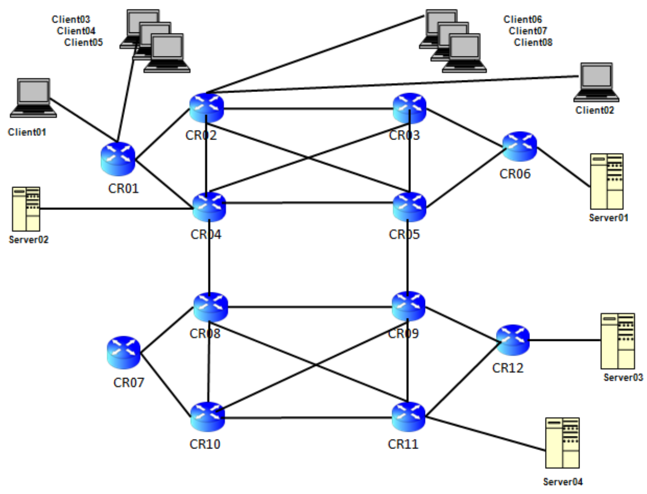
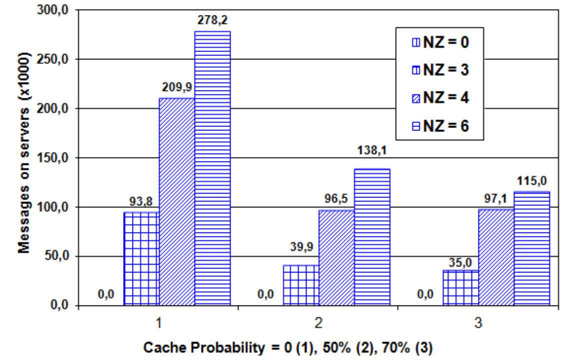
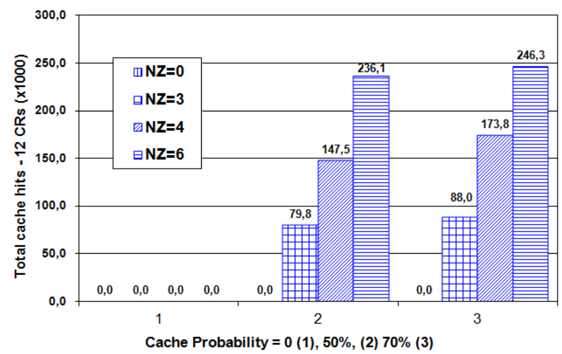


Fig. 2. Experiment participants - Topology



(a) Requests solved by servers



(b) Requests solved by CRs (cache hit)

Fig. 3. REQUEST messages handling (Ethernet)

different content files (text, image, songs) following a power-law distribution: 10 files are used for requests but some of them have a higher popularity and will be requested more times. The CR tables (route and cache) were configured to support 10K entries. This means that entry replacements occurs on both tables since the total number of content chunks for each tests run is around 25K. This setup, although simplistic, allows to consider the likely scenario of limited caching and routing table sizes.

The wireless home network scenario is illustrated in Figure 6. The popular NetEm [15] tool was used to emulate the lossy behaviour of the wireless link between the clients and the WiFi-serving CR. Different packet loss rates were tested to randomly drop a percentage of the packets before they are queued. The CR network elements are running in real machines. The tests presented in this paper consider three different client devices requesting, one after the other, a large size video file (92 MB). The tests run without cross-traffic and allow to evaluate in isolation the performance of the CR with probabilistic caching over a wireless link. The use case would correspond to a popular content being shared between family members (e.g., via a social app) or accessed through different devices in a short period of time.

The initial condition in all scenarios under study is that CRs do not have any information at all about available contents, i.e. routing and cache tables are empty at experiment start.

C. Results

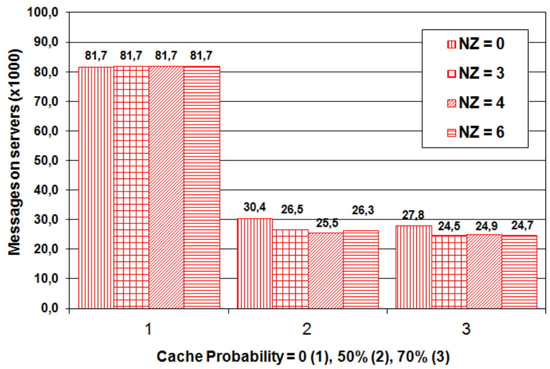
1) Internet-like Topology (Wired): To reduce the amount of results presented in this paper, we will focus on a single $Client01$ (in Fig. 2) issuing requests. A single test scenario is based on a combination of different parameters that may cause an impact in the results: (i) caching threshold (Prob: 0%, 50%, 70%), neighbor zones (NZ = 0, 3, 4, 6), and flooding enabled (yes/no). The metrics extracted on each experiment round include: servers and CR utilization (# and type of handled messages), distance from content source point (# hops), and total time to transfer a content since the first client request.

a) Link Level Approach: Figures 3(a) and 3(b) show the behaviour for link layer CR proposal in terms of number of messages handled for varying *caching* probability and *Neighbor Zones – NZ*, recalling that the *flooding* behaviour will stop when NZ reaches zero and *Visited Neighbors* is also checked to control loops and *flooding*.

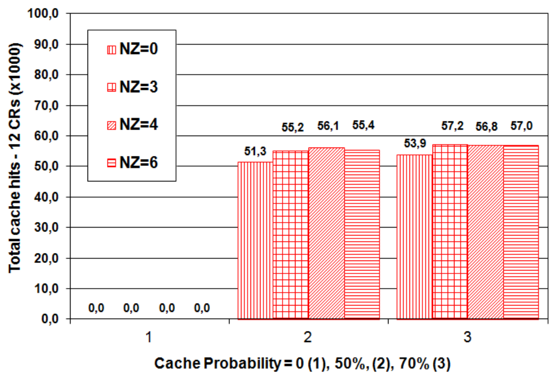
In the scenarios without caching in the CRs (1), requests are solved only at servers and the NZ parameter value determines the following impacts: $NZ = 0$: nothing is recovered as expected. $NZ = 3$: generates a low number of messages due to resolution happening in the closest server ($Server02$). $NZ = 4$ or 6 : the number of messages increase in the network due to larger flooding zones and resolution is distributed along the serving nodes ($Server01$ to $Server04$).

When opportunistic caching is enabled with a certain probability, it is possible to verify that the request resolution happens on CRs, reducing as expected the load on servers. The variance on caching probability from 50% to 70% does not cause big differences, suggesting that uncoordinated CR probabilistic caching may work well in practice without requiring to commit every possible storage resource. The use of NZ causes an expected increase of messages in the network due the flooding trying to find the content when it is not known by the CR. The controllable flooding mechanism also contributes to more responses being generated; some of them will be ignored by CRs which already processed the same message and consequently removed the *chunkId* from the pending list.

b) IP/Overlay Approach: Figures 4(a) and 4(b) show the results obtained for the original CR proposal based on an IP



(a) Requests solved by servers



(b) Requests solved by CRs (cache hit)

Fig. 4. `REQUEST` messages handling (IP)

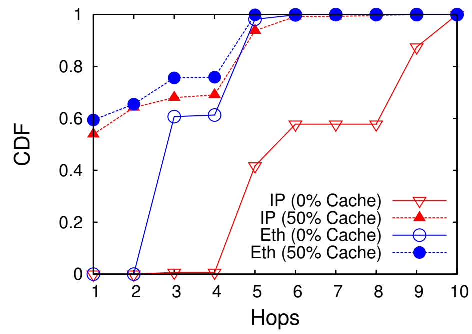


Fig. 5. IP vs. Ethernet: Fraction of content delivery distance in hops.

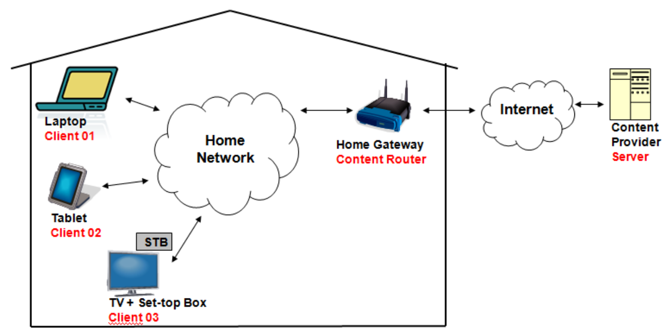


Fig. 6. Home Network Experiments. Three clients requesting content over a wireless link with a single CR acting as the Wifi access point and home gateway to the Internet.

overlay environment. Noteworthy, OSPF protocol messages (e.g., Hello, LSA, LSU) are not included as network control messages, presenting thus a favorable condition.

In the scenarios where caching is disabled (1), the requests are resolved by the servers and the NZ value is not taken into account since in the IP environment the server address is known and the request is forwarded via IP routing.

When caching is enabled, the major part of requests are resolved by CRs and the servers' load is reduced as expected. Again, there is no big advantage to use 70% rather than 50% for caching probability. There are some benefits when varying NZ to increase the request resolution range.

c) Link Layer vs. IP: We can observe in Figures 3 and 4 that although the amount of messages are notably different in both approaches due to the *flooding* effects. Once caching is introduced the overall behaviour tends to be the same. As the number of requests for the same content increases, the probability of a CR caching the content increases as well. Since the caching probability is randomly chosen, it is unpredictable where the content will be placed and whether a given request will be forwarded via a CR holding that chunk.

Figure 5 compares both environments considering the distance until the request serving node (CR or Server) in hops from the requesting client. It is worth to highlight that the link layer approaches achieves shorter ways (60% of times with 4 or less $hops$) than IP (60% of times with 6 or less $hops$) when there is no caching on CRs. The explanation being that flooding contributes to finding the nearest content source. The results are comparable where cache is enabled and most of the content is delivered within 3 and 4 $hops$). For some cases, the distance from client to content source is even greater than the NZones defined for the test scenario ($NZ = 6$). This is a consequence of an unsuccessful content search attempt via content routing, which increases the number of hops (via IP routing) until the original server.

2) Home Network Scenario (Wireless): Table I shows the test results for the home networking scenario illustrated in Figure 6. The CR opportunistic caching probability is set to 50% and the three clients request the same 92 MB content file in sequence. As expected, caching reduces the server load on the 2^{nd} and 3^{rd} requests in all cases. Caching also causes a positive effect as packet loss rates increase and requests re-sent due to client timeouts are served with CR cache. The observed download time decreases in subsequent requests even for higher wireless loss rates. Since the client application does not have any flow control mechanism (chunks are request one after the other) including delays in the CR-Internet link would not bring new insights.

This test suggests that shared contents in a family scenario supported by a plug-and-play content-oriented link layer architecture can contribute to reduce bandwidth demand in the access and core networks and accelerate content retrieval. Future test extensions will consider client flow control, more realistic traces, and a mesh topology of home CR gateways in the spirit of condominium networking scenario.

TABLE I

TEST RESULTS FROM HOME NETWORKING SCENARIO WITH ONE CR-BASED WIRELESS GW ($Cache = 50\%$).

	Packet Loss = 0.01%			Packet Loss = 0.1%			Packet Loss = 1%			Packet Loss = 5%		
Request #	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}
Req. served by Home CR	15	33536	50751	42	33788	50674	348	33697	50644	1708	35096	51955
Req. served by Server	68377	34856	17641	68350	34604	17718	68044	34695	17748	66684	33296	16437
Timeout errors on client	8	8	8	71	71	66	701	668	688	3526	3547	3579
Content tx time (seconds)	722	533	383	753	561	412	1066	861	724	2480	2304	2170

V. CONCLUDING REMARKS AND FUTURE WORK

We have presented and discussed an alternative architecture for Content Routers aiming at link layer (Ethernet) environments that circumvent the cost and operational hazards of an IP infrastructure. The obtained results show that it is possible to work directly on link level replacing a routing based on location (and the companion control and management planes) with native content routing and controlled flooding. The side effect being the overhead of messages generated by the flooding mechanisms which need to be contained by the content routing strategy and network topology scoping.

The ideas presented in this paper are applicable to other ICN architectures that wish operating under the IP layer. While a detailed analysis is out of the scope of this paper, a few related works are worth to mention.

The CONET architecture [16] suggests the link layer as an alternative approach to overlay and IP integration. Similar to the content route learning at CRs, CONET uses limited tables to cache routes but introduces a centralized routing engine to serve all nodes in a sub-system with unknown route information. The idea of specialized nodes matches the proposed Super CR which could act as gateways to integrate Ethernet with IP networks.

Recent work [10] comparing deterministic forwarding towards a "known" copy and a random network exploration towards an "unknown" copy (e.g., via request flooding), shows the possibility of using these strategies besides a hybrid one for intra-domain scopes. Our *Neighbor Zones – NZ* concept is similar to the *Kpercentile* proposed in the hybrid strategy. In line with our results and related work [7], authors [10] report that neighborhood exploration can improve delivery performance by discovering close-by content replicas.

As future work we intend to improve the Super CR to make it work as a gateway to mix Ethernet/IP environments since typical home networks have different devices, all connected to the Internet through a single point, the home gateway. We will also work on strategies for routing and cache table optimization, e.g., LRU mechanisms in addition to novel Bloom filter inspired data structures [14].

Along the home networking application scenario, we intend to combine wired and wireless connectivity where any home device can work as CRs themselves, contributing with caching capacity in a collaborative way and having a Super CR acting as a gateway to IP and the rest of the Internet whenever content cannot be provided locally. We will also investigate the application of Wi-Fi Direct technology to allow direct content transfers between devices to deliver a cloud-less, application-independent "family dropbox" type of service. New network architectures that directly support the user's intention as a native semantic like in the case of ICN proposals appears as one of the promising approaches to rethink home networking.

REFERENCES

[1] C. E. Rothenberg, F. L. Verdi, and M. F. Magalhães, "Towards a new generation of information-oriented internetworking architectures," in *Proceedings of the 2008 ACM CoNEXT Conference*, ser. CoNEXT '08. ACM, 2008, pp. 65:1–65:6.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM CCR*, pp. 181–192, 2007.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *CoNEXT'09*, Rome, Italy, 2009, pp. 1–12.

[4] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line speed Publish/Subscribe Inter-Networking," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, ser. SIGCOMM '09. ACM, 2009, pp. 195–206.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[6] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proceedings of the ACM SIGCOMM workshop on ICN*. ACM Press, Aug. 2011, pp. 44 –49.

[7] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM CCR*, vol. 42, no. 3, pp. 62–67, Jul. 2012.

[8] W. K. Edwards, R. E. Grinter, R. Mahajan, and D. Wetherall, "Advancing the state of home networking," *Communications of the ACM*, vol. 54, pp. 62–71, Jun. 2011.

[9] W. Wong, M. Giraldi, M. F. Magalhaes, and J. Kangasharju, "Content Routers: Fetching Data on Network Path," in *IEEE ICC*. IEEE, Jun. 2011, pp. 1–6.

[10] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino, "Exploit the known or explore the unknown?" in *Proceedings of the second edition of the ICN workshop*. ACM, Aug. 2012, pp. 7 – 12.

[11] Choi et al., "A Survey on content-oriented networking for efficient content delivery," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 121–127, Mar. 2011.

[12] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proceedings of the ACM SIGCOMM workshop on ICN*, ser. ICN '11. ACM, 2011, pp. 1–6.

[13] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: Seeing the forest for the trees," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, Cambridge, USA, 2011, pp. 1–6.

[14] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and Practice of Bloom Filters for Distributed Systems," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 1, pp. 131–155, 2012.

[15] A. Jurgelionis, J.-P. Laulajainen, M. Hirvonen, and A. Wang, "An empirical study of netem network emulation functionalities," in *Proceedings of 20th ICCCN*, 31 2011-aug. 4 2011, pp. 1 –6.

[16] A. Detti, N. Blefari Melazzi, S. Salsano, and M. Pomposini, "CONET: a content centric inter-networking architecture," in *Proceedings of the ACM SIGCOMM Workshop on ICN*, 2011, pp. 50–55.