Link Layer Content Router

Lisiane Maria Bannwart Ambiel , Christian Esteve Rothenberg (Coorientador)
Maurício Ferreira Magalhães (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Av. Albert Einstein, 400 – Campinas, SP, Brazil – CEP 13083-852

{lisiane,christian,mauricio}@dca.fee.unicamp.br

Abstract – This paper proposes an architecture for content oriented networking at the link layer without the use of network addressing schemes. Different from IP environments where the destination address of the content source is known, 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 some scenarios compared to an IP approach and also in a home network scenario.

Keywords – Information Centric Networking; Caching

1. Introduction

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 were 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, but also music, images, and user-generated files. Users are not interested to know where these information came from and only need to verify its integrity and authenticity. 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 [4], CCN [2], PSIRP [3] and NetInf [1]. Common to these architectures are in-network caching, content names 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.

This paper presents and evaluates an architecture for content-oriented networking at the link layer 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 [5] 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 assumed the existence of a fully-functional IP network infrastructure including 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 routing table is used to alleviate the flooding effects. Our design choices to contain the impacts of the content-oriented flooding approach introduces (*i*) content announcing, (*ii*) compact data structure for pending request (Bloom filter), (*iii*) a specialized (but optional) CR with a better condition for caching and which can function as a gateway to link the network to IP environment. We verify the experimental behaviour 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.

2. 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 [5] is based on an IP overlay augmented with in-network caching and a flat naming scheme. 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 network elements are *clients*, which request for content, and *servers*, which hold the content and attend the requests. These elements presented in Fig. 1 apply to both environments.

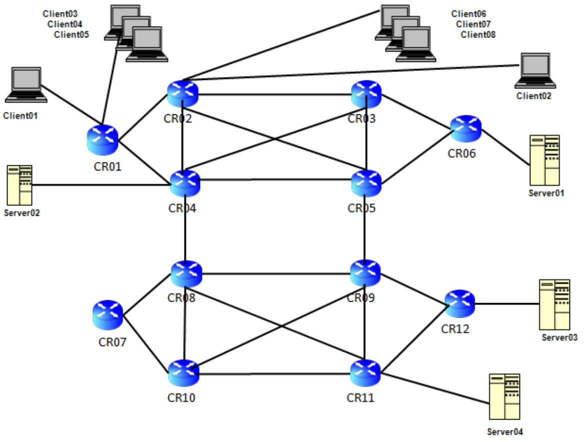


Figure 1. Internet Like Topology Experiment

2.1. 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. The *ChunkId* is generated by content providers based on data chunks using a cryptographic hash function.

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 [5]).

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) [2]. 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 [2]. 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 opportunistically learn about routes when handling `RESPONSE`. Servers may announce their content by issuing `ANNOUNCE_CONTENT` messages. Both processes allow CRs to fill their tables with *chunkId* entries and avoid *flooding*.

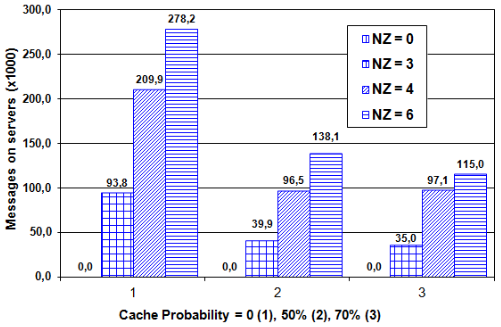
3. Results

3.1. Internet-like Topology

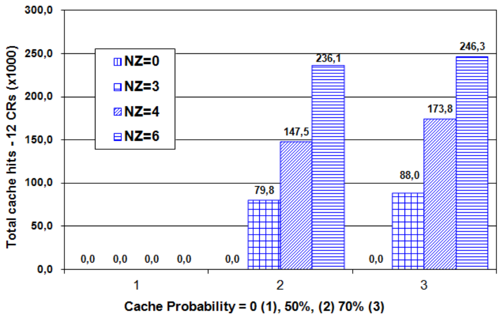
The first experimental environment is presented in Fig. 1. We will focus on the results from a single $Client01$ issuing requests. The test scenario is based on a combination of different parameters that may cause impact in the results: caching threshold (Prob: 0%, 50%, 70%) and neighbor zones (NZ = 0, 3, 4, 6). 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.

Link Level Approach: Figures 2(a) and 2(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, 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 responses being generated will be ignored by CRs which already processed the same message and consequently removed the *chunkId* from the pending list.



(a) Requests solved by servers



(b) Requests solved by CRs (cache hit)

Figure 2. `REQUEST` handling (Ethernet)

IP/Overlay Approach: Figures 3(a) and 3(b) show the results obtained for the original CR proposal based on an IP 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.

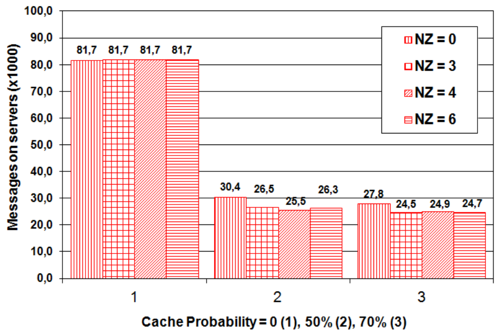
3.2. Home Network Scenario

In this scenario, three clients request content over a wireless link with a single CR acting as the Wifi access point and home gateway to the Internet. The popular NetEm tool was used do emulate the lossy behaviour between clients and CR.

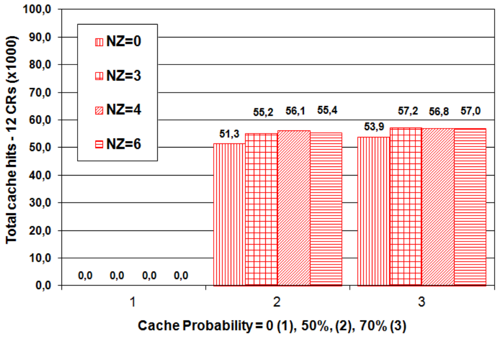
Table 1 shows the test results for the home networking scenario illustrated in Figure 4. The CR opportunistic caching probability is set to 50% and the three clients request the same 92 MB content file in sequence. As expected, caching re-

Table 1. Test results from home networking scenario with one CR-based wireless GW ($Cache = 50\%$).

	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}
Req. served by Home CR	42	33788	50674	348	33697	50644	1708	35096	51955
Req. served by Server	68350	34604	17718	68044	34695	17748	66684	33296	16437
Timeout errors on client	71	71	66	701	668	688	3526	3547	3579
Content tx time (seconds)	753	561	412	1066	861	724	2480	2304	2170



(a) Requests solved by servers



(b) Requests solved by CRs (cache hit)

Figure 3. `REQUEST` handling (IP)

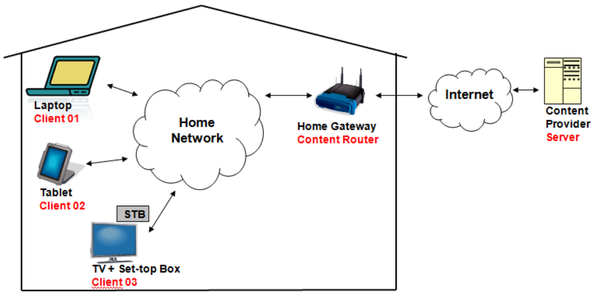


Figure 4. Home Network Experiments

duces 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.

4. Conclusion

We have presented and discussed an alternative architecture for Content Routers. 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. Test with home network environment 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. Similar benefits is expected considering a mesh topology of home CR gateways in the spirit of condominium networking scenario.

References

[1] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. A survey of information-centric networking. *IEEE Comm Magazine*, 50(7):26–36, 2012.

[2] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *CoNEXT'09*, pages 1–12, 2009.

[3] Petri Jokela, András Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, and Pekka Nikander. LIPSIN: Line speed Publish/Subscribe Inter-Networking. In *Proceedings of the ACM SIGCOMM 2009*, pages 195–206. ACM, 2009.

[4] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. *ACM SIGCOMM CCR*, pages 181–192, 2007.

[5] Walter Wong, Marcus Giraldi, Mauricio F. Magalhaes, and Jussi Kangasharju. Content Routers: Fetching Data on Network Path. In *IEEE ICC*, pages 1–6. IEEE, June 2011.