

# Towards Content-Centric Geometric Routing

Wouter Tavernier\*, Sahel Sahhaf\*, Didier Colle\*, Mario Pickavet\* and Piet Demeester\*

\* Ghent University, IBCN-iMinds, INTEC, Gaston Crommenlaan 8 bus 201, B-9050 Gent, Belgium

Email: {wouter.tavernier, sahel.sahhaf, didier.colle, mario.pickavet, piet.demeester}@intec.ugent.be

**Abstract**—Content delivery is a crucial feature of existing cloud and telecom networks. This is confirmed by the tremendous success of media streaming services such as Spotify and Netflix, as well as the content and file-distribution systems such as BitTorrent. A recurring problem in these type of network services is about keeping the protocol overhead as low as possible while maximizing the efficiency of such systems in terms of network delay to customers.

In this paper we propose the use of a routing system-inferred coordinate system to improve: i) content server selection upon receiving content requests, and ii) the mapping of content to servers/caches. We describe the required protocol mechanisms, and evaluate potential gains using coordinates of Geometric Tree Routing and compare it to pure IP-based mechanisms or measurement-based content systems relying on coordinates. The proposed approach can be further extended in order to include alternate geometric systems for example supporting hyperbolic geometries.

## I. INTRODUCTION

Content delivery has become an essential aspect of current cloud and telecom networks. The wide range of media services such as Spotify, Netflix, or content distribution systems such as BitTorrent are omnipresent on our smartphones, tablets, smart tvs and pcs. Most of these services somehow rely on caching and replication. However, being proprietary and deployed in isolation, these content distribution technologies do not allow to uniquely and securely identify named information independently of the distribution channel. In many cases, these technologies are implemented as an overlay, leading to needless inefficiency.

Content-Centric Networking (CCN) [1] aims at enabling data to become independent from their network/physical location, application, storage support but also their transport enabling in turn in-network transparent caching and replication. CCN design focusing on name-based data access therefore gets increase interest from the network research community. The conventional CCN concept of proposing name-based data access at the network level where content names get overloaded with a locator semantic exacerbates the memory scaling limits of shortest path routing already observed in many experimental studies and theoretic investigations. This approach is confronted with the issue that these name spaces have not been designed to sustain forwarding performance and scaling compared to IP address space which can be efficiently aggregated, summarized and translated.

This paper argues that efficient and scalable content-based forwarding requires to keep addresses or locators (instead of names) as information unit for routing/forwarding decision.

The *name* of a resource indicates what we seek and an *address* (*locator*) indicates where it is. In addition, the selection of the addressing space for CCN should also elevate the memory scaling problem induced by shortest path routing. Therefore, we propose an alternative model for Content-Centric Networking (CCN) in which content names are assigned locators and content routing is performed on these locators. Routing decisions being performed on locators, this model avoids name-to-locator resolution by intermediate nodes but instead considers that name resolution provides sufficient information for the receiver to determine the best route towards the selected destination.

The rest of this paper is structured as follows. Section II details the architecture of the proposed CCN approach and provides elements of the design and different steps of the content-centric geometric routing approach. Next, the concept is evaluated in a simulation setup, as well as validated at a lower level through emulation, both documented in Section III. Section IV recapitulates the learned lessons and formulates some potential future work.

## II. ARCHITECTURE AND DESIGN

A Content-Centric Networking system consist of two fundamental functions: the localization function and the routing function. The *localization function* performs by using information name as input to determine the corresponding locator, say  $x$ . The *routing function* (distributed by nature) determines locally and independently for any destination the adjacent node along a loop-free path such that incoming messages directed to this destination can reach it. Obviously, by unifying the interpretation of the locator  $x$  and its interpretation of such information as distance  $d$  from the end-point or terminal perspective, a distance metric can be defined. Putting all these ingredients together, leads to consider a locator space that can be processed at the end-points by a localization function and at intermediate nodes by a routing function.

### A. Geometric routing and coordinate systems

It remains to determine which locator value space would best fit this role. An obvious choice would be a so-called topology dependent label space. However, such value space is prone to renumbering even in case of non-local topological change; hence, unsuitable. Ideally, geometric space whose values are coordinates assigned in a way which closely reflects the topology-inferred graph distances (e.g., hop count or weighted path length) while being robust to topology changes.

Several attempts have been made in order to identify geometric spaces for applying *greedy routing* in order to reduce

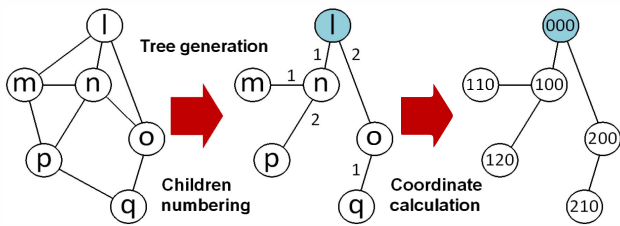


Fig. 1. Greedy Tree Routing coordinate construction

memory requirements of forwarding nodes in the context of wireless ad-hoc networks, sensor networks, or large-scale networks such as the inter-AS Internet topology [2], [3], [4], [5]. Internet topologies have been shown to follow hyperbolic geometries [6], and several approaches have been proposed in order to optimally estimate and characterize the resulting geometric space. Robert Kleinberg proposed a simple spanning tree-based mechanism to embed<sup>1</sup> nodes of any topology into a hyperbolic geometry, which leads to low stretch in scale-free graphs when combined with a local greedy forwarding policy (i.e. every node forwards packets to the neighbors which reduces the distance maximally towards the coordinate of the indicated destination). In addition, Kleinberg proved that such distance-decreasing path is always possible at any node for any destination the network [2]. When an embedding has such properties, it is referred as a *greedy embedding*.

However, in many cases, simple geometric approaches yield similar performance. In this paper we evaluate the potential value of relying on the coordinates produced by Geometric Tree Routing (GTR) [7].

In GTR, network nodes receive a coordinate which is based on their location in a selected spanning tree with respect to a selected root node. Various distributed approaches are known to generate such a setup [8]. The resulting coordinates are created as follows:

- 1) The root node has coordinate  $(0, \dots, 0)$  and numbers its children from 1 to  $d$  (child number)
- 2) A coordinate of a child is such that its own child number is assigned to the last non-zero number of the coordinate of its parent

The above numbering process is illustrated in Figure 1. Using this coordinate system, the distance between two nodes is defined as the distance in the defined tree between the two nodes. This scheme allows greedy forwarding by forwarding any received packet towards the neighbor which maximally reduces the distance towards the coordinate of the destination. This is not the same as forwarding on the tree because shortcuts can be taken. In addition, the resulting embedding is greedy, as a distance-decreasing path is always possible. As evaluated in [8], the choice of the spanning tree may impact the resulting performance of the system. For this reason, we focus on the tree which has minimal depth with a root node

which has large(st) degree in order to reduce average stretch and reduce coordinate memory. Trees with lower depths result into coordinates with less components.

The described process might be perfectly integrated into existing routing logic, either as additional protocol, or embedded as particular case of existing distance/path-vector protocols or on top of link-state routing protocols. As indicated in the next section, by enabling the use of coordinates in the forwarding plane of network nodes, more intelligent (content) routing decisions can be made compared to when only traditional IP addresses are available, as these coordinates are in fact a compressed form of their location within the topology.

Note that the described coordinate system is actually a direct result of the routing system (routing-inferred), compared to alternate coordinate systems for content distribution systems such as Vivaldi [9], which rely on active round-trip measurements between end-hosts and/or servers. It is important to note that the mechanism we propose directly reflects topology changes as integrated part of the routing system, while approaches such as Vivaldi are re-active and prone to additional measurements and network traffic as well as inference mechanisms (in order to estimate coordinates between nodes for which no direct RTT delays are measured) in order to create a consistent coordinate system.

### B. Content-Centric Geometric Routing

The operations underlying the geometric information routing model consists of three steps: i) registration, ii) resolution and iii) delivery. Figure 2 depicts the content object registration and resolution procedure. In the first step, the entity hosting the content object registers its corresponding locator to a name resolution server. In the second step, the requester sends a request to the resolution server including the content object name. The name-to-locator resolution server replies by providing a list of content locators. In the delivery step, the requester selects the closest content server from the list provided by the resolution server and sets up the connection with the corresponding content server. When the resolution server also acts as a dispatcher (e.g., http forwarder), which is a realistic assumption for many of current media services, this selection can be made by the dispatcher itself. In this traditional scenario, this selection process might rely on inferring a geographic location of the involved IP addresses. However, this relationship changes over time and does not guarantee that geographical closer servers are indeed a better choice in terms of end-to-end delay towards the requester. In the scheme we propose, the coordinate system resulting of geometric routing is used in all content-related steps to make selection decisions. Next, we detail the design and operations of the proposed geometric information routing model.

Assume a source node associated to coordinate  $x$  initiates a query for a given content object and issues a request following the process illustrated in Figure 2. As a copy of a given object can be available at multiple servers which is often the case in information systems, the requester may receive multiple locators in response, i.e., set of coordinates  $\{y_i\}_{i=1}^n$ .

<sup>1</sup>assigning coordinates to the nodes within the geometric space

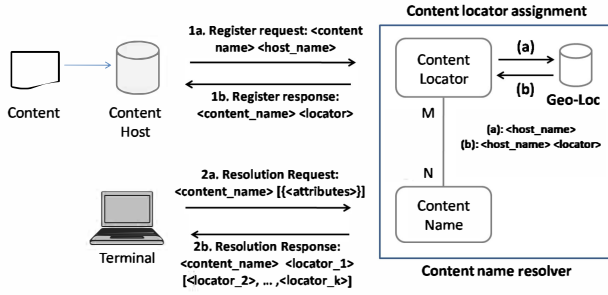


Fig. 2. Content name registration and resolution mechanism

The source node with coordinates  $x$  can then determine the distance  $d(x, y_i)$  without requiring additional resolution or translation. Hence, it can also select the ‘nearest’ content server where a given content object is accessible by computing  $\min_i \{d(x, y_i)\}$ . Let  $k \in \mathbb{N}$  be the index of the coordinates selected by the source such that  $d(x, y_k) < d(x, y_i) \forall i \neq k$ . The reverse operation also applies: by receiving incoming packets that include in their header the coordinate associated to the source  $x$  (i.e., the locator associated to  $x$ ), the destination  $y_k$  can determine the distance  $d(y_k, x)$  without involving additional resolution or translation. This technique avoids thus requiring discovery of a reverse forwarding path from  $y_k$  back to the requester  $x$  as it is the case when the content object name is used to forward the request towards the content host.

It must be noted that the resulting delay improvement by relying on routing-inferred geometric coordinate system for cache selection is dependent on the quality of the fit of the embedding to underlying graph. This relates to the fact that obtained coordinates for network nodes must well reflect their position in the graph. A typical quality metric to estimate the way to which a geometric system relying on greedy forwarding fits the underlying topology, is the routing stretch. This refers to the proportion of the path length (usually hop count) when greedy routing compared to shortest path routing. Shortest path routing therefore has by definition stretch 1. The closer the average routing stretch of a geometric routing system is to 1, the better its coordinates can be assumed to reflect the underlying topology.

### C. Geometric caching strategies

Content distribution systems or media services are usually controlled by a single business entity. The resolution server introduced in the previous section is therefore doubling in many cases as dispatcher, load balancing or steering requests to those servers which reduce the delay towards the requesters. These dispatcher entities have a good overall view on the request matrix, and are therefore well-positioned to optimize the caching strategy among a fixed set of distributed servers. In this scenario, the dispatcher might rely on the routing system-inferred coordinates in order improve the caching strategy.

Consider a network topology modeled by an undirected graph  $G = (V, E)$  where, the set  $V$ , for which  $|V| = n$ , represents the finite set of nodes, the set  $E$ , for which  $|E| = m$ ,

represents the finite set of links. Every network topology has a subset of nodes  $S \subseteq V$  with  $|S| = k$  which also act as content servers. The content set  $C$  is a set of  $|C| = l$  content identifiers. Initially, every content identifier can be found on just one server, defined by the function  $IC : C \rightarrow S$ .

A request matrix<sup>2</sup>  $T : V \rightarrow P(C)$  defines which nodes request content ids. In order to optimize the caching towards additional servers for a given traffic matrix, the goal is to determine a caching function  $CACHE : C \rightarrow P(S)$  which defines for every content id on which servers it needs to be replicated. It is not the intention of this paper to solve this problem as a true optimization problem, however we consider caching heuristics exploiting coordinate knowledge of requesters as well as servers.

Based on a given request matrix  $T$ , determine for every content id, which is the set of the requesting nodes  $RN : C \times (V \rightarrow P(C)) \rightarrow P(V)$ . Now suppose for content id  $c_x$ , the set of requesting nodes is  $RN(c_x) = \{v_1^{c_x}, \dots, v_o^{c_x}\}$ . We might now rank servers considered for content replication according to a distance function  $DS : P(V) \times S \rightarrow \mathbb{R}$ . We consider two distances functions:

$$MAXD(V_{c_x}, S_y) = \max_{v \in V_{c_x}} (d(v, S_y))$$

and

$$SUMD(V_{c_x}, S_y) = \sum_{v \in V_{c_x}} (d(v, S_y))$$

for which  $V_{c_x}$  refers to the set of those nodes which request content id  $c_x$  according to the given traffic matrix. The first distance function reflects the maximal distance of any of the requesting nodes to the considered server, while the second refers to the sum of distances of the requesting nodes to the considered server.

The defined distance functions might be used to rank available servers with respect to increasing distance for caching the considered content id. When only selecting one caching server in addition to the existing server, the resulting ranking functions selects the server which is either closest to the geometric midpoint of the requesting nodes (MAXD), or which minimizes the resulting sum of distances towards the requesting nodes. Note that the latter does enable “outliers”, while the first does not.

## III. EXPERIMENTAL EVALUATION

We evaluate the potential gain of the proposed Content-Centric Geometric Routing architecture in a custom simulation environment based on scientific Python libraries `scipy` and `networkx`<sup>3</sup>. The simulation setup evaluates two aspects: the performance impact in terms of delay of having routing system-inferred coordinates on server selection, and the impact on improving the caching strategy itself. In addition, Section III-C validates the proposed protocol mechanics in an emulation setup on the virtual wall of iMinds.

<sup>2</sup> $P(C)$  refers to the power set of a set, i.e., all subsets of the set  $C$ .

<sup>3</sup>Cfr. websites: <http://www.scipy.org/> and <https://networkx.github.io/>

In order to compare the impact of using virtual coordinate systems to state-of-the-art mechanisms, we evaluate the proposed mechanism on the two largest network topologies from the sndl repository [10]: the ta2 and zib54 topologies. The first is a network topology provided by the network operator Telekom Austria, consisting of 65 nodes, the second is a slightly modified real-world instance from a telecom network operator of 54 nodes. Server nodes were fixed on 5 well-spread higher-degree nodes for both topologies. Content generation (content ids) is assumed to be uniformly spread over the servers. For the sake of the experiment, we evaluated for 1000 content ids. Content requests (we generated 100 requests per node) are also assumed to be uniformly spread over the nodes, and the distribution of the requested content ids follows a typical Zipf distribution ( $\alpha = 2$ ) following usual popularity phenomena. Each experiment was executed 10 times to obtain representative average values.

#### A. Improving server selection

The objective of the first experiment is to estimate the potential gain one might expect when routing-inferred coordinates are available in order to improve server selection from either a requester or dispatcher point of view (see Section II-B). The potential gain is measured as the potential reduction in delay with respect to randomly selecting a server. Delay is measured as a linear function of the length of the used path. Figure 3 depicts the distribution of resulting delay reduction compared to random cache selection for the two selected networks (ta2 on the left side, zib54 on the right), for the two coordinate variants: *tree\_coordinates* as introduced in the GTR description in Section II-A and *geo\_coordinates* representing the traditional approach where geographic locations are inferred from the corresponding IP addresses of requester and server nodes. The average stretch of the GTR is 1.096 for the ta2 topology, and 1.075 for the zib54 topology. This refers to the fact that on average, the routing path induced by greedy routing using these coordinates is at worst 9.6 and 7.5 percent longer compared to shortest path routing. A similar metric cannot be determined for the geographic coordinate system, as it is not guaranteed that a distance-decreasing path is available from any node (leading to loops).

In both cases the closest server out of 2 is chosen for every of the generated requests, and the resulting histogram of the delay improvement factor is drawn for both coordinate variants. As might be expected, there the delay improvement distribution is clearly in the advantage of using *tree\_coordinates*. For the ta2 topology, the average delay improvement of *tree\_coordinates* is 11.5 percent better than randomly selecting a server, while the use of IP-inferred *geo\_coordinates* is on average even 20 percent longer than randomly selecting one out of 2 servers for the available content. For the zib54 topology, the corresponding average improvement factors are 14 percent for *tree\_coordinates* and -18 percent for *geo\_coordinates*. The bad results for IP-inferred geographic coordinates confirm the fact that geolocators are usually incorrectly reflecting the underlying topology.

One might wonder how this trend evolves when more than 2 caches are available for requested content. In Figure 4 this relationship is further evaluated up to 5 caches per content identifier. As might be expected, when the underlying coordinate system does not well reflect the position of servers within the topology, as is the case for IP-inferred *geo\_coordinates*, increasing the number of caches does not necessarily reduce the resulting delay (minor delay factor variations might be noticed for *geo\_coordinates* for different number of available caches). In contrary, when using routing-inferred coordinates, such as *tree\_coordinates*, having more caches available does effectively reduce the average delay towards the requesters resulting from choosing the closest server. For the ta2 topology, having 5 caches available results into an additional 10 percent in delay, compared to only 2 caches, while in the zib54 topology, the additional gain is between 5 and 10 percent.

#### B. Improved caching strategies

In a second experiment, the goal is to evaluate if actively selecting servers for caching content with respect to given request matrices could even result into larger benefits with respect to delay reduction towards requesters. For this purpose, we evaluate four caching strategies: selecting  $x$  caches to minimize the SUMD or MAXD function for IP-inferred geographic coordinates, i.e. i) *geo\_coordinate\_max* and ii) *geo\_coordinate\_sum*, and selecting  $x$  caches to minimize SUMD or MAXD function for routing-inferred GTR coordinates, iii) *tree\_coordinate\_max* and iv) *tree\_coordinate\_sum*.

This results into 4 caching strategies which are benchmarked against randomly selecting the same number of caches for content replication among servers. Figure 5 again reflects the resulting delay factor improvement of these strategies compared to the random caching strategy for the ta2 and zib54 topologies. As might be observed from the graph, actively caching seems to further magnify the potential gain of using routing-inferred coordinate systems for guiding the caching strategy. In the depicted graph, the delay factor improvement can be 28 percent when comparing to randomly selecting servers. However, adding more caches does not further improve the performance. This might be explained because of the fact that adding more caches by selecting those servers which is the second, third, etc. closest to the requesters is not a good strategy once the first, second, etc. closest server (w.r.t. the defined ranking of the distances functions defined in Section III-B) have already been selected for caching the content. Future work might involve optimizing more complementary server selections compared to the proposed strategy.

The impact of the proposed ranking functions SUMD and MAXD seems to be minimal in the evaluated scenarios. For strategies based on *geo\_coordinates*, the MAXD function seems to perform slightly better, while for *tree\_coordinates*, the opposite seems to be the case. Although further experimentation would be needed, the SUMD functions seems to be the function which might be performing slightly better in case an adequate (low stretch) embedding is available as it

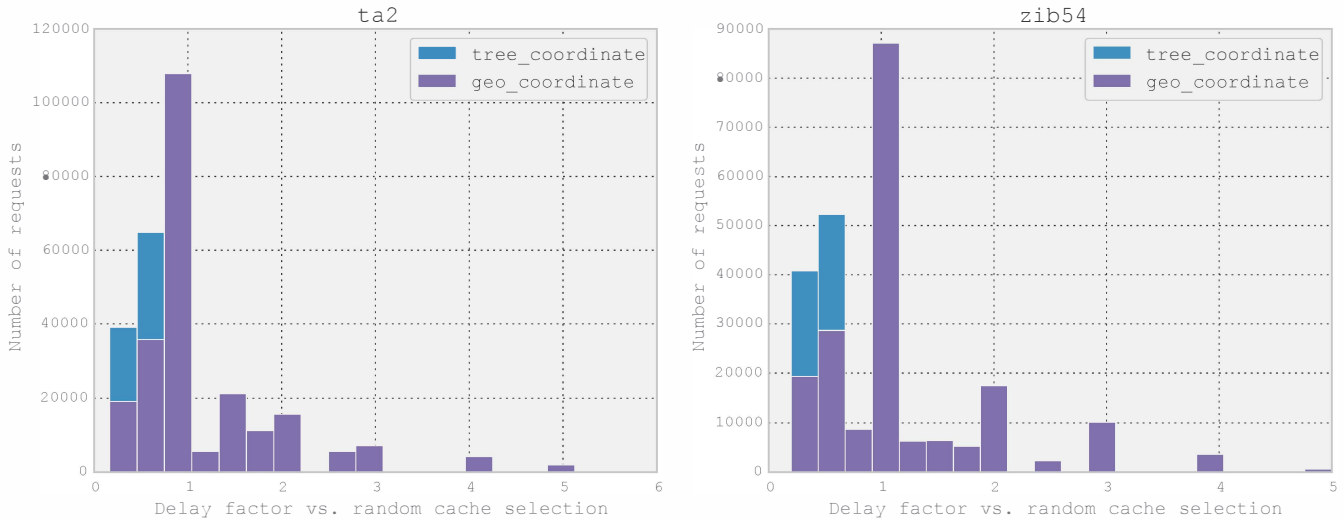


Fig. 3. Histogram of delay improvement compared to random cache selection

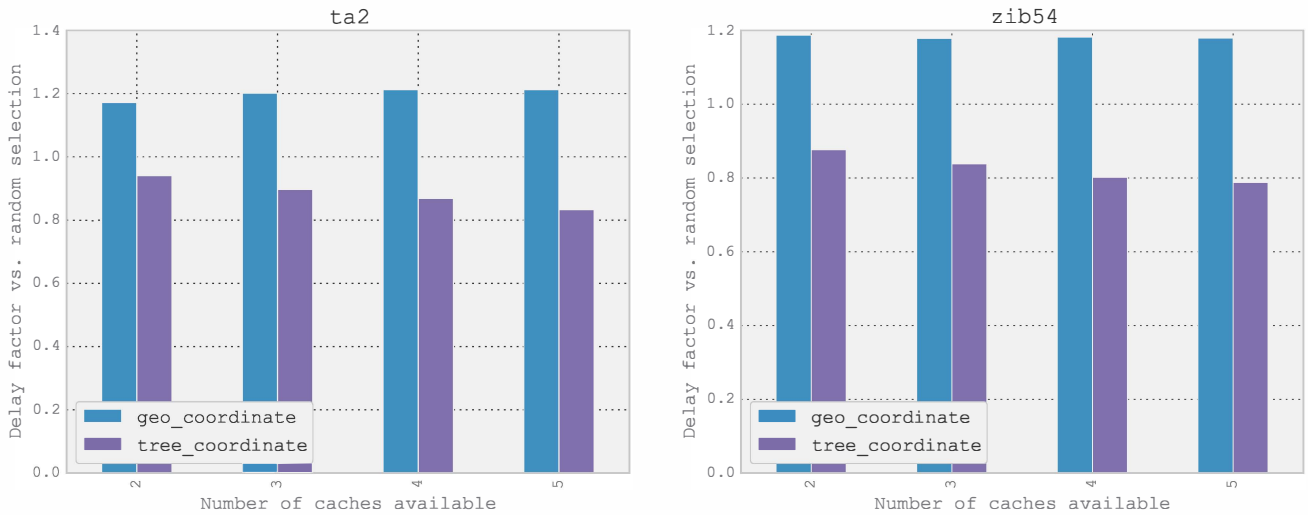


Fig. 4. Average delay improvement compared to random cache selection when multiple caches are available

almost directly minimizes the average delay between servers and requesters while enabling a few outliers (in terms of distance of requesting node to server).

### C. Protocol validation

The feasibility of the proposed approach for server selection as well as for applying active caching strategies has been validated in an emulation setup on the virtual wall of ilab.t of iMinds<sup>4</sup>. This environment consists of a rack of 100 servers interconnected by a non-blocking switch. Together with the Emulab platform, this enables emulation experiments on any logical topology between the servers. The emulation environment makes use of the following entities:

- *Content server*: Apache HTTP server
- *Request handler*: HTTP server and HTTP forwarder

- *id2loc*: script mapping identifier to coordinate
- *requester/client*: VLC player
- *DNS service*: DNS Bind9 implementation

Figure 6 depicts the interaction between the described entities. Content servers register their contents to the Request handlers. The clients send requests to the Request handler which replies with a list of servers hosting the requested contents. Based on the locators, the client selects the closest server and sends a request for the content. Edge routers perform a lookup in their local forwarding table to determine the next-hop along the trajectory towards the content locator.

The results within this environment follow similar trends as those indicated in prior sections, and confirm the feasibility of the proposed approach. A demo of this setup has been presented at CNERT 2014. More results and detailed documentation on the proposed setup can be found in [11].

<sup>4</sup><http://ilab.t.iminds.be/>



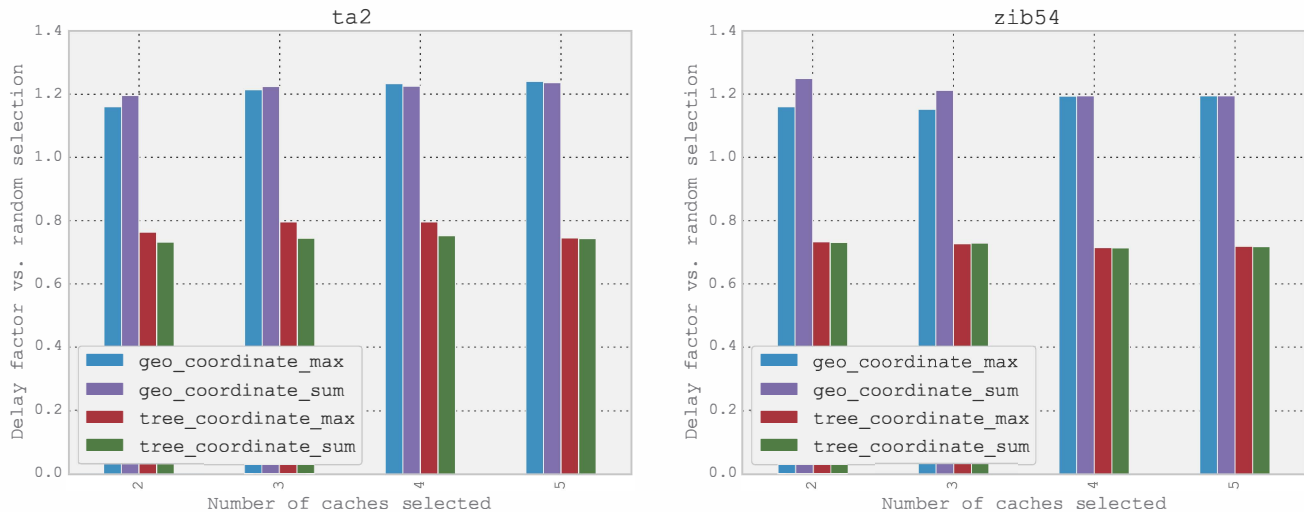


Fig. 5. Average delay improvement compared to random cache selection when multiple caches can be actively created

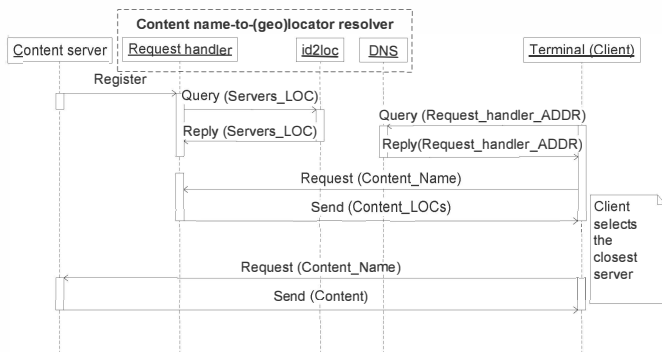


Fig. 6. Sequence diagram of component interaction

#### IV. CONCLUSION AND FUTURE WORK

This paper presented a novel architecture for Content-Centric Networks supported by a geometric routing system. The underlying idea is to improve CCNs by incorporating coordinate systems based on routing information, rather than on active (RTT) measurements or IP-inferred geographic information. The added value of such a scheme is demonstrated in a simulation setup. Experiments illustrated that average delay can be reduced with more than 20 percent compared to random caching strategies. In addition, it was shown that relying on IP-inferred geographic coordinates does not necessarily improve the resulting average delay. In fact, average delay figures based on geographic coordinates were even worse than just randomly selecting caches. Next to the simulation experiments, an emulation setup confirmed the feasibility of the proposed approach using available components.

Future work might consist of further evaluating the impact of using different coordinate systems on different topologies and the design of absolute optimization formulations in order to optimize the caching strategy for given request matrices, or heuristics to optimize caching in in online settings.

#### ACKNOWLEDGMENT

This work is partly funded by the European Commission through the EULER project (Grant 258307), part of the Future Internet Research and Experimentation (FIRE) objective of the Seventh Framework Programme (FP7). This project was partly funded by the UGent BOF/GOA project "Autonomic Networked Multimedia Systems".

#### REFERENCES

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.
- [2] R. Kleinberg, "Geographic routing using hyperbolic space," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 1902–1909.
- [3] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.
- [4] C. Westphal and G. Pei, "Scalable routing via greedy embedding," in *INFOCOM 2009, IEEE*, 2009, pp. 2826–2830.
- [5] D. Eppstein and M. Goodrich, "Succinct greedy geometric routing using hyperbolic geometry," *IEEE Transactions on Computers*, vol. 60, no. 11, pp. 1571–1580, 2011.
- [6] F. Papadopoulos, D. Krioukov, M. Bogua, and A. Vahdat, "Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces," in *INFOCOM 2010, IEEE*, 2010, pp. 1–9.
- [7] S. Sahhaf, W. Tavernier, D. Colle, M. Pickavet, and P. Demeester, "Single failure resiliency in greedy routing," in *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*. IEEE, 2013, pp. 306–313.
- [8] A. Cvetkovski and M. Crovella, "On the choice of a spanning tree for greedy embedding of network graphs," *Networking Science*, vol. 3, no. 1–4, pp. 2–12, 2013.
- [9] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 15–26.
- [10] S. Orłowski, R. Wessälly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0: survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.
- [11] S. Sahhaf, D. Papadimitriou, W. Tavernier, D. Colle, and M. Pickavet, "Experimentation of geometric information routing on content locators," in *International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT)*, 2014.