Geometric Information Routing

Dimitri Papadimitriou
Alcatel-Lucent Bell
Antwerp, Belgium
Email: dimitri.papadimitriou@alcatel-lucent.com

Didier Colle, Pieter Audenaert, Piet Demeester
Ghent University - iMinds
Gent, Belgium
Email:{didier.colle,pieter.audenaert,p.demeester@ugent.be}

***Abstract*— In response to the increasing traffic volume in the Internet for applications such as (mobile) video and cloud computing, various proprietary technologies enabling content distribution have been developed that rely on caching and replication. Being deployed in silos, it is not possible to uniquely and securely identify named information independently of the distribution channel; moreover, these different content distribution technologies are typically implemented as an overlay, leading to unnecessary inefficiency. By introducing uniquely named data and name-based data access, Information-Centric Networking (ICN) enables data to become independent from their network location, application, storage support but also means of content exchanges enabling in turn in-network caching and replication. However, content name spaces have not been designed to sustain forwarding performance and forwarders scaling contrary to IP addresses which can be efficiently aggregated, summarized and translated. Consequently, alternatives such as name-based routing, which aim at better accommodating information/ content routing in the Internet, would also become the scaling and performance bottleneck. To address these problems, this paper proposes a third alternative: geometric information routing on universal content locators. This technique operates by associating to content identifiers (names) a content locator taken out of a geometric coordinate space from which a routing path (more precisely, a geodesic) can be derived without requiring non-local information. Upon querying specific content multiple locators can be received enabling the receiver to select the (geometrically) closest locator. Since it is based on local information only, routing on such locator space is less memory consuming than non-local information dependent routing. We analyze the performance (in terms of memory space required to locally store routing states and the resulting routing path stretch) and compare them against path-vector routing.**

Keywords—information networks; geometric routing; naming; addressing; content locator

I. INTRODUCTION

The increase in Internet traffic volume for applications such as (mobile) video and cloud computing has led to various technologies enabling content distribution that rely on caching and replication. Being proprietary and deployed in silos, these content distribution technologies do not allow uniquely and securely identifying named information independently of the distribution channel. Moreover, these technologies are usually implemented as an overlay, leading to needless inefficiency. Information-centric networking (ICN) a.k.a. content-centric networking (CCN) [1], by proposing uniquely named data and name-based data access, enables data to become independent from their network location, application, storage support but also their transport enabling in turn in-network caching and replication. While several attempts and trials can be cited over the last 20 years, many approaches have been explored since then such as DONA [2], NDN [3], PURSUIT [4], COMET [5], CONVERGENCE [6] to cite just a few of them, as recently surveyed in [7]. However, content name spaces have not been designed to sustain forwarding performance and forwarders scaling contrary to IP addresses which can be efficiently aggregated, summarized and translated. Indeed, following the seminal work of P.Baran [8] in the early 60's, information is segmented into packets (elementary data unit of the networking functionality) before being transmitted. Without anticipating the main consequences of such decomposition, P.Baran laid the distinction between "names" and "addresses" as stated by J.Shoch (1982): *The 'name' of a resource indicates what we seek, an 'address' indicates where it is, and a 'route' tells us how to get there* [9]. As a result of this decomposition, the functionality of the network layer has been since more than 30 years confined to packet forwarding (connectivity function) whilst it is mainly used nowadays for information exchanges (distribution function). Their unification as intrinsic network functionality can now be justified in line with the functional placement principle.

The fundamental architectural problem this (general) concept triggers can be phrased as follows. On the one hand, name spaces haven't been designed to sustain forwarding performance and forwarders scaling contrary to IP addresses which can be aggregated, summarized and translated. On the other hand, all forwarding-level processing has been designed assuming data structures were IP addresses, in particular IPv4. It is noticeable to observe behind this argument the reason why transition to IPv6 that is incompatible with IPv4 took almost 20 years to get deployed in regions where there was no IP address shortage. Consequently, now that names become the bottom-line structure of the information exchange process, name-based routing becomes also the scaling and performance bottleneck. The two major alternatives currently under investigation consider either routing on content names (with IP addresses keeping network locator semantic) or introducing one more level of indirection (overlay) by extending IP addresses semantic as location name from where the information can be retrieved. The first alternative suffers from limited scaling (in terms of memory space). Moreover, the adoption of a new content name space is challenging but the maintenance of a hierarchical tree-based structure is even more difficult to reach. The second alternative raises similar issues as any network overlay following the well-known aphorism of D.Wheeler [10] which coins the problem of too many levels of indirections[1].

---

[1] D.Wheeler complete statement quotes *There is no problem in computer science that can't be solved by adding another level of indirection to it... but that usually will create another problem.*

Hence, with both alternatives, the fundamental question remains: location, identification, and routing refers to distinct objects (address vs. identifier vs. route) associated to distinct functions which can't be derived from each other using local knowledge. Moreover, the higher the level of identifiers on which the routing decision is performed, the higher the memory cost; on the other hand, lowering the level by providing additional resolution processes increases the communication cost and latency. To address these problems, we propose a third alternative: information routing on universal content locators taken out of geometric coordinates. This technique operates by associating to content identifiers (names) a content locator taken out of a geometric space from which a routing path (more precisely, a geodesic path) can be derived without requiring non-local information. Upon querying specific content object, multiple locators can be received enabling the requestor to select the (geometrically) closest content locator. The requestor receives all needed information to select the location to access the corresponding content object being files or other digital content. Moreover, since it is based on local information only, routing on such locator space is less memory consuming than non-local information dependent routing, including name-based routing.

The remainder of this paper is structured as follows. We first document in Section II the preliminaries used throughout this paper. Prior and related work are documented in Section III where after modeling the architecture of the proposed model, we compare it to existing content naming and routing systems, including name-based routing and overlays. In Section IV, we provide a performance analysis in terms of memory space required to locally store routing states and the resulting routing paths stretch and compare these performance against path-vector routing. Finally, Section V concludes this paper together with future perspectives and research challenges.

II. PRELIMINARIES

Consider a network topology modeled by an undirected graph $G = (V, E)$ where, the vertex set V ($|V| = n$) represents the finite set of nodes and the edge set E ($|E| = m$) represents the finite set of links. For $u, v \in V$, the loop-free path $p(u, v)$ from vertex u to v is defined as the finite sequence $[x_0(= u), x_1, \ldots, x_{i-1}, x_i, \ldots, x_p(= v)]$ such that the vertex x_{i-1} is adjacent to x_i, $\forall (x_{i-1}, x_i)_{i=1,\ldots,p} \in E$. The *length* $\ell(u, v)$ of the path $p(u, v)$ is defined as the finite number of edges a given path $p(u, v)$ traverses from vertex u to v. The *average path length* is defined as the average of the shortest path length, averaged over all pairs of vertices. Given a distance metric d, the *distance* $d(u, v)$ between two vertices $u, v \in V$ denotes the length of a shortest path $p(u, v)$ from vertex u to v. The *diameter* $\Delta(G)$ of the graph G is defined by the maximum distance between any two vertices $u, v \in V$, i.e., $\delta(G) = max_{u,v}\{d(u, v) | u, v \in V\}$.

Let D denote the set of reachable network destinations (n |D|). The *routing function* denoted by f determines $\forall\, t \in D$ and $\forall\, u \in V$ the adjacent node (next-hop) w of u, $(u, w) \in E$, along a given trajectory from node u to destination t (reachable via node v). This trajectory is determined by the *routing algorithm* which computes $\forall\, t \in D$ and for each node $u \in V$, a (routing) path $p(u, w, \ldots, v{:}\, t)$ or the minimum neighbor's distance to destination t, denoted $d(w, v{:}\, t)$. The application of the routing function f to the result of this computation enables any node $u \in V$ to forward its incoming traffic directed to destination t along a loop-free path to that destination. We refer to a distributed routing function f_u when the function f is executed locally at each node $u \in V$ and independently of all other nodes $v \in V \setminus \{u\}$. The associated distributed routing algorithm performing at each node $u \in V$, computes for each destination t a routing path $p(u, w, \ldots, v{:}\, t)$ or the minimum neighbor's distance $d(w, v{:}\, t)$ to t. The algorithm output is then used by the routing function f_u so that node u can forward its incoming traffic directed to destination t along a loop-free to that destination. Distributed path computation algorithms are assumed to perform asynchronously; moreover, associated decisions at each node are taken independently of other nodes.

In the information routing context, it is also crucial to distinguish between the location and the identification functions and their associated information units, i.e., locator and identifier. The *locator* function identifies a location in an internetwork. Locators identify "where" the node is positioned in the network. Nodes and endpoints are assigned locators (LOC) taken from a value space being topology-dependent labels or topology-dependent addresses (when these are IP addresses assigned by ISP they are referred to as Provider Allocated (PA) addresses) or coordinates (as determined by a geometric space). A node is assigned only one locator. An endpoint can be assigned more than one locator so that a locator might appear in more than one location of an internetwork. The *identification* function identifies unambiguously a node independently of its actual location. The identifier (ID) value space verifies thus the topology-independence property. The identifier value space spans from any unstructured character string to structured names until topologically independent addresses (when these are IP addresses they are referred to as Provider Independent (PI) addresses).

III. CONTENT LOCATORS MODEL

For the sake of clarity, we introduce our proposed model starting from the two main paradigms underlying most (if not all) known content/information routing systems. Their motivation primarily results from the design (and underlying principles) of the Internet naming and addressing model. The description provided here below relies on the distinction between location and identification function even if this fundamental design principle is weakly supported in the Internet. The main reason stems because a single value space, the IP addressing space, is used simultaneously as locator and identifier space (often referred to as the semantic overload of IP addresses). On the one hand, the forwarding function has been optimized to operate on IP addresses. On the other hand, content name spaces have not been designed to sustain forwarding performance and forwarders scaling. Therefore, any departure from this model leads necessarily to reconsider or extend either intermediate nodes functionality or the current naming and addressing spaces (or even both).

A. *Main Models*

Two seed models namely, the overlay model and the name-based routing model, can be identified as the root of most currently proposed approaches to information-centric/content routing (see Fig.1). We deliberately avoid the multicast /rendez-vous model also referred to as the publish-subscribe model [4]. The latter corresponds indeed to the overlay model with the addition of routing functionality to improve content delivery performance in terms of capacity consumption.

1) Overlay model

This model considers an additional level of indirection between content names and host identifiers. As such this model complies with the Internet model where a host identifier space is inserted in between existing content names and network locators. When no distinction is made between host identifiers and network locators, content names get simply resolved into IP addresses. These IP addresses are often assigned independently of the topology which in turn prevents renumbering in case of multi-homing (about 25% of the sites connected to the Internet). When such distinction is made, the model complies with the (generic) architecture underling the Host Identity Protocol (HIP) [11] which also requires an additional level of indirection. However, as of today there is no known identifier space outside IP fulfilling this role. In other terms, the distinction results in combining PI addresses (acting as host identifier) on top of PA addresses (acting as locator).

This apparent simplicity constitutes also the main weakness of this model. This architecture would indeed exacerbate the well-known problem generated by Provider Independent (PI) addresses. PI addresses are not topologically aggregable. They are allocated independently of the topology without taking into account effects on the (global) Internet routing system, making in turn Classless Inter-Domain Routing (CIDR) ineffective in handling address scaling. Routing on PI addresses implies indeed to store and process non-aggregable routing table entries in the Internet routing system. On the other hand, the cost incurred by these additional routing entries (in terms of memory space and processing capacity) is directly supported by the global routing system rather than the owner of these addresses. Assume for instance that a Provider Independent IP address would be allocated to each content name domain (which the coarsest level of granularity one can reasonably expect) the number of active routing entries would increase from about 5.10^5 to $2.5\ 10^8$ [12]. Moreover, the resulting increase of the routing table sizes (in terms of number of entries) and associated processing would worsen over time as the number of domain increases also by 10 to 15% per year (Verisign report, April 2013). In summary, all elements are currently available to build this model; however, the increase in routers memory and processing cost (to ensure forwarding performance) outweigh the gain in capacity and transit cost. Finally, it is worth remembering that evolution by overlay results from a particular application of the modularization/ Dijkstra principle [13] to the design of communication systems and stacks. Observe also that a similar reasoning could explain (at least partially) why multicast routing has not been widely deployed on the Internet.

2) Name-based routing model

The name-based routing model refers in this paper to the process by which the routing function f_u locates a content object based on its name which is initially provided by a requestor. The corresponding process can be decomposed into three main sub-functions: an (optional) name resolution, a discovery, and a delivery function (further details can be found in [14]). The name resolution function translates the name of the requested content object into its network locator. The discovery function routes the request to the content object based on its name or network locator. The delivery function routes the content object back to the requestor.

Depending on how these functions are combined, name-based routing schemes leads to two main alternatives. The first alternative omits the first name resolution function. The name of the content object is directly used to route the request towards the host of the content object. Therefore, the routing information corresponding to each content object has to be maintained in the routing table. Since the number of content objects is very large (in between 10^{15} and 10^{22} [14]), the size of the routing tables becomes a key concern, as it can be proportional to the number of content object unless an aggregation mechanism is introduced. Assume for instance that routing tables would include one entry per top-level domain, name-based routing table would have to hold 2.10^8 routes as reported in [13]. With current naming system further summarization is almost impossible to define. Hierarchical renaming would seriously affect compatibility and introduce serious maintenance problems. On the other hand, this alternative reduces the overall latency by omitting the resolution process. An alternative often considered relies on predictive cache replacement algorithm to mitigate latency effects. Moreover, the delivery function needs another identifier (ID) of either host or location to forward the requested content object back to the requestor. Otherwise, an additional routing mechanism has to be introduced to enable forwarding of the actual content object back to the requestor. The second alternative (combination of alternatives depicted in Fig.1) relies on the name resolution function to translate the name of the requested content object into its locator. Then, the discovery function is carried out based on the locator (that can take an IP address as value). Consequently, the delivery function can be implemented similarly to conventional IP routing. As the locator of the requestor is included in the request message, the requested content object is delivered to the requestor based on the locator. The main challenges with this alternative include mainly the design of a scalable resolution system which provides fast lookup (mapping the name of the content object to its locators) and fast update (as the location of data object is expected to change frequently). Once can also observe that finding best tradeoff by means of a hybrid model may be attractive at first glance but combining their respective advantages also comes with additional disadvantages such as coordination inside and between domains.

Hence, the name-based routing approaches emphasize the major and well-known tradeoff experienced when designing routing systems: the first alternative exacerbates the main drawback of the push model, i.e., storage, and the second the main drawback of the pull model, i.e., latency. The survey [7] produced under the auspices of the Information-Centric Networking Research Group (ICNRG) and the analysis of its associated challenges [14] demonstrate that all name-based routing approaches share common scaling problem.

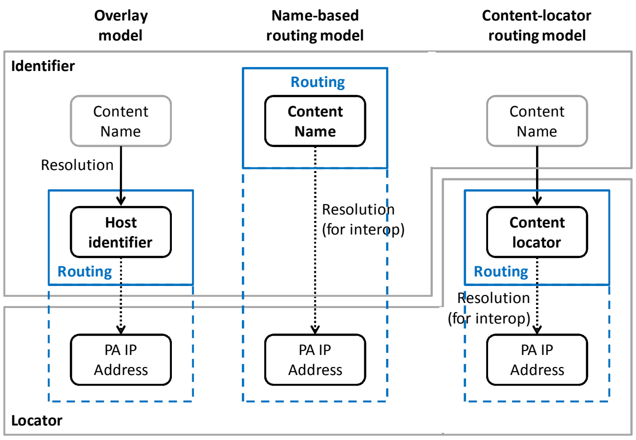

Fig.1. Content naming and addressing models

B. *Proposed Model*

The proposed alternative model consists in assigning content locators to content identifiers or content names (being an addressable information unit). The principle underlying this alternative is relatively simple, perform information routing decision on locators avoiding name-to-locator resolution by intermediate nodes.

Consider the content locator space denoted by X. Each element $x \in X$ is represented by its globally unique geometric coordinates (which correspond to the content locator value space). Let d denote the associated distance metric such that the tuple (X, d) defines a metric space. By obtaining the coordinates associated to the destination y (i.e., the locator associated to y), the source node with coordinates x can determine the distance $d(x, y)$ without requiring additional resolution or translation. Hence, it can also select the "nearest" content server where a given content object is accessible. 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 can determine the distance $d(y, x)$ without requiring additional resolution or translation.

The relationship between content name (identifier) and content locator is $M{:}N$. In other terms, a given content object can be assigned multiple locators (it can be retrieved from multiple servers) and a given locator can host multiple content (a server can host many content object). Observe that no intermediate entity or node is required to maintain the entire set comprised as part of $M{:}N$ relationships in the end-to-end communication process, i.e., the resolution process can be entirely distributed across the whole network. Content locators are defined as coordinates that are automatically allocated with respect to the content network topology. Hence, the problem of topology-dependent address allocation does not exist for this naming and addressing scheme. No arbitrary rules or allocation policy are actually required neither additional registration processes. Moreover, this scheme is backward compatible as it does not require or imply re-naming of content objects or domains. It is also forward compatible as it does not prevent the definition and usage of new content naming spaces in the future. Nevertheless, such scheme still mandates the allocation of network-wide unique coordinates, this necessary condition is ensured by the use of a geometric space. This process is further documented in Section IV.

C. *Theory of Operation*

The model proposed in Section III.B relies on the following functions (as depicted in Fig.2):

1. *Registration*: the entity hosting the content object registers its corresponding locator (denoted by `<locator>`), i.e., its coordinates $x = (x_1, x_2, \dots, x_k)$) together with the content object name (denoted by `<name>`) to a name resolution server. The name resolution server maintains records using the syntactic form `{<name>,<locator_set>,<ttl>}` where `<locator_set> := <locator> | <locator> [<locator_set>]` and `<ttl>` corresponds to the usual time to live value associated to the name server record. The content hosting entity can be a server or an intermediate node/cache. The mechanism by which the hosting entity determines its actual locator (coordinate) is documented in Section IV.
2. *Resolution*: before requesting a given content object, the requestor sends to a name resolution server a name resolution message which includes the name associated to the requested content object together with a set of optional attributes. This message takes the form `<name> [<attributes>]`. The actual address of the name resolution server is assumed to be known to the requestor as IP addresses of Domain Name System (DNS) server are currently communicated dynamically to IP hosts (or via manual configuration). The name resolution server replies by providing the content object name together with a list of content locators. The server reply message takes the form `<name> <locator_set> <ttl>`. In case the resolution request includes optional attributes, this list is then limited to the set of locators meeting these attributes.
3. *Delivery*: upon reception of the response message by the name resolution server, the requestor can then initiate the exchange with the entity hosting the content object itself. Remember also that no additional resolution mechanism is required for forwarding back the content object to the requestor. Indeed, incoming packets include in their header the coordinate associated to the requestor; thus, the requested content object can be forwarded to the requestor without requiring additional resolution or translation.

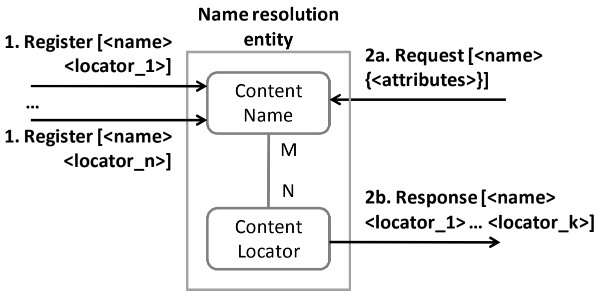

Fig.2. Content locator Registration and Resolution

Up to now, the reader could question the actual value of the proposed model; in particular, since it also requires resolving content names to content locators. The salient feature of this model comes from the property of coordinate-based content locators: these coordinates can be used by the distributed routing function to perform geometric routing decisions. Geometric routing operates by assigning to each node virtual coordinates in a metric space (X, d) that are used as locators to perform point-to-point routing decisions in this space. Thus, content locators would substitute to network locators (stricto sensu, the routing function acts on locators thus this fundamental principle remains unchanged) but they can also be used in combination with other network locator spaces, e.g., IP addresses. The situation where content locators would require resolution to network locators ensures interoperability when messages are transmitted across IP-only forwarding networks. Moreover, there is no distinction between "server" and "cache" locators, i.e., if an intermediate node determines it has to keep a local copy of a content object, it can decide to apply the registration step described here above. Subsequently, a requestor could receive the content locator associated to an intermediate node. This paper does not further elaborate on the issue of cache placement or storage capacity allocation on intermediate nodes. Indeed, this problem refers to operational research considerations more than networking research or protocol architecture (at least once the protocol functionality enables temporary storage at and retrieval from intermediate nodes). Further results on cache placement strategy can be found in the recent comparative performance study [15] which contrasts pervasive caching and nearest-replica routing against edge caching.

D. Main Implications

Before detailing the design of the geometric routing scheme (and possible alternatives), it is instructive to underline the main implications and consequences of the model documented in Section III.C and III.D.

1) Name resolution system

The proposed model does not fundamentally change the current working principles of the name resolution system. RFC1876 [16] for instance describes a mechanism to allow the Domain Name System (DNS) to carry location information about hosts, networks, and subnets. Even if of experimental nature, this specification provides a tangible proof that coordinate-based locators remain compatible with current DNS principles and its architecture. In particular, it defines the format of a new Resource Record (RR) for the DNS which allows encoding of coordinates in the form of longitude and latitude. These differ from geometric coordinates but the underlying mechanism used by the DNS to translate domain names into location information (search by name algorithm) remains unchanged. It also indicates that a similar format could be defined to encode coordinates of the form $x = (x_1, x_2)$).

2) Communication stack (hosts/terminals)

The main implication on host terminals comes from the association of content locators (which determine the content server location) to content names. This association triggers the fundamental question if transport-level exchanges should be determined by individual content requests or delimited by the end-to-end relationship established between terminals (i.e., between the content requestor and the content server). In the former case, one has to consider that each exchange of a content object corresponds to a dedicated transport-level connection. In the latter case (shared) connections can be used to exchange multiple content objects. The underlying design choice has different impact. In the former case, transport layer segments could be directly encapsulated into network layer packets. On the other hand, in case of shared connection (like TCP/IP), one would have to encapsulate the transport layer segments into, e.g., an IP packet and then encapsulate that packet into another protocol data unit to forward it towards its destination. The second alternative rises though the question of bringing the model back to an overlay approach as the IP address to coordinate association would have to be known prior to the establishment of the connection (even if IP addressing would only be processed by terminals being hosts or servers).

There is no definitive answer as of which model would or could be adopted in the longer term though dual-stacks are to be expected during transition period. However, connection by content name provides certainly the potential to offer higher granularity of flow control compared to current TCP. In addition, connection by content name enables decoupling transport addresses from network-layer addresses (though some protocol designers consider this coupling as a feature). This technique which involves similar mechanisms to the name-oriented sockets interface [17] aims at solving the more general challenge of providing a name-based network-API [18]. The main drawback of this approach stems from the need to introduce new locator family (geometric coordinates) as access point to the transport layer. The introduction of this new family required at both end-points (symmetric) leads to a fundamental question which goes well beyond the scope of this paper: could there be an additional end-to-end locator space next to IP address spaces ?

3) Routing and Forwarding system

The necessary condition to be met is to augment the routing (and forwarding) system to effectively cope with new locator (or address) family. We refer to term augmentation because any router software support neighbor tables and adding neighbor's coordinate attribute does not lead to a daunting development effort. The most significant change results from moving current adaptive routing protocol functionality of routing path computation (per destination) to coordinate computation whereas the forwarding process requires (stateless) computation of next-hop neighbor on a per-packet basis. Several experiments have shown that the forwarding level processing is achievable. However, designing the routing protocol to enable the computation and assignment of local coordinates leads to a much less trivial task. The concept is actually not new and already used in satellite networks and other confined networks (such as sensor networks). The real challenge comes from the scale of the Internet and the underlying computational procedures to assign nodes coordinates when timely knowledge of the complete topology is not locally available at each node.

IV. GEOMETRIC ROUTING ON CONTENT LOCATORS

A. Greedy Geometric Routing

Greedy geometric routing performs by assigning to each content object (virtual) coordinates in a metric space (X, d);

these coordinates correspond to the "location" of the nodes hosting or temporarily caching content. Assuming that intermediate (forwarding) nodes are also included in that space, each node $u \in V$ of the graph $G = (V, E)$ where, V represents the set of content forwarding and hosting nodes, knows its own position (coordinate) and the position of its neighbors $N(u) = \{v \mid (u, v) \in E\}$. These coordinates are then used as locators to perform point-to-point routing by selecting the neighbor that is closest to the destination. We will see hereafter that this knowledge becomes the most critical part of this routing paradigm (as it determines its computational complexity and communication cost, i.e., the number of messages exchanged). The greedy geometric routing process can thus be decomposed into two main sub-functions: the coordinate (distributed) computation and the routing function performing on these coordinates.

1) Routing function. Given the distance function $d_g: V \times V \rightarrow \mathbb{R}^+$, each node performs the following computation (and associated decision for next-hop selection). For each destination $t \in D$, a node $u \in V$ routes incoming messages (directed to destination t) to its neighbor $v \in N(u)$ if $d_g(v, t) = min_{x \in N(u)} d_g(x, t)$. When $d_g(u, t) > d_g(v, t)$ at each node along the routing path from the source to the destination t, the distance d_g decreases monotonically along this path. Consequently, the resulting distance decreasing routing path is loop-free. This routing process is referred to as greedy since the local computation of the next-hop is performed for each incoming message without maintaining any routing state information per destination. In other terms, the routing table at each node corresponds to a neighbor (or adjacency) table. The salient feature of geometric routing is that it builds a set of local routing entries whose memory size is proportional to the degree of each node (if we exclude the memory mobilized for storing the results of the operations for coordinate assignment). This process comes at the expense of coordinate computation in a metric space allowing to derive the distance between any two vertices from their coordinates.

2) Distributed coordinate computation: introduced by R.Kleinberg in 2005, greedy embeddings of connected finite graphs in the hyperbolic plane constitute a crucial step in the search of provable means to overcome known limitations of geometric routing on undirected graph $G = (V, E)$ embedded in the Euclidean space [21]. R.Kleinberg proved in [20] that every finite, connected, and undirected graph has a greedy embedding (in polynomial time) in the two-dimensional hyperbolic metric space $\mathcal{H}^2$. An embedding of an undirected graph $G = (V, E)$ into a metric space X equipped with a distance function d is defined as a one-to-one mapping function μ of the nodes of G, $V(G)$, to the points of X ($V(G) \rightarrow X$). The notion of greedy embedding introduced in [21] adds the distance decreasing path property, i.e.,, $\forall\, u, t \in V(G)$, $u \neq t$, $\exists\, v \in V(G)$ with $(u, v) \in E(G)$ such that $d(\mu(u), \mu(t)) > d(\mu(v), \mu(t))$. In other terms, greedy embeddings ensures that the distance decreasing path towards any destination can always be determined by local routing decisions (if such path exists). It is instructive to compare the situation with the Euclidean space. In the latter, two-dimensions are insufficient to ensure the distance decreasing path property yielding routing paths along which messages are trapped at local minima; thus, preventing messages from reaching the destination. On the other hand, $log(n)$ dimensions are required (at least) to ensure that the greedy condition holds; however, an $O(log\, n)$-dimensional space is still too high for most applications.

Greedy embeddings suffer from three main drawbacks: 1) The distance decreasing path property is a necessary but insufficient condition to yield routing paths whose stretch[2] is polylogarithmic in the number of nodes n. Many studies have been conducted to determine the routing path stretch increase when applying geometric routing on greedily embedded graphs in particular scale-free/Internet-like graphs. These results confirm the initial results of [20] though several improvements can be provided to mitigate the routing path stretch increase; 2) Algorithms for producing greedy embeddings yield vertex coordinates representation requiring $O(n\, log(n))$ bits. Thus, the memory space required to store the resulting routing table entries corresponds to the memory space requirement imposed by "conventional" shortest path routing algorithms (such as distance- or path-vector). As reported also in [22], greedy embeddings to be useful for geometric routing should be succinct, i.e., vertex coordinates shall have a number of bits polylogarithmic in the number of nodes n. Authors in [22] show the existence of a polynomial time algorithm for greedy embedding of finite and connected graphs in the hyperbolic plane $\mathcal{H}^2$. This constructive algorithm produces vertex coordinates that can be represented using $O(log(n))$ bits and allows computation of the standard hyperbolic distance between any two vertices from their coordinates; 3) Graph dynamics due to topology changes (as links may experience failures or can be added or removed and nodes can join or leave the network) may invalidate the greedy embedding property and thus the success rate of greedy geometric routing. In these conditions, when applying the greedy embedding algorithm of [20], $O(n)$ operations are required to globally reconstruct the embedding and retain the greedy property. To address this problem, the seminal work of R.Kleinberg has been recently extended by [23] which suggests an incremental greedy embedding algorithm of graphs in the hyperbolic plane (without disturbing the global embedding). However, this generalization of greedy geometric routing requires Gravity–Pressure (GP) routing which shows several drawbacks as reported in [19] that are particularly problematic when intermediate routers temporally store content objects. Moreover, the routing path stretch is still not guaranteed with the hyperbolic coordinates generated by the online embedding algorithm. Indeed, the resulting stretch can be very large, as in the worst case a message may need to visit a large portion of the network nodes before finding its actual destination!

These arguments lead us to consider alternative approaches. These approaches exploit the actual topological properties of the graph underlying the network. The expectation is that the resulting adaptive and distributed routing scheme would provide a competitive memory-routing stretch tradeoff, robustness against topology dynamics while preserving relative "simplicity" to enable deployment in wide-scale environments.

---

[2] The routing path stretch is defined as ratio of the length of the path as produced by greedy geometric routing to the length of the shortest topological path between the same pair of nodes.

B. *Geodesic Geometric Routing*

Instead of assigning (virtual) coordinates and compute distances from these coordinates, geodesic geometric routing operates by computing the distances between vertices from the length of the corresponding geodesic drawn out of negatively curved geometric space (the hyperbolic plane). It then derives the vertex coordinates from the selected geodesics. The theoretic foundations underlying the geodesic routing approach are derived from [24].

More formally, a metric space (X, d) is said to be a geodesic (metric) space, if every pair of points $x, y \in X$ can be joined by a geodesic path. A geodesic path joining $x \in X$ to $y \in X$ (or, more briefly, a geodesic from x to y) is a map μ from a closed interval $[0, l] \in \mathbb{R} \rightarrow X$ such that $\mu(0) = x$, $\mu(l) = y$ and $d(\mu(a), \mu(b)) = |a - b|$, $\forall a, b \in [0, l]$; in particular, $l = d(x, y)$, i.e., the length of the geodesic path joining x to y coincides with the distance $d(x, y)$. Following the Hopf-Rinow theorem (see [24]), a complete, connected Riemannian manifold characterized by its sectional curvature κ is a geodesic space; however, the graph modeling a network topology is a discrete structure instead of a differentiable structure. By reformulating the concept of curvature in terms of distance, the latter becomes applicable to graphs following the seminal work of M.Gromov [25]. Intuitively, the δ-hyperbolic property introduced by M.Gromov measures the deviation of the graph from tree-likeness (obtained when $\delta = 0$).

A quasi-geodesic follows closely the geodesics if the geodesic space itself is hyperbolic (i.e., negatively curved). Moreover, following the Theorem III.H.1.13 in [24], quasi-geodesics are k-local geodesics if X is a δ-hyperbolic space provided that k satisfies certain conditions with respect to the value of δ. However, the δ-hyperbolic property provides a characterization at infinite scale. For finite geodesic metric spaces, and in particular for finite graphs G, the issue is whether δ is "small" compared with its diameter $\Delta(G)$. Indeed, as demonstrated in [26], a finite metric geodesic space in which the ratio $\delta/\Delta(G)$ is (strictly) bounded from above by 3/2 for all geodesic triangles exhibits the same metric properties as a Riemannian manifold of negative curvature $\kappa < 0$.

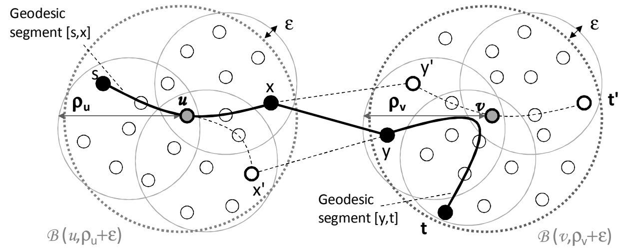

Fig.3. Geodesic Geometric Routing

Consequently, it suffices to determine the value of k such that the ball $B(x, \rho)$ of radius ρ centered at x defines a ρ-geodesic space of negative curvature. Provided that the edges of the graph are properly weighted, the selection of routing paths corresponding to quasi-geodesics ensures in turn that the routing path stretch remains bounded by δ. In general, the closer the values of δ to 0, the lower the increase of the routing path stretch. This rule of thumb is also verified by this scheme; it also explains the recent interest in determining the δ-hyperbolicity of large-scale networks. The stability property of quasi-geodesics (see Theorem III.H.1.7 [24]) implies that these constructions are robust and offer the possibility to consider alternate geodesic path(s) to ensure destination reachability.

At the inter-domain level, geodesic geometric routing operates as follows: the k-local geodesics are extended to $(k+1)$-local geodesics. At the intra-domain level, it operates similarly to a path vector (per destination t) where a single geodesic segment is communicated to the preceding node until reaching domain boundary nodes. The example depicted in Fig.3 shows how the geodesic geometric routing operates. Assume that source s receives content locator t, t'; since the distance $d(s, t) < d(s, t')$ the source s selects t as destination. Routing from $s \in B(u, \rho)$ to $t \in B(v, \rho')$ runs as follows: s uses the geodesic path segment $[s, x]$, x then selects y along the geodesic path segments $[x \rightarrow y, t]$. The algorithm performs by avoiding multiplicity of paths to the same locator. Comparison with the Border Gateway Protocol (BGP) shows that geodesic geometric routing provides remains competitive in terms of memory-stretch tradeoff.

- *Routing state*: BGP stores $O(f(n))$ routing states per node where the function $f(n) = (n-1)!$ for a complete graph. However, the topology underlying the Internet does not form a complete graph, we can relax this upper bound by assuming that each node accumulates $(n-1)$ routing states corresponding to a destination each and sends them to its neighbor nodes. Hence, each node stores $O(\nu(n-1)^2)$ states where the size of the neighbor set of each node $|N(u)| = \nu$. On the other hand, assuming that each geodesic routing processes at most $O(\nu n)$ states.
- *Memory space*: assuming that the average BGP path length λ determines the size of each routing entry, the memory-space (measured in memory-bits unit) required at each node reaches $O(\nu(n-1)^2\lambda)$. Assuming that each geodesic is represented by a succinct coordinate pair, geodesic routing requires $O(\nu\, n \log(n))$ memory bits to store locally the routing table entries.
- *Stretch*: for simplicity, we consider here the additive stretch which measures the difference between routing path length and the topological path length. BGP belonging to the class of shortest-path routing the additive stretch of this routing algorithm is 0. On the other hand, the additive stretch of geodesic routing upper bound is determined by $\delta \log(n)$ where δ characterizes the hyperbolicity of the graph underlying the topology.

The above analysis shows (as expected) that decreasing the memory space consumption comes at the detriment of the routing path stretch. However, assuming in-network caching is enabled along routing paths, its effect would be further limited. As previously stated, it also explains the importance of proper characterization of the value δ for the network environments under consideration. Observe this upper bound fulfills the expectation of routing path stretch being polylogarithmic in the number of nodes n.

Actually the main challenge of geodesic routing concerns the coordinate assignment process. There are basically two methods to assign (hyperbolic) coordinates in dynamically evolving networks. The first method relies on the Time-Difference Of Arrival (TDOA) technique which consists in estimating the (hyperbolic) location of a source from the arrival time at receivers of signal reaching receivers along geodesics. The second method, referred to as HyperMap [27] is "constructive" in the sense that it replays the geometric growth of the topology by determining at each step the hyperbolic coordinates of newly added nodes whose appearance time is estimated by means of the Maximum Likelihood Estimation (MLE) technique. Several techniques exist that assign (hyperbolic) coordinates in stationary conditions. However, finding (estimation) methods offering a compelling tradeoff between accuracy, computational complexity and efficiency in dynamically evolving settings is still an active area of research.

V. CONCLUSION

This paper investigates alternatives to the overlay routing and name-based routing model when applied to information-centric networks. Indeed, the former shows architectural deficiencies (resulting from either addressing space or communication stack design; thus, similar to those that have hampered the mobile IP and IP multicast on the Internet). The latter shows scalability limits in terms of memory space required to locally store the (name-based) routing table entries with an increase of about 3 orders of magnitude compared to BGP routing tables. The proposed alternative considers assignment of content locators to content objects, where content locators identify the "position" of the content in the content-centric network. Content locators are drawn from hyperbolic space which allows in turn considering geometric information routing that relies on hyperbolic coordinates.

Two variants of geometric routing are considered: greedy geometric routing and geodesic geometric routing. Instead of assigning (virtual) coordinates and compute distances from these coordinates, geodesic geometric routing operates by computing the distances between vertices from the length of the corresponding geodesic drawn out of negatively curved hyperbolic plane. It then derives the vertex coordinates for the selected geodesics. Up to certain extent, with geodesic routing the addressing space follows the topology whereas with greedy geometric routing the topology follows the addressing space build upon global/network-wide structure. This difference leads to deep implications in terms of routing path stretch, succinctness but also robustness. Research is still ongoing to determine if geometric routing either by means of greedy embeddings (greedy geometric routing) or geodesic segments (geodesic geometric routing) could provide a suitable answer to the routing (and addressing) challenges of information-centric networks. Indeed, on the one hand, limits of greedy geometric routing as analyzed in this paper are now well understood and efforts are ongoing to mitigate their effects. On the other hand, geodesic routing still lacks a definitive answer concerning the best tradeoff between accuracy, computational complexity and efficiency in dynamically evolving networks. In both approaches, the question as of which entity will actually compute and assign the coordinates which correspond to the "location" of the nodes hosting or temporarily caching content remains open.

REFERENCES

[1] V.Jacobson, D.K.Smetters, J.D.Thornton, M.F.Plass, N.H.Briggs, and R.L.Braynard, Networking named content, Proc. of CoNEXT, 2009.

[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, Proc. of ACM SIGCOMM, 2007, pp.181-192.

[3] V.Jacobson, J.D.Thornton, D.K.Smetters, B.Zhang, G.Tsudik, k.claffy, D.Krioukov, D.Massey, C.Papadopoulos, T.Abdelzaher, L.Wang, P.Crowley and E.Yeh, Named Data Networking (NDN) project. http://named-data.net/techreport/TR001ndn-proj.pdf, 2010.

[4] N.Fotiou, P.Nikander, D.Trossen, and G.C.Polyzos, Developing information networking further: From PSIRP to PURSUIT, Proc. of BROADNETS, 2010.

[5] FP7 COntent Mediator architecture for content-aware nETworks (COMET) Project. [Online]. Available http://www.comet-project.org/.

[6] FP7 CONVERGENCE project. [Online].
Available: http://www.ictconvergence.eu/

[7] R.G.Xylomenos, C. N.Ververidis, V.A.Siris, N.Fotiou, C.Tsilopoulos, X.Vasilakos, K.V.Katsaros, and G.C. Polyzos, A Survey of Information-Centric Networking Research, To appear in IEEE Communications Surveys and Tutorials.

[8] P.Baran, On Distributed Communications Networks, RAND Corporation papers, Document P-2626.

[9] J.Shoch, IEN (Internet Experiment Note) document: IEN 19, A note on Inter-Network Naming, Addressing, and Routing, January 1982.

[10] D.Spinellis, Another level of indirection, In A.Oram and G.Wilson (Eds), Beautiful Code: Leading Programmers Explain How They Think, Ch.17, pp.279–291, O'Reilly and Associates, Sebastopol (CA), 2007.

[11] R.Moskowitz, P.Nikander, Host Identity Protocol (HIP) Architecture, Internet Engineering Task Force (IETF), RFC 4423, May 2006.

[12] Verisign, The Domain Name Industry Brief, vol.10, issue.1, April 2013.

[13] E.W.Dijkstra, On the role of scientific thought. Selected writings on Computing: A Personal Perspective. New York, NY, USA, Springer-Verlag. pp.60–66, 1982.

[14] D.Kutscher (Ed.), ICN Research Challenges, Internet Research Task Force (IRTF), Work in progress, July 2013.

[15] S.K.Fayazbakhs, Y.Lin, A.Tootoonchian, A.Ghodsiz, T.Koponen, B.M. Maggs, K.C.Ng, V.Sekar, and S.Shenker, Less Pain, Most of the Gain: Incrementally Deployable ICN, Proc. of ACM SIGCOMM, 2013.

[16] C.Davis, P.Vixie, T.Goodwin, and I.Dickinson, A Means for Expressing Location Information in the Domain Name System, Internet Engineering Task Force (IETF), RFC 1876, January 1996.

[17] C.Vogt. Simplifying Internet Applications Development With A Name-Based Sockets Interface, 2009.

[18] Name Based Sockets project. https://www.sics.se/projects/name-based-sockets.

[19] D.Papadimitriou, Increasing the success rate in geometric routing upon distance re-labeling and coordinate re-assignment, Tech. Report, 2011.

[20] R.Kleinberg, Geometric routing using hyperbolic space, Proc. of 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'07), pp.1902-1909, Anchorage (AK), May 2007.

[21] C.H.Papadimitriou, D. Ratajczak, On a Conjecture Related to Geometric Routing, Theoretical Computer Science, vol.344, no.1, pp.3–14, 2005.

[22] D.Eppstein and M.T.Goodrich, Succinct Greedy Geometric Routing using Hyperbolic Geometry, IEEE Transactions on Computers, vol.60, no.11, pp.1517-1580, November 2011.

[23] A.Cvetkovski and M.Crovella, Hyperbolic embedding and routing for dynamic graphs, Proc. of 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'09), April 2009.

[24] M.R.Bridson and A.Haefliger, Metric spaces of non-positive curvature, Grund. Math. Wiss., 319, Springer-Verlag, Berlin, 1999.

[25] M.Gromov, Hyperbolic groups, Essays in group theory (S. M. Gersten, Ed.), Springer Verlag, MSRI Publ. 8, pp.75-263, 1987.

[26] E.A.Jonckheere and P.Lohsoonthorn, Geometry of network security. In Proceedings of the American Control Conference (ACC 2004), Boston, (MA), USA, June 2004.

[27] F.Papadopoulos, C.Psomas, and D.Krioukov, Network Mapping by Repalying Hyperbolic Growth, June 2013 [Online]. Available: arxiv.org/abs/1205.4384.