Prototyping a Network of Information

Christian Dannewitz and Thorsten Biermann
University of Paderborn, Germany
{christian.dannewitz|thorsten.biermann}@uni-paderborn.de

***Abstract*— The information-centric nature of today's main Internet use cases contrasts the node-centric paradigm of the Internet architecture. This results in problems like inefficient information dissemination and limited data availability. To tackle these problems, information-centric networking has become an important research area. In this paper, we present a prototype of an information-centric network in combination with three applications, illustrating how new as well as legacy applications can benefit from an information-centric network architecture.**

I. Introduction

Early use cases that drove the development of the current Internet architecture have been node-centric, e.g., to enable remote connections to a certain network node. As a result, the guiding principle of today's Internet architecture is based on a node-centric paradigm. However, the usage of the Internet has changed dramatically during recent years and focus has shifted towards accessing and disseminating information. For these use cases, the source (i.e., network location) of the information is rather irrelevant from a user's point of view. The user cares about getting the right information as fast as possible. We call this paradigm *information-centric networking*.

Research on information-centric network architectures has gained significant importance in recent years and is pursued by several projects like Content-centric Networking (CCN) [1], PSIRP [2], and 4WARD [3]. This paper is based on results of the 4WARD project and describes the prototype of our information-centric network architecture, which we call *Network of Information (NetInf)*. In NetInf, we take *information per se* as the starting point for the network architecture in contrast to today's host-centric architecture. We design a communication infrastructure which is much better adapted to the task of disseminating and exchanging information. Akin to similar projects, large-scale information dissemination is based on an identifier/locator split. Going beyond similar projects, we improve information dissemination by integrating caching and data storage right into the architecture. Furthermore, NetInf increases data availability via *resolution services* that can provide the best network location for information access in a certain context. Information availability is increased via a flexible, encoding-independent information model. Information access is secured via an information-centric security model which is deeply integrated into the architecture and enables, among others, owner authentication and self-certification.

In this paper, we will first give a brief overview of the NetInf prototype in general (Section II). Thereafter, Section III describes the three use cases that our demonstration focuses on: (1) The *Augmented Internet* use case that provides support for applications focusing on real-world/Internet integration, (2) a NetInf-based Web browser plugin that supports persistent, location-independent links, and (3) an email client plugin that uses persistent person identifiers (IDs) and resolves the destination email addresses on the fly via NetInf. Section III-D briefly describes the demo setup requirements.

II. The Prototype

This section describes our current NetInf prototype implementation. All participating nodes provide an information-centric Application Programmer Interface (API), called *NetInf API*, that supports searching for information and resolving information IDs into appropriate content locators. Nodes providing this API are called *NetInf nodes* (*NIN*) (Figure 1).

Each individual NetInf node is able to perform lookup and search locally, e.g., to respond to broadcast requests from neighbor nodes. In addition, our prototype implements an infrastructure of NetInf nodes that provides these services on a global scale. These services are used by other nodes (clients) to easily implement various information-centric applications.

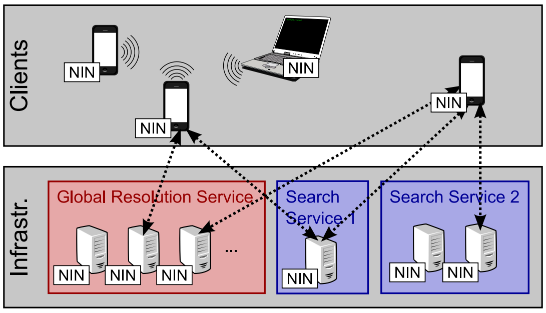


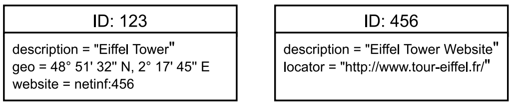
Fig. 1. Architecture overview. All participating nodes are NetInf nodes (NIN).

A. Information Objects

All information in NetInf is stored in Information Objects (IOs). IOs can represent files or services that are provided outside of NetInf, e.g., by a Web server, or they can represent real-world objects, like a person or a building. All IOs have a common structure: they consist of a set of attribute/value pairs that describe the IO and have a globally unique *NetInf ID* (Figure 2). NetInf IDs are persistent in spite of owner change, content change, as well as storage location change.

B. Infrastructure

The infrastructure consists of the *Identifier Lookup Service (ILS)* and the *IO Lookup Service (IOLS)*. The IOLS is a



(a) IO representing the Eiffel Tower.

(b) IO representing the website of the Eiffel Tower.

Fig. 2. Two IOs with ID 123 and 456. IO 123 refers to IO 456.

global resolution service that performs lookup operations for clients, i.e., it returns a corresponding IO for a given NetInf ID. This service is the central component for a global NetInf and, hence, must scale and must have a high availability. Therefore, we implemented the IOLS based on a Distributed Hash Table (DHT) using Chimera [4]. Note that this Peer-to-Peer (P2P) implementation lacks important features, like low query latency, which is acceptable for our prototype. A productive implementation would use more sophisticated techniques, like Multiple DHTs (MDHT) [5].

The second component of the infrastructure are search services. These services perform search operations for clients, i.e., return a set of NetInf IDs for a given attribute query, like `rfid == 'A1B2C3'`. This way, IOs can be found based on their descriptive attributes without knowing their ID.

We have implemented two different search services. First, the distributed *exact-match ILS* is able to resolve exact-match queries for arbitrary attributes. It can be used for, e.g., Radio Frequency Identification (RFID) and barcode lookups. Second, the *geographic ILS* performs advanced searches of IOs based on a geographic location specified, e.g., via rectangles, squares, or pie slices. For complexity reasons, the prototype implements the geographic ILS in a centralized manner; the implementation of a highly scalable productive system is out of the scope of NetInf.

Our current infrastructure testbed consists of several servers running FreeBSD. All of them participate in the Chimera P2P overlay which is the base for the IOLS and exact-match ILS. One of the servers additionally hosts the central geographic ILS. An overview is shown in Figure 3.

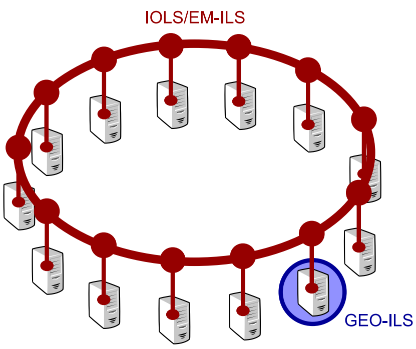


Fig. 3. Testbed infrastructure with IOLS, exact-match ILS (EM-ILS), and geographic ILS (GEO-ILS).

C. Clients

To use the infrastructure, client applications communicate with the IOLS and ILS via the NetInf API. This API is provided by the *NetInf Node Middleware (NNM)* that permits to use this API remotely (implemented via Google Protocol Buffer [6] messages). The resulting interaction between client and infrastructure is illustrated in Figure 4.

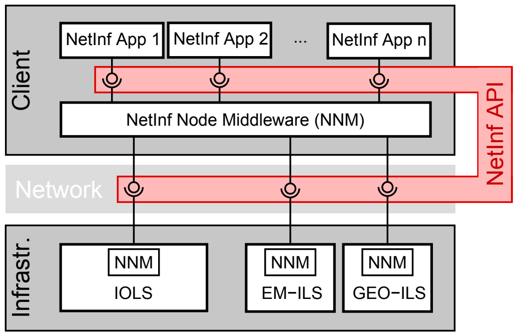


Fig. 4. Information-centric applications (NetInf apps) access the NetInf functionality via the NetInf API. The global infrastructure provides the same API to provide NetInf service on a global scale.

The middleware is written in Java to guarantee high portability between different system platforms. Currently, we use it on Android, Windows, and Linux to implement several NetInf applications as described in Section III.

III. Demonstration

A. Augmented Internet Use Case

The Internet contains a lot of information that can be very useful to support users in real-world activities and while on the go. Unfortunately, this information is currently very cumbersome to access and retrieving the information disrupts the users' workflow. Several Internet applications are being developed lately that provide a better real-world/Internet integration, e.g., for the iPhone and Android operating system [7], [8]. We call those applications *AugNet applications*. For example, this includes applications that provide users with information corresponding to their current surroundings (Fig. 5). However, AugNet applications are currently difficult to develop on a large scale because conceptual support for such applications is missing in today's network architecture.

In contrast, our information-centric NetInf architecture inherently provides conceptual support for AugNet applications: IOs can be used to represent real world entities like buildings, places, and objects in the Internet, thereby becoming *Virtual Entities* that represent the respective real world entities. Based on the real world entity attributes like GPS coordinates or RFID tags, users can search for Virtual Entities via specialized search services like the geographic ILS, retrieving the IDs of Virtual Entities that match their query. Finally, these IDs can be resolved into the respective Virtual Entities via NetInf resolution services (e.g., locally or globally).

We have implemented and will demonstrate an information-centric, location-dependent AugNet application called *AugNet* *browser* running on an Android phone. The main goal is to illustrate how the NetInf architecture can be used to implement such applications very easily. With the AugNet browser, a user can create Virtual Entities and can bind them to real world entities via GPS coordinates, RFID, etc. Likewise, users can search for Virtual Entities based on real world attributes and can present them, e.g., on a map (Figure 5).

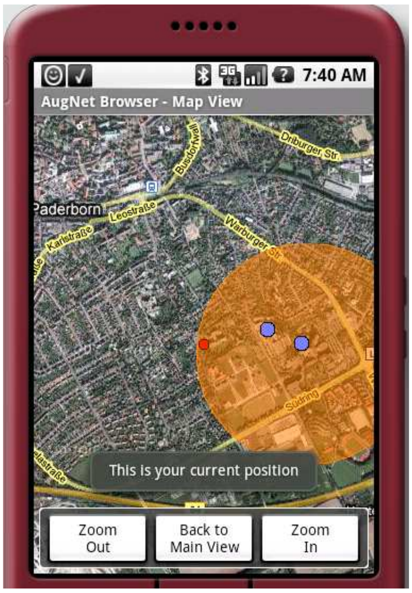


Fig. 5. AugNet browser application showing two retrieved Virtual Entities

B. NetInf Web Browser

In the dominating namespaces of today's Internet – IP addresses and URLs – names are bound to certain network locations. This fact hampers scalability mechanisms, makes persistent information naming difficult, and reduces information availability. The NetInf architecture solves these problems via its naming framework that names information persistently and location-independently via NetInf IDs.

To illustrate how these advantages can be provided for legacy applications, we have developed and will demonstrate a Firefox plugin that supports websites which link to other websites based on their NetInf IDs instead of URLs. When a NetInf ID is clicked, the plugin sends a request to a NetInf resolution service. The resolution service resolves the NetInf ID into several locators. The best locator based on some adjustable metric is chosen either by the resolution service or by the local plugin. Finally, the requested website is downloaded and displayed. In addition, the Firefox plugin can benefit from all other NetInf functionality like information-centric security and integrated content caching.

C. NetInf Email Client

Email addresses have become important personal identifiers for Internet users. However, this identifier is not persistent as the business email address typically changes when changing the company. Also, most people have one or more private email addresses in addition that might be bound to their current Internet Service Provider. Changing email addresses produces significant hassle to inform all contacts of the new email address and might even result in lost contacts.

This problem can be solved by using NetInf IOs as persistent representations of people, called *Person Object (PO)*. A PO can contain a varying amount of information about the PO's owner and access to this information can be restricted as desired. To provide persistent email communication, the PO contains at least one valid email address of the owner. Instead of propagating his current email address, the user propagates the PO's ID to his contacts. In return, these contacts can use the PO's persistent ID as destination address instead of the user's email address. When his email address changes, the user only has to update the email address in his PO instead of propagating the new email address to all contacts. This is possible because NetInf IDs (including the IDs of the POs) are inherently persistent even in the event of a provider change.

We have developed a Thunderbird plugin that translates the PO's ID into the currently valid email address of the user. In our demonstration, we will illustrate how the plugin retrieves the user's email address and how the core NetInf functionality is used to realize this use case. First, the plugin resolves the PO's ID into the corresponding PO via the resolution service. Thereafter, the plugin extracts the email address from the retrieved PO, uses this email address as destination address, and hands the email back to Thunderbird for transmission.

D. Setup

To reduce the complexity of the setup, all demos will use a single local NetInf node running on a laptop to perform search and ID resolution, i.e., we require space for two laptops. In addition, we use an Android cellphone for the AugNet use case. We require WLAN with Internet access for three devices as well as power supply. Setup will take about 30 min.

REFERENCES

[1] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves, "Content-centric networking," Palo Alto Research Center," Whitepaper, January 2007.

[2] C. E. Rothenberg, F. Verdi, and M. Magalh, "Towards a new generation of information-oriented internetworking architectures," in *Proc. First Workshop on Re-Architecting the Internet*, Madrid, Spain, 12 2008.

[3] B. Ahlgren, M. D'Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, B. Ohlman, K. Pentikousis, R. Rembarz, O. Strandberg, and V. Vercellone, "Design considerations for a network of information," in *Proc. First International Workshop on Re-Architecting the Internet (ReArch2008)*, Madrid, Spain, December 2008.

[4] CURRENT Lab, UC Santa Barbara, "Chimera, a light-weight & efficient implementation of a structured peer-to-peer," 2009. [Online]. Available: http://current.cs.ucsb.edu/projects/chimera/

[5] M. D'Ambrosio, P. Fasano, M. Marchisio, V. Vercellone, and M. Ullio, "Providing data dissemination services in the future Internet," in *World Telecommunications Congress (WTC'08)*, New Orleans, LA, USA, Dec. 1-2, 2008, at IEEE Globecom 2008.

[6] Google, "Protocol Buffers – Google's data interchange format," 2009. [Online]. Available: http://code.google.com/p/protobuf/

[7] Mobilizy, "Wikitude AR Guide," 2008. [Online]. Available: http://www.mobilizy.com/

[8] M. Braun and R. Spring, "Enkin," 2008. [Online]. Available: http://www.enkin.net/