

Demonstrating Usage Diversity Over an Information-Centric Network

George Parisis*, Ben Tagger*, Dirk Trossen*,
Christos Tsilopoulos†, Yannis Thomas†, Charilaos Stais† and George Xylomenos†

*Computer Laboratory, University of Cambridge, Cambridge, UK

Email: {firstname.lastname}@cl.cam.ac.uk †Department of Informatics, Athens University of Economics and Business, Athens, Greece

Email: {tsilochr,thomasi,stais,xgeorge}@aueb.gr

Abstract—Information-Centric Networking (ICN) is becoming increasingly popular: various architectures have been proposed to transform today’s host-centric networks to information-centric ones. An important question that begs to be answered is whether the ICN paradigm can adequately support various types of existing network usage, as well as set the foundations for new application types. In this demonstration, we exhibit four different communication models via corresponding applications supported by our architecture. All applications are realized through the publish/subscribe service model exported by our ICN nodes.

I. INTRODUCTION

Information-centric networking (ICN) has been attracting increasing attention, with various efforts creating architectures, design choices and network artefacts. Crucial in showcasing the suitability of ICN for eventually providing an alternative to the IP architecture is its support for various network usage models and the fostering of novel applications. In this demonstration we categorize these usages into four distinct areas, presenting applications for each of them. All applications are built within our ICN architecture presented in [1], elaborated in [2] and [3], and implemented on top of our prototype [4].

Firstly, we demonstrate the *native multicast* support of the underlying forwarding fabric within our architecture through a multi-party audio conferencing demo. Secondly, we show how the exposure of individually identified information objects to the network allows for exploiting *fountain codes* for reliable content dissemination. In this context there are opportunities for efficient caching, multi-path and multi-source support. Thirdly, we present a novel proposition for delivering content in the Internet by decomposing it to its individual media assets, each of which is identified individually, while the overall content is disseminated as a *storyline metadata*, pointing to the individual media assets. Last but not least, we showcase the support for *legacy applications* by exposing a socket abstraction on top of the ICN stack, enabling backward compatibility with existing applications.

In the following section, we describe in more detail the applications to be demonstrated. While our list is not exhaustive with respect to the opportunities that our ICN architecture provides, this work integrates a diverse set of usage models within a single demonstration over an international ICN deployment.

II. USAGE MODELS IN AN ICN ENVIRONMENT

1) *Multi-Party Audio Conferencing*: Exploiting the publish/subscribe service model and the underlying forwarding mechanism supported by our ICN, we demonstrate a real-time multi-party audio conferencing application that supports a dynamic group of users without requiring a *Multipoint Control Unit (MCU)*. In particular, we exhibit multi-party interactive audio exchange, allowing the participation of a predefined maximum number of endpoints, with latecomers joining in at any time. Utilizing the information-centric perspective of content delivery, each participant advertises a unique information stream and subscribes to the information streams of the other participants. Stream identifiers (represented as information items in the network) are algorithmically calculated based on the maximum number of participants in the conference and correspond to the virtual seats of the conference table, hence they are known to participants during bootstrapping.

By exploiting the native multicast capability in our ICN [5], our application avoids the delay introduced by MCUs relaying each stream to the participants. Instead, each participant directly multicasts its audio stream to all others using a forwarding identifier that is dynamically updated when the participant list changes. Listeners receive an audio stream per participant and can mix the distinct channels based on local preferences, instead of relying on a global mix policy determined by the MCU.

2) *Reliable Content Distribution*: For this demonstration we present a simple application that exploits the publish/subscribe nature of our network node along with a fountain coding technique [6] to reliably disseminate information to a potentially changing set of subscribers. New publishers can also advertise the same item at any time, creating a CDN-like network that can be easily deployed and managed. Digital fountains involve coding information to a very large number of symbols. Receivers decode the original content after having received a number of symbols which is slightly larger than the number of fragments comprising the content.

In our scheme, a publisher advertises the content it stores and is notified by the network stack when subscribers subscribe to it. Then, it starts publishing encoded symbols. To create each symbol, it calculates the number of fragments (of constant and known size) from the content that will be

encoded in the symbol (the symbol *degree*) [6], selects that many fragments (the *neighbour set*) and encodes them into the final symbol by XORing them. A receiver utilizes symbols with *degree* 1 to partially or fully decode other symbols by XORing them with the decoded symbol. The receiver only needs to know the degree and neighbour set of each symbol.

We utilize the principles of information labelling and scoping, which are inherent to our ICN architecture, to create statistically unique identifiers for encoded symbols. We then utilize the native multicast support of our ICN, to multicast encoded symbols to a changing set of subscribers. Information labels are used to communicate encoding-related information that is necessary for subscribers to decode each symbol by calculating its degree and neighbours. Publishers utilize a pseudo-random number generator with randomly generated seeds for each symbol to calculate the degree and neighbour set for each symbol. Assuming that all subscribers utilize the same generator, the *seed* is the only information that needs to be communicated in the statistically unique label assigned to each encoded symbol. Note that subscribers can join at any time, even while symbols are being published. Each subscriber only needs to stay in the fountain until it is able to decode the initial content; publishers stop publishing symbols when they are notified that no more subscribers exist.

3) *Media Story Delivery Network (M-SDN)*: The delivery of personalized content remains largely elusive in the current Internet. Our M-SDN solution provides a framework for measuring the end user's needs, capabilities and permissions and matches that to available media assets within the network. The key to enabling this environment is the metadata management that we implemented within the M-SDN application, its underlying ontology-based middleware and our ICN architecture.

We demonstrate the M-SDN applications that formally specify (through metadata) media and media stories within the network. We then deliver these stories to the viewer based on a set of their implicit and explicit metadata (captured within a GUI). The efficacy of such a system depends on the efforts placed at specifying the media as well as capturing the viewer's requirements. Ultimately, however, our M-SDN application utilizes the explicit naming of media assets, therefore utilizing any potential provider for the asset throughout the network. This effectively democratizes the media delivery across caches, end user storages and professionally provided content storages. We demonstrate a scenario where two viewers subscribe to a single media story from heterogeneous locations and we then take a look at their respective (and differing through policy dependencies) media experiences.

4) *Legacy Application Support*: Gradual deployment of new network architectures involves a transition phase during which new applications co-exist with legacy ones. The transition phase must be transparent to legacy applications so as to avoid re-building and re-deploying them.

We demonstrate the backwards compatibility of our system with a Socket API emulator [7]: a software library that operates on top of our network node prototype and provides the well-known socket abstraction to applications by mapping *send/receive* socket calls to *pub/sub* primitives. To receive packets on a socket (UDP or TCP) with address *IP:port*, the

emulator subscribes to the item with identifier */proto/IP/port*, where *proto* is UDP or TCP. Similarly, to send packets to a socket with address *IP:port*, the emulator becomes a publisher of */proto/IP/port*. The network matches these publications and subscriptions and creates appropriate forwarding information that is sent to the publisher. The uniqueness of the above identifiers results in the creation of LIPSIN identifiers representing unidirectional pipes between the sender and the receiver; source-specific IP multicast can also be supported by simply having many receivers subscribe to the same item.

We present two example applications, namely FTP for byte-stream socket (TCP) compatibility and TFTP for datagram socket (UDP) compatibility. We use open-source FTP and TFTP implementations that are slightly modified in order to use the *emulated* sockets API. In a production deployment, emulated sockets would be provided by the host operating system, thus the operation would be completely transparent. The rest of each file transfer application remains untouched, yet fully functional while operating on top of our ICN.

III. ICN DEMONSTRATION ENVIRONMENT

Our demonstrations are showcased within an international deployment of our ICN prototype; a testbed that spans several European, US and Japanese sites. We utilize the standard Internet by tunnelling the prototype's Ethernet traffic via an OpenVPN overlay. Throughout this overlay network, more than 40 ICN nodes currently operate in a permanent installation, allowing for demonstrations, experimentations and tests to take place throughout several international ICN collaborations. Our demonstration showcases the usage variety of ICN within such an international environment, as well as the general abilities of ICN technologies and solutions.

ACKNOWLEDGMENT

The work reported in this paper was supported by the FP7 ICT project "Publish Subscribe Internet Technology" (PURSUIT), under contract ICT-2010-257217.

REFERENCES

- [1] D. Trossen, M. Särelä, and K. R. Sollins, "Arguments for an information-centric internetworking architecture," *Computer Communication Review*, vol. 40, no. 2, pp. 26–33, 2010.
- [2] D. Trossen and G. Parisi, "Designing and realizing an information-centric Internet," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 60–67, 2012.
- [3] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, "Caching and mobility support in a publish-subscribe Internet architecture," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 52–58, 2012.
- [4] PURSUIT. (2013) Blackadder node implementation. [Online]. Available: <https://github.com/fp7-pursuit/blackadder>
- [5] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," in *SIGCOMM*, 2009, pp. 195–206.
- [6] M. Luby, "Lt codes," in *Proceedings of the 43rd Symposium on Foundations of Computer Science*, ser. FOCS '02, 2002, pp. 271–.
- [7] G. Xylomenos and B. Cici, "Design and evaluation of a socket emulator for publish/subscribe networks," in *FIS*, 2010, pp. 11–19.