

Demonstrating an Information-Centric Network in an International Testbed

George Parisi, Ben Tagger, Dirk Trossen
Cambridge University, Computer Lab, Cambridge, UK
Email: {georgios.paris,ben.tagger, dirk.trossen}@cl.cam.ac.uk
Dimitris Syrivelis, Paris Flegkas, Leandros Tassioulas
CERTH-ITI, Volos, Greece
Email: {jsyr, pflegkas, leandros}@inf.uth.gr
Charilaos Stais, Christos Tsilopoulos and George Xylomenos
Athens University of Economics and Business, Athens, Greece
Email: {stais, tsilochr, xgeorge}@aueb.gr

Abstract—Information-Centric Networking (ICN) has increasingly been attracting attention by the research community. In ICN the center of attention becomes the information itself and not the endpoints as in today’s IP networks. In this demonstration we present applications that we developed as proof of concepts for our ICN approach. A video streaming as well as a voice and a HTTP over publish/subscribe application that run on top of our ICN prototype will be demonstrated running in an international testbed.

Information-centric networking has been touted as an alternative to the current Internet architecture by several research groups. Several technological solutions within a range of architectures have been proposed, such as in [1], [2], [3]. In this proposed demo we present three applications that we have implemented on top of the network prototype, named Blackadder [8], which we have implemented in the context of the EU FP7 project PURSUIT [4]. Blackadder is based on the principles outlined in [1] and is implemented using the Click router project [5].

The Testbed. We have deployed our prototype in an international network in the context of the EU FP7 PURSUIT project [4]. As shown in Figure 1, eight sites across Europe as well as MIT in the US currently host 26 virtual machines, interconnected in an OpenVPN overlay. This enables us to emulate various topologies by assigning multiple link identifiers to the single network interface of each virtual machine. Note that forwarding in our prototype (called *Blackadder* in the following) is currently based on the LIPSIN approach [6], directly being implemented on top of Ethernet. A deployment tool has been developed in order to automate the deployment of network topologies and start experiments. This tool reads a configuration file that describes all nodes in the testbed along with their connections with other nodes and creates Click [5] configuration files for all testbed nodes, which in turn are copied to the respective nodes, initiating all testbed nodes.

Demonstrated Applications. In the proposed demonstration we will present a video streaming as well as a voice and a HTTP over publish/subscribe application that run on top of our Blackadder prototype, based on the high-level architecture outlined in [3].

Video Streaming: A video store across the network publishes

an information item that represents a video catalogue. Whenever a subscriber subscribes to this catalogue, rendezvous takes place and, subsequently, the Topology Manager (TM) creates a forwarding identifier that is published to the publisher (the video store). The publisher then publishes the catalogue to the subscriber, which can then browse all published videos and subscribe to any one of them. Subscribers can calculate the identifiers of each video by hashing its name in the received catalogue. Therefore, an information identifier in this demo represents the channel used to publish the video with video fragments being published using the same identifier. The media server is a VLC server that streams the selected video via a local UDP socket. The publisher on same physical host as the video server receives the video through a UDP client socket. When subscribers for that specific information ID join the system, the publisher starts publishing the video. If no subscribers exist for the advertised information item, then the publisher remains idle. If all previously joined subscribers unsubscribe from the channel, the rendezvous system notifies the publisher to stop publishing the video. During the video transmission, subscribers may join or leave. Subscribers forward all received video fragments to a VLC client, via a loopback UDP socket, which reproduces the transmitted video. The demonstrated application is written in Java and C++.

Voice over Blackadder [7]: A voice connection in our demo is established similar to today’s Voice over IP, with the callee being the subscriber and the caller being the publisher. The information identifiers of the caller and callee are calculated by applying a hash function on each name. Firstly, the caller publishes to the callee’s identifier and includes in that message its own identifier, in order for the callee to be capable to send messages back to the caller. The callee publishes to the received caller identifier as an acknowledgement. With this, the participants are able to commence a two-way voice data message exchange. In order to identify successive voice messages exchanged between the two parties, we rely on information exchanged during the establishment of the connection as well as on sequence numbers. This provides some degree of flow control in that if a packet arrives with a sequence number less or equal to the previous one, we can drop it rather than push it

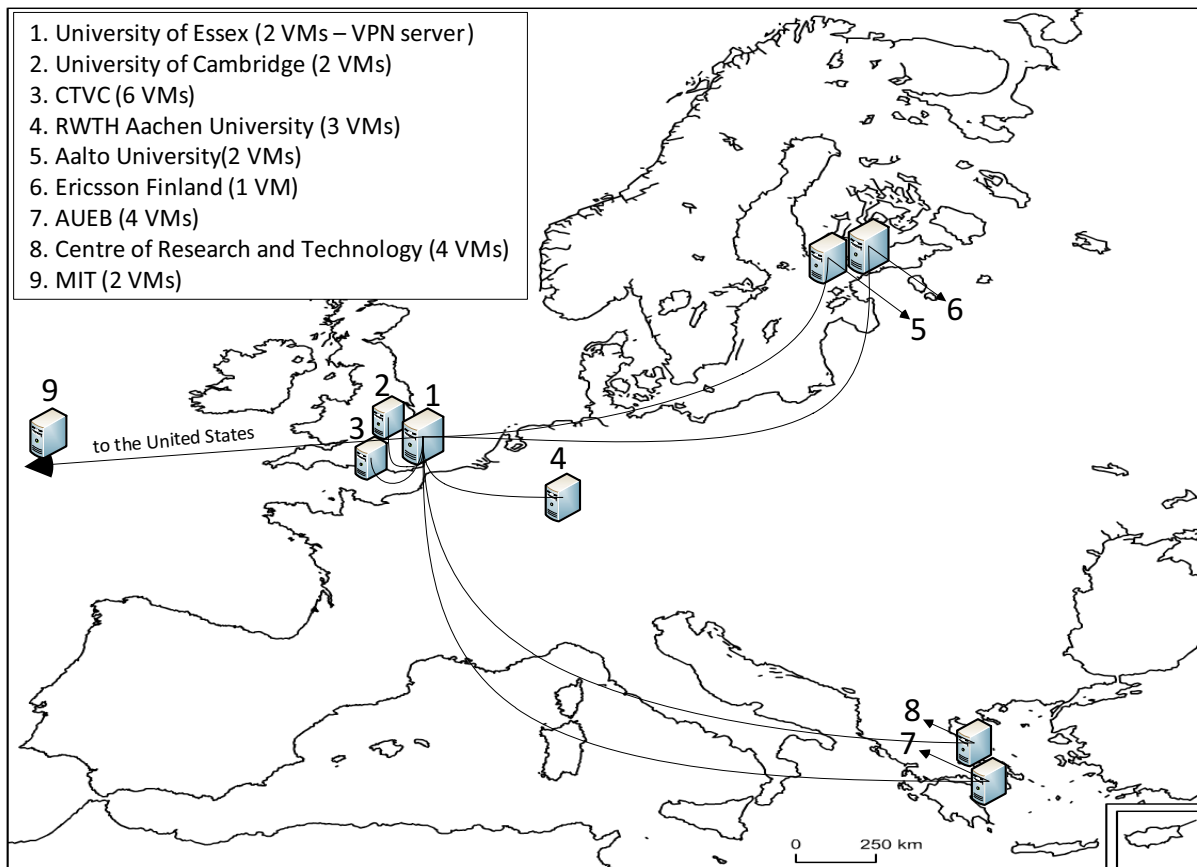


Fig. 1. PURSUIT International Testbed

to the audio component of the application. The demonstrated application is written in Java.

HTTP over Blackadder: We have implemented HTTP subscribe-GET and subscribe-POST operations for Blackadder. Each web content server, subscribes to its domain name where content browsers may publish subscribe-GET requests for a specific domain content. After publishing a request, the browser subscribes to an algorithmic identifier for that request and the web server publishes the requested content under the same request identifier. A similar transaction takes place for subscribe-POST requests. The s-GET and s-POST operations can be public or private. This is enabled by the metadata session identifier that standard HTTP requests carry. If a response data contains a session identifier, then the algorithmic content identifier for the s-GET and s-POST requests is uniquely formed to implement sessions. We have developed a python content proxy for IP browsers and a python-based Internet proxy server that fetches and publishes content directly from the Internet.

Acknowledgments. The research presented in the demo is supported by the EU FP7 project PURSUIT under grant FP7-INFISO-ICT 257217.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, R. Braynard, "Networking named content", *Communications of the ACM*, Vol. 55, No. 1, 2012.

[2] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of SIGCOMM 2007*.

[3] D. Trossen, M. Sarela, K. Sollins, "Arguments for an Information-Centric Internetworking Architecture", *ACM Computer Communication Review*, April 2010.

[4] PURSUIT project at <http://www.fp7-pursuit.eu>, 2011.

[5] E. Kohler, R. Morris, B. Chen, J. Jannotti, M. F. Kaashoek, "Click modular router", *ACM Trans. on Computer Systems (TOCS)*, Vol. 18 No 3, 2000.

[6] P. Jokela, A. Zahemszky, S. Arianfar, P. Nikander, C. Esteve, "LIPSIN: Line speed Publish/Subscribe Inter-Networking", *ACM SIGCOMM*, Aug 2009.

[7] C. Stais, D. Diamantis, C. Aretha, G. Xylomenos, "VoPSI: Voice over a Publish-Subscribe Internetwork", *Future Network & Mobile Summit*, June 2011.

[8] J. Kjällman (ed.), "First Lifecycle Prototype Implementation", PURSUIT Deliverable D3.2, September 2011 (available at <http://www.fp7-pursuit.eu>).