Using ORBIT for Evaluating Wireless Content-Centric Network Transport

Feixiong Zhang[†], Alex Reznik[∗], Hang Liu[§], Chenren Xu[†], Yanyong Zhang[†], Ivan Seskar[†]

[†]WINLAB, Rutgers University, North Brunswick, NJ, USA
[∗]InterDigital Communications LLC, King of Prussia, PA, USA
[§]Catholic University of America, Washington, D.C., USA

ABSTRACT

Evaluation of Content-Centric Networking (CCN) transport protocols, especially over wireless networks, brings about new requirements for testbeds. Typical CCN transport features such as hop-by-hop transfer, caching support, and receiver-driven traffic and congestion control differ significantly from those of classical transport protocols, most notably TCP. In this paper, we highlight an implementation of a transport protocol designed specifically for CCN and demonstrating its over-the-air, real time operation using OR-BIT - an open access platform maintained by WINLAB, Rutgers University, USA.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Protocol verification

Keywords

Testbed; Transport; Content-centric networking

1. INTRODUCTION

Current Internet is designed around a point-to-point communication model between two stationary entities identified by their network locators (IP addresses). However, over the last decade, content distribution and retrieval has come to represent a significant proportion of overall Internet traffic. The increasing importance of content has triggered a large shift to Content-Centric Networking (CCN) [1]. In contrast to the traditional communication model, CCN names content directly and routes content by its actual name. Such name-based approaches allow efficient retrieval of content wherever it's located. However, it also introduces new challenges for the transport protocol. Among these are the need for hop-by-hop transfer, caching support, and receiver-driven traffic and congestion control.

In this paper and associated demo, we aim at complementing existing CCN work by providing an implementation of

a "typical" CCN protocol and providing real testbed experimental experience in evaluating a wireless CCN transport scheme. To do so, we propose a CCN transport protocol addressing the particulars of wireless transmission, including the impact of wireless link variation and support for multi-homing. We implement a complete protocol stack associated with the protocol, and use it for our evaluation.

2. ORBIT OVERVIEW AND CAPABILITIES

ORBIT testbed [2] is developed with the goal to provide scalable and reproducible evaluation of next generation wireless network protocols. It consists of an indoor radio grid emulator, 9 sets of sandboxes, and an outdoor field trial network. The indoor grid consists of a two-dimensional 20-by-20 grid of radio nodes supporting Ethernet and a variety of radio technologies including 802.11b/g/n and WiMAX. The grid also has a centralized arbitrary waveform injection subsystem (CAWIS) located at four corners of the grid, which can be used to inject arbitrary AWGN signals into the grid, thus creating different wireless topologies. The ORBIT grid is well-suited for large-scale controlled wired/wireless experiments. In addition to the grid, ORBIT contains 9 sets of sandboxes for small scale dedicated experiments. Among these sandboxes, sandbox 4 is customized for high-accuracy wireless experimentation. Each wireless node in sandbox 4 is housed in an RF enclosure that provides 80dB of isolation, and is fed through the enclosure access port to outside RF transmission sub-system (RFTS). Such configuration enables fairly tight control of RF attenuation between nodes. Sandbox 9 is dedicated to OpenFlow experimentation and NetFPGA development. ORBIT testbed uses the cOntrol and Management Framework (OMF) and ORBIT Measurement Library (OML) for experiment control and measurement collection. A time-multiplexed reservation scheme is used, in which certain resources are exclusively used by an authorized user within a timeslot.

3. EXAMPLE TRANSPORT PROTOCOL

We briefly describe the protocol used for the demo, which we call Chunk-switched Hop Pull Control Protocol (CHoP-CoP). CHoPCoP uses the *Interest-Data* handshake and hop-by-hop transfer for data transmission, where any *Data* chunk transferred from the sender is preceded with an explicit *Interest* from the receiver and both *Interest* and *Data* are transmitted in a hop-by-hop fashion. In addition, CHoPCoP consists of the following design elements: 1) reliable hop-by-hop chunk transfer; 2) receiver-driven *Interest* control; 3)
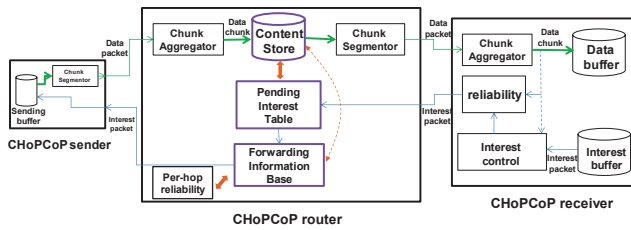
Figure 1: Model of CHoPCoP sender, router and receiver

end-to-end reliability with hop-by-hop reliability. Figure 1 illustrates the model of transport control functionalities of CHoPCoP sender, router and receiver.

As seen in Figure 1, three data structures are used for data forwarding in the router as defined in CCN [1]: the forwarding information base (FIB), content store (CS) and pending *Interest* table (PIT).

4. IMPLEMENTATION

We implement a complete network stack for our protocol as a user-level daemon based on Click Modular Router framework [3] as seen in Figure 2.
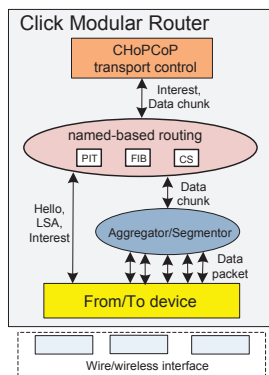

Figure 2: The software architecture of the Click-based CHoPCoP prototype

Parameters: We use 1 KB data packets and set each chunk to consists of 1024 data packets.

Packet: The following packet types are defined in our implementation: Interest, Interest ACK, Data (chunk/pkt), CSYN, CACK, 'hello' message and LSA.

Receiver-driven connection management and congestion control: A CHoPCoP association is connectionless, with no three-way handshake needed to initiate or terminate a connection. An AIMD-based *Interest* control mechanism is implemented at the receiver to manage the congestion window.

5. EVALUATION

We illustrate the capability of ORBIT in evaluating wireless CCN transport by measuring file transfer cost of CHoPCoP compared with TCP. We choose to transmit random data chunks instead of transferring real files to avoid disk operation overhead. Iperf [4] is used to emulate TCP-based data transfer.

5.1 Experimental setup

All ORBIT nodes that we use run Linux 2.6, and each has a 2.93GHz Intel i7 quadcore processor with 4GB memory.

In our experiments, we utilize link layer packet filtering technique [5] to generate a specific network topology. In this approach, packets from specified MAC address or node ID are dropped at the receiver to create logical disconnections between nodes. In addition, we use the RF transceiver system in sandbox 4 to precisely control wireless link to emulate time varying characteristics.

5.2 Benchmark evaluation

To validate basic capabilities of ORBIT in performing a CCN experiment, we conduct a benchmark evaluation of bulk data transfer of CHoPCoP compared with TCP, in a combined wired/wireless setting in Figure 3a. The content source is connected to router through a 20Mbps, 100ms delay link to emulate a wide area Internet connection.

For the network in Figure 3a, we utilize the RF transceiver system of ORBIT's sandbox 4 to change the signal level at the wireless end from best to the point of disconnection, and measure data transmission rate of CHoPCoP and TCP. Figure 3b shows that data rate achieved by our implementation out-performs TCP by as much as 30%.
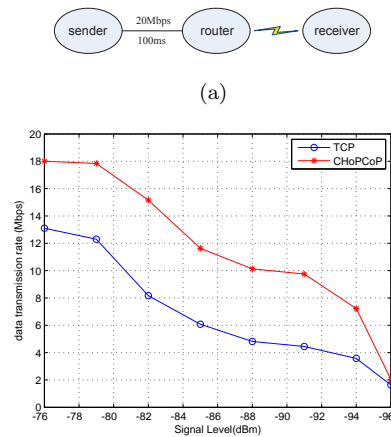

(a)

(b)

Figure 3: (a) mixed wired/wireless topology. (b) Comparing data transfer rate at different signal level

6. SUMMARY AND NEXT STEPS

We presented initial results in using the ORBIT testbed for evaluation of wireless CCN transport. More detailed evaluations, including on more complex topologies and using a multi-homes parallelized version of CHoPCoP are planned.

7. REFERENCES

[1] Van Jacobson et al. Networking named content. In *Proc. of ACM CoNEXT*, 2009.
[2] ORBIT testbed. http://www.orbit-lab.org/.
[3] Eddie Kohler et al. The click modular router. *ACM Transactions on Computer Systems 18(3)*, pages 263–297, August 2000.
[4] Iperf. http://iperf.sourceforge.net/.
[5] Gautam D. Bhanage, Yanyong Zhang, and Ivan Seskar. On topology creation for an indoor wireless grid. In *Proc. of WiNTECH*, 2008.