Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 29, 2012

S. Salsano
A. Detti
N. Blefari-Melazzi
M. Cancellieri
Univ. of Rome "Tor Vergata"

May 28, 2012

ICTP - Information Centric Transport Protocol for CONET ICN
draft-salsano-ictp-00

Abstract

Let us consider an Information Centric Networking (ICN) solution, in which an End Node requests for a content sending "content requests" (or "interest packets"). The content is provided back to the requestor by the "origin" node or by an intermediate node that had cached the content. The content is usually divided into "chunks" that can be individually requested, sent back to the requester, cached into intermediate nodes. The sending rate of content requests can be adjusted in order to perform congestion control, implementing a receiver driven transport protocol. As it can be useful to have large chunks (significantly larger than the Maximum Tranfer Unit across the network), the transport protocol should also be used to further segment the chunks rather than relying to IP fragmentation. In this memo we define ICTP (Information Centric Transport Protocol), a receiver driven transport protocol for ICN, which relies on the CONET ICN solution described in a companion draft.

Table of Contents

1. Introduction

[I-D.CONET] proposes an approach to Information Centric Networking [Koponen07][Jacobson09] based on extending the IP protocol by using a new IP Option called CONET IP option (defined both for IPv4 [RFC0791] and IPv6 [RFC2460]). The CONET IP option can be used by routers to support content aware networking, in addition to classical address based networking. Further information on the proposed solution can also be found in [CONET11].

In this memo we define a receiver driven transport protocol for CONET ICN, called ICTP - Information Centric Transport Protocol. The transport protocol is able to provide a reliable transfer of the content and to perform congestion control in TCP-friendly way. A discussion about the definition of a transport protocol for ICN can be found in [ICTP12].

As shown in Figure 1, the CONET architecture proposed in [I-D.CONET] foresees End-Nodes, Serving Nodes and CONET nodes. End-Nodes request for content. Serving Nodes provide content. CONET nodes: i) forward content requests from End-Nodes to Serving Nodes; ii) deliver content from Serving Nodes to End-Nodes; iii) may cache content and therefore provide it to End-Nodes without contacting the Serving Node.

```
                 requests for content
                 ------------------->
                 content is provided
                 <-------------------
     +----+                              +----+      +----+
     |    |                            --|    |------|    |
     +----+\                         /   +----+      +----+
            \    +----+      +----+ /
             ----|    |------|    |/
                 +----+      +----+
   End-Node      legacy    Intermediate   Border     Serving
                 IP router     Node        Node        Node
       |                                   |
       +---------CONET next hop----------->+
```

Figure 1: CONET architecture

2. CONET Basics

In this section we recall the basic aspects of the CONET ICN solution, as needed to introduce the proposed receiver driven

Information Centric Transport Protocol (ICTP).

The figure below shows the CONET protocol stack. CONET protocol is divided in two sub-layers, whose data unit are respectively denoted as "Carrier Packets" and "CONET Information Units". Two types of CONET Information Units are currently defined ("Interests CIU" and "Named Data CIU"). The CONET Information Unit Type field in the CONET IP option differentiates among the two types. A CONET Information Unit (CIU) can be split into different Carrier Packets. Each Carrier Packet is transported by an IP packet.

```
+--------+--------+--------+ \
| CONET Information Units  |  |
+--------+--------+--------+  |
                              |
+--------+--------+--------+  |- CONET protocol
|     Carrier Packets      |  |
+--------+--------+--------+  |
                              |
+--------+--------+--------+ /
| IP (with CONET IP option)|
+--------+--------+--------+
```

Figure 2: CONET protocol layers

The generic structure of a Carrier Packet (CP) is reported hereafter:

```
+-------------------------+
|    CP Payload header    |
+-------------------------+
|       CP Payload        |
+-------------------------+
|      CP Path state      |
+-------------------------+
```

"Interest CIU" are used by End-Nodes to request for content. The Interest CIUs contain the identifier of the content called ICN-ID and transported within the IP CONET Option. Optionally, the IP CONET option can explicitly carry a "Chunk Sequence Number" to identify one of the chunk in which a content has been split. Another possibility is that the chunk number is carried within the ICN-ID itself. A Serving Node or a CONET node that had previously cached the information can reply to the content request by sending a "Named-data CIU". These Named-data CIU will also contain the ICN-ID in the IP CONET Option.

The CP payload header contains the length of the CP Payload and allows to identify the start of the CP Path state field. The use of CP Path state field was explained in [I-D.CONET] and is out of scope here.

The information contained in the CP Payload header is specific for each CIU type. The information transported in the CP Payload header can be used to implement the functionalities of a transport protocol, providing a reliable transmission of content and performing congestion control. In this document we define the CP Payload header for Interest and Data CIUc using the ICTP protocol.

An end-node that wants to retrieve a content (or better a Chunk of a content) issues an Interest CIU, the ICN-ID and (optionally) the Chunk Sequence Number of the required Content are respectively transported in the ICN Identifier (ICN-ID) field and in the CSN field of the CONET IP option. Assuming for simplicity that the Interest CIU will fit into a single Carrier Packet, the Interest CIU will be included in the Carrier Packet that in turn is inserted into an IP packet. The ICTP comes into play because a Chunk of content may need to be fragmented in more than one Carrier Packet, as the chunk size can be much larger than the layer 2 MTU. For example with a chunk size of 64 KB or 256 KB and an MTU of 1500 bytes, the chunks need to be split in tens or hundreds of packets. In this case the RDTP allows to specify in the request the specific segment of the chunk that is required.

3. ICTP Data Structures

3.1. Interest CIU Payload Header

The structure of the interest CP payload header is reported hereafter:

```
+--------------------------+
|TTrrrrrPI| ..Left Edge...|
+--------------------------+
|    ...Right Edge...     |
+--------------------------+
```

Flags:

TT : Transport protocol type. It allows to define different transport protocols. 0 indicates ICTP which is defined in this draft. 1-3 are reserved and can be used to indicate different transport

protocols. The rest of the bits in this field and in the following bytes may have a different semantic depending on the transport protocol, we are providing here only the defintion for TT=0 i.e. the ICTP protocol. Therefore the number of transport protocols is NOT limited to 4.

P : Prefetch flag - This flag indicates that this packet comes from a receiver that asks to perform prefetch on the content chunks.

I : Ask Chunk Info flag - If this flag is set, the serving node is requested to add chunk-related information to the data CIU payload, particularly the chunk size.

Left edge/Right edge : These fields contain respectively the value of first and the last byte of the requested chunk segment. The fields are encoded with the EVLE (Efficient Variable Lenght Encoding) mechanim described below

3.2. DATA CIU Payload Header

The structure of the data CP payload header is reported hereafter:

```
+-------------------------+
|TTrFSSSS| ...Left Edge...|
+-------------------------+
|    ...Right Edge...     |
+-------------------------+
| (Optional) Chunk size   |
+-------------------------+
```

Flags:

TT : Transport protocol type. It allows to define different transport protocols. 0 indicates ICTP which is defined in this draft. 1-3 are reserved and can be used to indicate different transport protocols. The same considerations apply that have been reported above for the TT subfield in the interest CIU payload header.

F : Final segment flag - If this bit is set to 1 this carrier packet carries the last segment of a chunk.

SSSS : Chunk size flag - This flag describes the size of a chunk as follows:

```
      0 : unspecified (it may have been already indicated
          in a previous data)
      1 : the chunk size is trasnported in the optional
          Chunk size field, encoded with the EVLE variable length
          encoding described below
      2-16 : let n be the value from 2 to 16, it can represent
             14 different chunk sizes from 2KBytes to 8Mbyte
             with the following relation:
             chunk size = 2 ^ (9+n)
```

Left edge/Right edge : These fields contains respectively the value of first and the last byte of the transported chunk segment. The fields are encoded with EVLE variable length encoding described below. These fields carry the actual value of the segment transported that may differ from the request.

3.3. EVLE Efficient Variable Length Encoding

Some of the fields described above are encoded using a variable lenght encoding that we denote as "Efficient Variable Lenght Encoding". The same encoding is used for the Chunk Sequence Number (CSN) field in the CONET IP Option, as described in [I-D.CONET]. To help the reader, we report the definition of the encoding hereafter. An EVLE field is represented with a variable number of bytes. An initial bit pattern determines the length of the EVLE field.

```
   1 byte EVLE (7 bits range)
       +--------+
       |0       |
       +--------+

   2 bytes EVLE (15 bit range)
       +--------+--------+
       |10               |
       +--------+--------+

   3 bytes EVLE (21 bit range)
       +--------+--------+--------+
       |110     |        |        |
       +--------+--------+--------+

   4 bytes EVLE (28 bit range)
       +--------+--------+--------+--------+
       |1110    |        |        |        |
       +--------+--------+--------+--------+

   5 bytes EVLE (32 bit range)
       +--------+--------+--------+--------+
       |11110000|        |        |        |
       +--------+--------+--------+--------+
       |        |
       +--------+

   6 bytes EVLE (40 bit range)
       +--------+--------+--------+--------+
       |11110001|        |        |        |
       +--------+--------+--------+--------+
       |        |        |
       +--------+--------+
```

As explained in in [I-D.CONET], binary patterns from 11110010 to 11111111 are reserved and could be used to extend the EVLE range if needed. With the above definion the maximum value is 2^40, roughly 1 Tera.

4. ICTP mechanisms

The transport protocol described in this draft mimics the TPC mechanisms described in [RFC2581], with the required adaptations for a receiver driven approach. In fact, while in TCP the sender sends data segment and implements retransmissions and congestion control based on the reception of ACKs, in ICN based content download the

receiver asks for content sending the Interests and implements re-sending of Interests and congestion control based on the reception of Data.

4.1. Congestion control mechanisms

4.1.1. Fast recovery and fast retransmit

The receiver bases its flow control on the received data CP. The out of sequence recognition is based on the expected chunk number and segment bytes. When three out-of-sequence data CP are received the slow start threshold (ssthresh, see [RFC2581]) is set to half the window plus the 3 data carrier packet already received and the interest for the missing segment is retransmitted.

4.1.2. Slow start and congestion avoidance

To manage the increase of congestion window slow start and congestion avoidance mechanism are performed. During slow start, (e.g. the beginning of the connection), the congestion window increases of an amount equal to the received data CP. This corresponds to an "exponential" growth of the window. During congestion avoidance, the growth of the window is aproximately "linear" with time. Let us define a "segment" as the portion of content transported in a data CP. We define the congestion window cwnd in segments. During slow start cwnd = cwnd + 1 for each received data CP. During congestion avoidance cwnd = cwnd + 1/cwnd for each received data CP.

4.2. ICTP specific mechanisms

4.2.1. Prefetch option

In a ICN based network, we can have applications separately requesting the download of each chunk of content. In our ICTP approach this means that we can only request the segments of a given chunk after that we have received the request for the chunk coming from the application. The application may implement retransmission and congestion control, therefore the ICTP could receive requests for more than a chunk of the same content in parallel. Anyway the interaction between the congestion control performed at application level and the congestion control performed at ICTP level could prove inefficient. Therefore we believe that the API offered to the application should allow the application to request for a whole content (or for the content starting from a given chunk number). Then the ICTP protocol can handle the retrieval of all the (rest of) the content. Therefore the ICTP protocol offers also a "prefetch option". Whit the prefetch option active, if the congestion window has space left, the mechanism allow to issue requests for segments of

next expected chunks, without waiting for an explicit request from application level.

4.2.2. Request chunk information

When the transmission start, the receiver must issue a request for at least the chunk size. Receiver should set the flags in carrier packet accordingly to Section 3.1. The data ciu requested will carry at least the chunk size, receiver should use this information to create data structure best suited for the chunk size.

5. Acknowledgments

We acknowledge the financial support by the EU in the context of the CONVERGENCE research project.

6. Performance Considerations

7. IANA Considerations

This document requires the allocation of one IP option by the IANA.

This document requires the allocation of one IP protocol number by the IANA.

This document requires that IANA will maintain the registry of CONET namespaces.

8. Security Considerations

Security considerations to be provided

9. References

9.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,

June 1999.

9.2. Informative References

[CONET11] A. Detti, et al., "CONET: A Content Centric Inter-Networking Architecture", ACM SIGCOMM Workshop on Information-Centric Networking (ICN-2011), Toronto, Canada , August 2011.

[I-D.CONET] Detti, A., Salsano, S., and N. Blefari-Melazzi, "An IPv4 Option to support Content Networking", draft-detti-conet-ip-option-01 (work in progress), September 2011.

[ICTP12] S. Salsano, et al., "Transport-layer issues in Information Centric Networks", ACM SIGCOMM Workshop on Information-Centric Networking (ICN-2012), Helsinki, Finland , August 2012.

[Jacobson09] V. Jacobson, et al., "Networking named content", Proc. of ACM CoNEXT 2009 , 2009.

[Koponen07] T. Koponen et al., "A data-oriented (and beyond) network architecture", Proc. of ACM SIGCOMM 2007 , 2007.

[RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

Authors' Addresses

Stefano Salsano
Univ. of Rome "Tor Vergata"
Via del Politecnico, 1
Rome 00133
Italy

Email: stefano.salsano@uniroma2.it

Andrea Detti
Univ. of Rome "Tor Vergata"
Via del Politecnico, 1
Rome 00133
Italy

Email: andrea.detti@uniroma2.it

Nicola Blefari-Melazzi
Univ. of Rome "Tor Vergata"
Via del Politecnico, 1
Rome 00133
Italy

Email: blefari@uniroma2.it

Matteo Cancellieri
Univ. of Rome "Tor Vergata"
Via del Politecnico, 1
Rome 00133
Italy

Email: matteo.cancellieri@gmail.com