Network Working Group
Internet-Draft
Intended status: Informational
Expires: 21 November 2020

L. Muscariello
G. Carofiglio
Cisco Systems Inc.
20 May 2020

Transport Framework for Hybrid Information-Centric Networking Architecture
draft-muscariello-hicn-transport-latest

Abstract

Information-Centric Networking (ICN) architectures introduce a location-independent communication model based on request/reply exchanges between a client and one or more a-priori unknown sources (ref to RFCs). Request packets carry a data indentifier which is not bound to any specific host location, hence is not affected by network mobility. They are routed in a hop-by-hop fashion over one or multiple paths until they are matched by requested data packets. Data packets follow the reverse path of requests to be conveyed to one or more requesting clients.

Mobility, Multi-path, Multi-source and Many-to-many communication (M4) characterize ICN transport making it a promising candidate for a various number of applications (e.g. live-streaming, IoT, mobile edge computing, realtime multimedia communications, etc.) The requirements of native M4 transport are not met by traditional host-to-host transport protocols which assume the knowledge of both client and server end-points before packet transmission and do not leverage dynamic forwarding/caching/control decisions taken by in-path nodes.

There have been various attempts in the literature to either adapt existing transport protocols to ICN or to define novel approaches, in all cases targeting a specific application (refs to papers) A few attemtps have been made at addressing the novel challenges brought by ICN in terms of flow definition, path labelling, etc., but to the best of our knowledge no unified ICN transport framework has been proposed so far. Without the ambition to design a specific ICN transport protocol, this document discusses a general M4 transport framework for Hybrid ICN, an IP-integrated open-source ICN architecture (ref to FD.io hICN) and illustrates its applicability to a few relevant use cases.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft will expire on 21 November 2020.

Table of Contents

1. Introduction

1.1. Problem and Scope

Information-Centric Networking (ICN) identifies a networking paradigm centering communication around named data, in contrast to host location. ICN offers a simplified and more efficient user-to-content communication that resolves the mismatch between content-centric usage and underlying host-to-host foundations of the Internet.

Despite important architectural differences in the proposed designs (REFS), a shared idea characterizes ICN, that of abstracting data delivery from underlying point to point connectivity by means of an information-aware network layer operating directly on secured named data regardless of their location. It results a native support for mobility, storage and security as network features are integrated by design, rather than as an afterthought.

Receiver-Driven Connectionless Transport - In contrast with the current sender-based TCP/IP model, ICN transport is receiver-controlled, does not require connection instantiation and accommodates retrieval from possibly multiple dynamically discovered sources. It builds upon the flow balance principle guaranteeing corresponding request-data flows on a hop-by-hop basis [9]. A large body of work has looked into ICN transport (surveyed in [78]), not only to propose rate and congestion control mechanisms [81, 100] - especially in the multipath case [30] - but also to highlight the interaction with in-network caching [29], the coupling with request routing [30, 50], and the new opportunities provided by in-network hop-by-hop rate/loss/congestion control [32, 93] for a more reactive low latency response of involved network nodes.

* Rephrase around M4 transport capabilities

1.2. Related work

Rel work : a summary There have been various attempts in the literature to either adapt existing transport protocols to ICN or to define novel approaches, in all cases targeting a specific application (refs to papers)

Rel work on ICN transport protocols

There has been huge amount of work on transport protocolsin the Internet. Here we focus on transport schemes proposedfor CCN. We give a discussion of such work in three aspects.RTT-based congestion control:Some designs, for instanceICP [3] and ICTP [19], rely on a single round trip time (RTT)estimator at the receiver to predict

network status which arenot suitable for CCN because CCN flows may have multiplesources and multiple paths. To adapt to CCN's multipathnature, authors in [5], [6] propose per-route transport control,in which a separate RTT estimation is maintained for eachroute at the receiver, similar to MPTCP [21]. Multiple RTTbased scheme works well for multipath transfer, however, itadds lots of complexity to the receiver and heavily relies onthe accuracy of timing. Our scheme, on the other hand, utilizesexplicit congestion signalling from network to effectivelynotify the receiver about network condition. Compared withmultiple RTT based scheme, our approach leads to a muchsimpler receiver design and is not restricted to limitations oftimers which have long been criticized [9], [11], [18], [24].hop-by-hop Interest shaping:Since in CCN one Interestpacket retrieves at most one Data chunk, a router can controlthe rate of future incoming data chunks by shaping the rate ofoutgoing Interests. In [4], [17], authors propose quota-basedInterest shaping scheme to actively control traffic volume.They assign a quota (in terms of the number of pendingInterests) to each flow, and if the number of pending Interestsfor a flow exceeds the quota, that flow's Interests will bedelayed or dropped. In [22], authors use per interface rate limitto avoid congestion at an interface and per prefix-interfaceratelimit to control Interest rate of each content prefix. The quota-based Interest shaping and the rate limiting Interest controlrequire to assign an appropriate quota value for each flowor rate limit value for each content, which is challenging in practice, if not impossible. And they can be rather inefficientunder dynamic traffic setting. Our fair share Interest shapingscheme also considers about fairness, however, resources areshared by all flows and Interest from a flow is delayedtemporarily only when shared resources become limited andthe corresponding flow unfairly consumes resources.flow-aware traffic control:Authors in [17] propose a flow-aware network paradigm for CCN. They define content flowas packets bearing the same content name identified on the flyby parsing the packet headers. Moreover, routers impose per-flow fair bandwidth sharing by having multiple data queues ateach interface for active flows and using deficit round robin(DRR) [20] for fair scheduling among these queues. Differentfrom such scheme, our FISP realizes per-flow fair sharing byhaving per-flow Interest queues at an interface and utilizingmodified DRR to approximate max-min fairness.

The Information Centric Transport Protocol (ICTP) [REF]adapts the existing ideas of CCNx. ICTP implements thesame algorithm of TCP in its congestion control mechanism,adapting it to the receiver-driven context. In particular, itoperates using interest-data sequences. The loss detection relieson the expected set of data. An equivalent of duplicate ACKin ICTP is the out-of-order received segments, i.e. a gap inthe flow of data. In NetInf TP the order of reception is non-

sequential and loss detection relies on the number of receivedsegments after a particular request has been issued.Carofiglioet al.propose the Interest Control Protocol (ICP)as a receiver-driven transport protocol for CCN [REF]. ICPimplements a window-based interest flow control mechanism.For reliability, ICP relies on delay measurements and timerexpirations only; this means, retransmissions are realized byre-issuing interest packets once the timers expire. This alsomeans that re-expression of an interest can happen even if apacket is only delayed. The authors used an analytic modelwith parameter setting to optimize the round-trip timers. Incontrast, NetInf TP uses an additional packet loss detectionmechanism that signals congestion more reliably.

NetInf TP [REF] applies a cyclic retransmission strategy which savesresources in congested scenarios.

A few attemtps have been made at addressing the novel challenges brought by ICN in terms of flow definition, path labelling, etc., but to the best of our knowledge no unified ICN transport framework has been proposed so far.

Rel work on new challenges brought by ICN transport

* Flow definition

* Path identification

* Out-of-order delivery or variations in inter-arrival inter-vals can no longer be used as an indication of networkcongestion. In fact, a packet might arrive out of ordersimply because it has been sent by a node located furtheraway than the source of the other packets.

* RTO estimation, as defined by Jacobson's algorithm [6],does not span multiple data sources, and as such it cannotbe used reliably.

1.3. Reference ICN architecture: hICN

Focus here is on defining a transport frameworkf for Hybrid ICN (REF to FD.io hICN)

We focus on hICN (refs) and on the open-sourced implementation in FD.io hICN project (ref)

Hybrid ICN is an architecture that brings ICN into IPv6 as described in [1]. By doing that, hicn allows to generalize IPv6 networking by using location-independent name-based networking. This is made

either at the network layer and at the transport layer by also providing name-based sockets to applications.

hicn allows to reuse existing IPv6 protocol and architectures, to extend them and deploy hybrid solutions based on the use case and application needs. Moreover, hicn can exploit hardware and software implementations heavily based on IP, making much simpler insertion of the technology in current networks, current applications and design future applications reusing what the industry already provides in terms of on the shelf components.

1.4. Terminology

1.5. Notational Conventions

The words "MUST", "MUST NOT", "SHOULD", and "MAY" are used in this document. It's not shouting; when these words are capitalized, they have a special meaning as defined in [RFC2119].

2. Transport framework architecture

M4 Transport framework provides segmentation and reassembly services to the applications, and sits on top of hICN network layer which provides a request/reply service to the transport layer itself. End-points in TCP/IP can be described as acting as transmitters or receivers, or both at the same time, whereas hICN transport is oriented to the consumer/producer paradigm where a transmission is always triggered by a matching request. Moreover, a TCP connection is a bidirectional communication channel, while in hICN there are always two unidirectional sockets: the consumer and the producer. Transport protocols in hICN can be stream or datagram oriented, depending on the application (as TCP or UDP). Currently, hICN prototype implements (i) the stream oriented ICN multi-path transport protocol with support for loss recovery, flow and congestion control introduced in \cite{carofiglio2016optimal} and (ii) the in-network loss control mechanisms in \cite{Papalini-WLDR}. We plan to enrich in the future the set of supported transport protocols. As previously mentioned, a transport manifest is used by the producer to distribute meta-data useful for consumer to retrieve the actual data. It includes low level segmentation information (name suffixes) as well as integrity information, and it is signed with the producer's private key~\cite{Manifest}. Such approach is adopted as the default \hicn{} approach instead of per packet signature computation. The manifest also carries the expiration time of the associated data which is an absolute quantity and requires reasonable synchronization to UTC. In case the manifest is not used by the producer, the expiration time is carried by the TCP timestamp option.

\subsection{Socket API}\label{subsec:socket-api} We use substantially unaltered the INET6 socket API for \hicn{} with the twofold advantage to provide a known API for developers of new applications and simple integration of existing ones. The system calls of \hicn{} socket API are based on the socket interface extensions for IPv6 \cite{rfc3493}. Both consumer and producer sockets bind to a socket address (\sa{}), which is initialized by specifying address family \af and name prefix. The range of addresses stated by the name prefix tells the namespace in which a producer socket is allowed to publish data and a consumer socket is allowed to request data.

The \texttt{bind()} system call takes care of setting up a local face to the \hicn{} forwarder, which in the case of the producer also sets a FIB entry \texttt{(name_prefix, socket_id)}. The \texttt{recvmsg()} and the \texttt{recvfrom()} system calls are used by a consumer for retrieving data, while \texttt{sendmsg()} and \texttt{sendto()} are used by a producer for publishing a content and making it available for the consumers. After data is published under a given name, subsequent requests for it can be satisfied by the producer socket without passing them to the application.

The consumer socket can use the \texttt{sendmsg()} for sending a single Interest with payload eventually triggering a \texttt{recvmsg()} on the remote producer to pass the payload to the application, e.g. to send and HTTP request in half RTT. In general, push semantics are realized using reverse pull techniques, which require all end-points to have a consumer and a producer socket open and bound to a valid name prefix. An end-point can push data to one or several remote end-points by (i) calling a \texttt{sendto()} from the producer socket with actual data; (ii) passing the resulting transport manifest to a valid open consumer socket; (iii) calling a \texttt{sendmsg()} from the consumer to trigger an Interest packet carrying the data manifest as payload. The remote producer (one or many) after reception of Interest can pass the manifest to the local consumer socket to trigger the reverse pull procedure. This technique can be used to send large HTTP requests to one or multiple servers, and has the advantage to enable multicast upload from one client to many servers.

2.1. Application APIs

Application requirements to tailor underlying transport ## Consumer and Producer Sockets We use substantially unaltered the INET6 socket API for \hicn{} with the twofold advantage to provide a known API for developers of new applications and simple integration of existing ones. The system calls of \hicn{} socket API are based on the socket interface extensions for IPv6 \cite{rfc3493}. Both consumer and producer sockets bind to a socket address (\sa{}), which is

initialized by specifying address family \af and name prefix. The range of addresses stated by the name prefix tells the namespace in which a producer socket is allowed to publish data and a consumer socket is allowed to request data.

The \texttt{bind()} system call takes care of setting up a local face to the \hicn{} forwarder, which in the case of the producer also sets a FIB entry \texttt{(name_prefix, socket_id)}. The \texttt{recvmsg()} and the \texttt{recvfrom()} system calls are used by a consumer for retrieving data, while \texttt{sendmsg()} and \texttt{sendto()} are used by a producer for publishing a content and making it available for the consumers. After data is published under a given name, subsequent requests for it can be satisfied by the producer socket without passing them to the application.

The consumer socket can use the \texttt{sendmsg()} for sending a single Interest with payload eventually triggering a \texttt{recvmsg()} on the remote producer to pass the payload to the application, e.g. to send and HTTP request in half RTT. In general, push semantics are realized using reverse pull techniques, which require all end-points to have a consumer and a producer socket open and bound to a valid name prefix. An end-point can push data to one or several remote end-points by (i) calling a \texttt{sendto()} from the producer socket with actual data; (ii) passing the resulting transport manifest to a valid open consumer socket; (iii) calling a \texttt{sendmsg()} from the consumer to trigger an Interest packet carrying the data manifest as payload. The remote producer (one or many) after reception of Interest can pass the manifest to the local consumer socket to trigger the reverse pull procedure. This technique can be used to send large HTTP requests to one or multiple servers, and has the advantage to enable multicast upload from one client to many servers.

* Transport semantics in L4 packet header

2.2. Flow identification

... ## Rate and congestion control at the receiver AIMD-based receiver Interest control(RIC). The receiveradjusts its Interest window based on the AIMD (AdditiveIncrease Multiplicative Decrease) mechanism. Here, re-ceiver detects congestion mostly by marked packets fromupstream routers.

RAAQM paragraph

Other proposals

2.3. Loss detection and control

at receiver and in the network

WLDR, MLDR

Interest shaping and control hop-by-hop

3. Coexistance with non-ICN transport

4. Examples of application

Live-streaming video distribution ## Realtime Communications

5. Security considerations

We also envision for future work, the design of a secure transport layer that provides confidentiality to the communication by extending the widely diffused point-to-point approach (e.g., as in TLS and DTLS), by means of secure group communications~\cite{baugher2005multicast,perrig1999efficient,rafaeli2003survey,sherman2003key} to cope and exploit \hicn{} in-network caching and multicasting.

5.1. Consumer

Producer

6. References

6.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC1081] Rose, M.T., "Post Office Protocol: Version 3", RFC 1081, DOI 10.17487/RFC1081, November 1988, <https://www.rfc-editor.org/info/rfc1081>.

[RFC1624] Rijsinghani, A., Ed., "Computation of the Internet Checksum via Incremental Update", RFC 1624, DOI 10.17487/RFC1624, May 1994, <https://www.rfc-editor.org/info/rfc1624>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, DOI 10.17487/RFC3587, August 2003, <https://www.rfc-editor.org/info/rfc3587>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

6.2. Informative References

[CCN] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and R. Braynard, "Networking named content", DOI 10.1145/1658939.1658941, Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09, 2009, <https://doi.org/10.1145/1658939.1658941>.

[I-D.irtf-icnrg-ccnxmessages] Mosko, M., Solis, I., and C. Wood, "CCNx Messages in TLV Format", Work in Progress, Internet-Draft, draft-irtf-icnrg-ccnxmessages-09, 24 January 2019, <http://www.ietf.org/internet-drafts/draft-irtf-icnrg-ccnxmessages-09.txt>.

[I-D.irtf-icnrg-ccnxsemantics]
Mosko, M., Solis, I., and C. Wood, "CCNx Semantics", Work in Progress, Internet-Draft, draft-irtf-icnrg-ccnxsemantics-10, 24 January 2019, <http://www.ietf.org/internet-drafts/draft-irtf-icnrg-ccnxsemantics-10.txt>.

[I-D.irtf-icnrg-mapme]
Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "MAP-Me : Managing Anchorless Mobility in Content Centric Networking", Work in Progress, Internet-Draft, draft-irtf-icnrg-mapme-04, 3 November 2019, <http://www.ietf.org/internet-drafts/draft-irtf-icnrg-mapme-04.txt>.

[I-D.irtf-icnrg-terminology]
Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): CCNx and NDN Terminology", Work in Progress, Internet-Draft, draft-irtf-icnrg-terminology-08, 17 January 2020, <http://www.ietf.org/internet-drafts/draft-irtf-icnrg-terminology-08.txt>.

[I-D.muscariello-intarea-hicn]
Muscariello, L., Carofiglio, G., Auge, J., and M. Papalini, "Hybrid Information-Centric Networking", Work in Progress, Internet-Draft, draft-muscariello-intarea-hicn-03, 30 October 2019, <http://www.ietf.org/internet-drafts/draft-muscariello-intarea-hicn-03.txt>.

[MAN]
Baugher, M., Davie, B., Narayanan, A., and D. Oran, "Self-verifying names for read-only named data", DOI 10.1109/infcomw.2012.6193505, 2012 Proceedings IEEE INFOCOM Workshops, March 2012, <https://doi.org/10.1109/infcomw.2012.6193505>.

[MIR]
Garcia-Luna-Aceves, J., Martinez-Castillo, J., and R. Menchaca-Mendez, "Routing to Multi-Instantiated Destinations: Principles, Practice, and Applications", DOI 10.1109/tmc.2017.2734658, IEEE Transactions on Mobile Computing Vol. 17, pp. 1696-1709, July 2018, <https://doi.org/10.1109/tmc.2017.2734658>.

[NDN]
Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., claffy, k., Crowley, P., Papadopoulos, C., Wang, L., and B. Zhang, "Named data networking", DOI 10.1145/2656877.2656887, ACM SIGCOMM Computer Communication Review Vol. 44, pp. 66-73, July 2014, <https://doi.org/10.1145/2656877.2656887>.

[RAQ] Carofiglio, G., Gallo, M., Muscariello, L., Papalini, M., and S. Wang, "Optimal multipath congestion control and request forwarding in Information-Centric Networks", DOI 10.1109/icnp.2013.6733576, 2013 21st IEEE International Conference on Network Protocols (ICNP), October 2013, <https://doi.org/10.1109/icnp.2013.6733576>.

[WLD] Carofiglio, G., Muscariello, L., Papalini, M., Rozhnova, N., and X. Zeng, "Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks", DOI 10.1145/2984356.2984361, Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking - ACM-ICN '16, 2016, <https://doi.org/10.1145/2984356.2984361>.

Authors' Addresses

Luca Muscariello
Cisco Systems Inc.

Email: lumuscar@cisco.com

Giovanna Carofiglio
Cisco Systems Inc.

Email: gcarofig@cisco.com