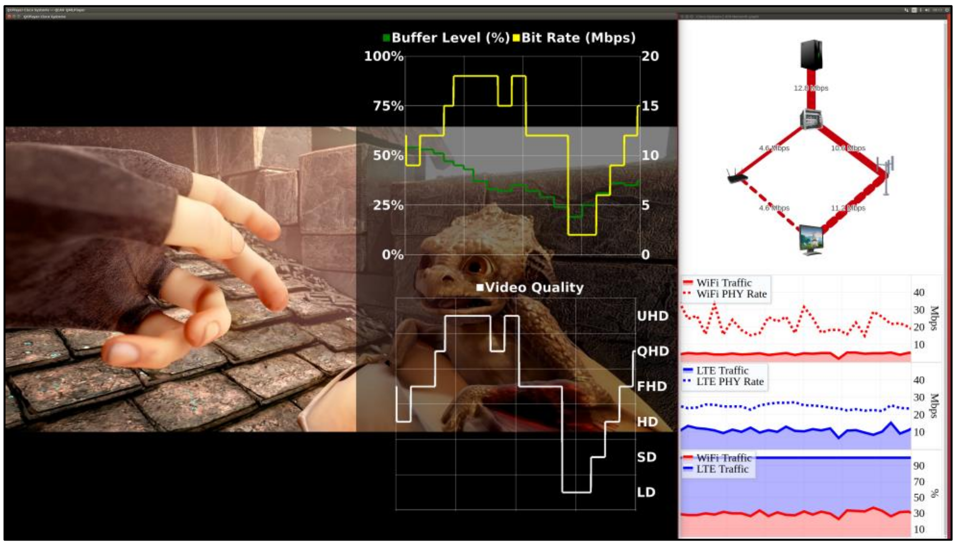
Mobile Video Delivery with Hybrid ICN

IP-integrated ICN solution for 5G



ICN communication paradigm presents numerous advantages in 5G context: anchorless mobility support, access-agnostic transport with native multipath support, unified unicast/multicast, edge-embedded caching/processing capabilities and flexible object-based security to cite the most important ones.

In this document, we define hybrid ICN (hICN), an incremental deployment strategy for ICN, and introduce a video-centric architecture where ICN communication principles are integrated in IP and optimized for Dynamic Adaptive Streaming end-to-end, from mobile devices connected over heterogeneous mobile access to an ICN-enabled backhaul/core infrastructure. Benefits are showcased both at mobile clients, due to enhanced rate adaptation and dynamic packet-granular load-balancing over multiple accesses, and in the network, as resulting from reactive caching and request forwarding, opportunistic multicast and dynamic network-assisted video rate adaptation.

The Proof of Concept leverages a virtualized hICN architecture at scale, integrating physical ICN-enabled devices with containerized high speed ICN nodes based on VPP.

Introduction

Mobile video to drive 5G

There is no doubt about video, and especially mobile video predominance in future traffic trends. By 2020, 82% of all IP traffic will be video and two-third of all Internet traffic will be generated from wireless and mobile devices, according to Cisco VNI forecast [1], the latter trend supported by heterogeneous and high speed 5G wireless access. Traffic growth goes hand in hand with evolving video services (e.g., UHD 4K-8K video, Virtual/Augmented Reality), driving future 5G networks design to meet new mobile video usages with very-high bandwidth requirements under ultra-low latency constraints.

In parallel, video consumption is changing: a consistent move has been observed in the last Rio 2016 Olympic games toward online viewing with an explosion of mobile and social platforms reaching millions of people. *"The first six days of NBC's Rio coverage generated 153.8 million "social media engagements", 10 times larger than the total for NCAA Basketball March Madness (19 days) and outpaced the 32-day total for the entire 2014 soccer World Cup in Brazil"* (NBC source). As a sign of change in video consumption, it has been observed less TV more connected devices usage, less broadcast more streaming, with a larger impact on network end-to-end from the access to the core.

...is video going to break the network?

All these factors put pressure on the capabilities of future 5G networks and highlight their critical role in the support of Dynamic Adaptive Streaming (DAS). With DAS, we refer here to the variety of techniques, in most of the cases relying on HTTP, that have bloomed in the last years to realize an efficient multimedia delivery over the Internet: many popular ones are proprietary (e.g., Apple HLS, Microsoft HSS), while Dynamic Adaptive Streaming over HTTP (DASH) has recently become a standard. Since DAS techniques were initially designed for CDN/OTT content delivery, their interaction with the network has been only superficially studied so far. In the 5G mobile and heterogeneous network access, it seems of utmost importance to consider DAS interaction with the network and to move caching and computing capabilities to the network edge in order to enable efficient mobile video delivery.

A scalable ICN as a natural answer

Information-Centric Networking (ICN) appears as a natural answer to support the evolution of video delivery by empowering the network with content-aware capabilities for a joint video/network optimization. ICN communication paradigm is based on location-independent network names and a named-based connectionless transport exploiting network-level caching, multi-path forwarding and seamless mobility support – features that are all very appealing for DAS systems, especially in a mobile environment.

The potential for ICN application in adaptive streaming services as an alternative to relieve from some of the recognized inefficiencies of standard TCP/IP transport has been only partially explored: initially hints have been given on the potential benefits of built-in caching and name-based forwarding to assist DAS rate adaptation inside the network, rather than only at the client side, but a feasible deployment path for insertion of ICN principles into today's networks has not been proposed yet.

To comprehensively test ICN potential for mobile video delivery, we in Cisco have worked on two directions: the design of hICN, an incremental deployment strategy for ICN that preserves all its benefits while integrating ICN into the existing IP infrastructure and the definition of a video-centric ICN architecture to support mobile video delivery over a heterogeneous mobile access.

In this paper, we introduce a virtualized video-optimized hICN architecture with containerized ICN router instances from the access to the network core, and physical or emulated mobile devices retrieving 4K video using DAS over a heterogeneous Wi-Fi/LTE access. We illustrate benefits coming from enhanced video rate adaptation, in-network loss recovery, dynamic load balancing, multicast and caching.

[1] "The Zettabyte Era" White Paper, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.

Key ICN features and advantages for 5G

ICN identifies a new networking paradigm centering network communication around named data, rather that host location. Network operations are driven by location-independent content names, rather than location identifiers (IP addresses) to gracefully enable user-to-content communication. There exist a few proposals (e.g. CCN or NDN) sharing the same set of core principles:

- **Named-Data** Information is addressed by **location-independent identifiers** and network operations (forwarding, caching, transport, security) are bound to named-data, not location.
- **Enhanced Transport**: in contrast with the current sender-based TCP/IP model, ICN transport is **pull-based** (i.e. data is triggered by per-packet user requests), **connectionless and natively multipath** (no connection instantiation, retrieval from possibly multiple a-priori unknown sources)**,** and not bounded to a network addressable interface.
- **Dynamic Forwarding** the name-based data plane is **stateful**: user requests are routed by named and a trace of pending requests is left to guarantee reverse path routing of corresponding data, to enable aggregation (**synchronous multicast**) and to drive forwarding strategies (based on popularity and on network status).
- **In-path Caching/Processing** Packet forwarding is enriched with **in-path buffering and processing** capabilities. In path buffering is exploited for re-use (**asynchronous multicast** of data via cached replica) and repair (in-network rate/congestion control).
- **Object-based Security** decoupling authenticity, integrity and confidentiality, so that a more flexible and application-centric approach can be decided with no modification of the underlying connectionless transport coupled with in-path caching.

As a result of such principles, ICN natively supports mobility, storage and security in the network.

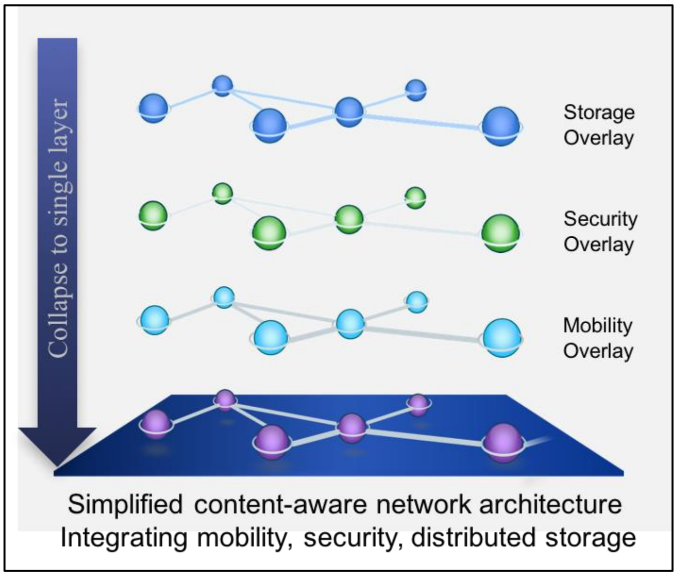


Fig.1 – ICN simplifies architecture

Mobility Model

In the ICN architecture, interfaces do not have network addresses, so a change in physical location does not imply a change of address in the data plane. Support for consumer mobility emerges naturally from the architecture because of the connectionless, symmetric transport model. With ICN's pull-based communication model, the consumer expresses interest packets that are routed in the network toward the data. The data is returned toward the client following the paths traversed by the interests. In a case of a move before data in flight is received, the consumer may simply re-express the interests for those data objects. The network may now be able to fetch it from local caches filled by the data in flight.

Producer mobility and real-time group communication are more challenging to support, depending on the frequency of the mobility and on the content lifetime. Again, the basic interest/data exchange mechanisms provide a means to rapidly update local forwarding tables to ensure continued reachability of mobile content. The distributed in-network caching of ICN allows to smooth handoffs and to prevent service quality degradation.

We, in Cisco, have developed an anchor-less mobility management model [2][3], addressing producer mobility, even in presence of latency-sensitive applications. The rationale behind is to exploit ICN features like stateful forwarding, dynamic and distributed Interest load-balancing and in-network caching to design a timely forwarding update mechanism at routers, relaying former and current producer locations.

The protocol does not rely on global routing updates which would be too slow and too costly, rather works at a faster timescale propagating forwarding updates and leveraging real-time notifications left as breadcrumbs by the producer to enable live tracking of its position.

Storage Model

ICN nodes temporarily store content items in order to serve future requests for the same content. Whenever an interest is received at an ICN node, it first checks if the requested data are present in the local cache. If so, the content is returned to the user. Otherwise, the request is forwarded to the next hop by the ICN request routing. In-network caching allows the network to exploit current buffers in routers, possibly enhanced by additional memory blocks, as intermediate caches.

The content-awareness provided by names to network nodes enables a different use of buffers, not only to absorb input/output rate unbalance but for temporary caching of in-transit data packets. Even without additional storage capabilities in routers, the information access by name of ICN allows two new uses of in-network wire-speed storage:

- **Reuse**. Subsequent requests for the same data can be served locally with no need to fetch data from the original server/repository.
- **Repair**. Packet losses can be recovered in the network, with no need for the sender to identify and retransmit the lost packet (cfr.[4] for more details on the functions we implement here).

Simple cache management policies and coordination techniques allow an efficient allocation of distributed in-network storage resources at very low computational overhead and without requiring the complex, often centralized, management of today's CDN.

The presence of distributed in-network storage and of name-based lookup automatically distributes copies of popular content closer to the users as demand materializes.

[2] J.Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, X. Zeng, *MAP-Me: Managing Anchor-less Producer Mobility in ICN, under submission, accessible at* http://arxiv.org/abs/1611.06785.
[3] J.Augé, G. Carofiglio, L. Muscariello, Cisco MWC'16 demo, https://www.youtube.com/watch?v=p26GODPxGGE
[4] N.Rozhnova, G.Carofiglio, L.Muscariello, M.Papalini, *Leveraging ICN in-network Control for Loss Detection and Recovery in Wireless Mobile Networks ,* in Proc. of ACM ICN 2016, Kyoto, September 2016.

Security Model

Current Internet security is made available by means of ad-hoc protocol extensions such as DNSsec, IPsec and TLS. TLS provides web security by encrypting a layer 4 connection between two hosts. Authenticity is provided by the web of trust (certification authorities and a public key infrastructure) to authenticate the web server and symmetric cypher on the two end points based on a negotiated key. In presence of TLS, many networking operations become unfeasible, including filtering, caching, acceleration and transcoding. ICN security model is radically different. Instead of securing by encrypting simply connections, the ICN object-security model allows the separation of security actions regarding privacy, data integrity and data confidentiality, all of which leverage an existing web of trust based on certification authorities and a public key infrastructure. The security actions are performed directly at network layer with content identification provided in data names. All data is integrity protected, whereas confidentiality (via data encryption) is optional. Integrity protection guarantees the authenticity of the data bound to the name by including the producer signature of the data plus its name.

The atomic security service provided by ICN guarantees that the producer has published a piece of data with the name available in the packet. This service enables location-independent secured content access. Denial-of-service attacks based on cache poisoning can be blocked using signature verification techniques. However, the cost is not negligible, and some recent work has started to build network layer trust management that does not required in-network signature verification by using interest-key binding.

Advantages for 5G

ICN appears as a promising networking technology candidate for 5G in view of the potential advantages that are associated to the key distinguishing features of its communication paradigm. To summarize the main ones:

1. **Simplified core network architecture**
   via built-in access-agnostic mobility support that does not require mobility anchors nor control-plane signaling to maintain connectivity under content/network mobility

2. **Seamless communication over mobile hetnet access**
   via connectionless receiver-driven transport that natively leverages multiple paths/sources not known a priori and dynamic load-balancing capabilities at every network node

3. **Latency-reduction**
   via in-network control (e.g. wireless/mobility/congestion loss detection and recovery) and hop-by-hop dynamic forwarding strategies minimizing per-content latency.

4. **Better user experience with transport cost reduction**
   via application-centric edge caching/processing policies (e.g. video specific) and unified unicast/multicast communication model with no need for pre-configuration/ flow synchronization

5. **Improved security/confidentiality**
   via object-based security : same approach supports different application requirements today incompatible (eg TLS-like confidentiality with caching)

6. **Richer network-aware content analytics**
   to optimize service delivery and enable new services leveraging content/network adherence

Hybrid ICN: an incremental deployment strategy for ICN

The definition of an incremental deployment solution for Information-Centric Networking (ICN) into existing IP networks is of crucial importance for ICN introduction, even in the long-term perspective of a wholesale replacement of IP as the inter-networking layer of the Internet. There are proposals of overlay approaches for the deployment of ICN over IP: the main disadvantage is that they require the definition and standardization of a new packet format and of protocols to manage the correspondence between ICN and IP layers. Other efforts have looked into the possibility of a partial integration of ICN semantics into IP at the cost of modifying ICN behavior and thus trading off ICN benefits in favor of IP compatibility. We, in Cisco, have designed Hybrid ICN (hICN), a solution for deployment of ICN inside IP, rather than over IP, that preserves all ICN features of ICN, while mapping names into IP addresses.

Hybrid ICN (hICN)

uses IPv4 or IPv6 RFC compliant packet formats, guarantees transparent interconnection of (a) a standard IPv4 or IPv6 router and (b) a hybrid ICN-IP router (hICN) that processes and forwards both regular IP packets in the standard way, and IP packets with an ICN semantic according to the typical ICN forwarding pipeline preserves pure ICN behavior at layer 3 and above (name-based forwarding, routing, connectionless transport, object-based security) by guaranteeing end to end service delivery between data producers and data consumers using ICN communication principles. hICN does not require to predefine adjacencies at the ICN level. In addition, since not all application can benefit from using ICN, this solution allows a selective choice of IP or ICN semantics. A comparison of hICN/ICN features is reported in Tab.1 It highlights that the two aspects hICN and ICN differ are (i) naming due to mapping introduced by hICN of ICN names into IP addresses and (ii) forwarding/routing: with hICN enabling both name-based and standard location-based forwarding over the same IP infrastructure.

IP content networking	hICN	ICN
• Names into IPv6 addresses	• Names into IPv6 addresses	• Variable length routable names
• L4-7 request routing based on names (e.g. with SR)	• L3 Name-based routing and h2h dynamic forwarding	• L3 Name-based routing and h2h dynamic forwarding
• Connection-based sender-driven transport	• Partially symmetric routing	• Symmetric routing
• Tunnel-based security	• Connectionless receiver-driven multipath transport	• Connectionless receiver-driven multipath transport
• Anchor-based mobility	• Object-based security	• Object-based security
• Application-layer (CDN) proactive caching	• Anchorless mobility	• Anchorless mobility
	• In-path reactive caching	• In-path reactive caching

Tab. 1 hICN/ICN characteristics.

Example of hICN communication involving both pure-IP and hybrid IP/ICN nodes (hICN).

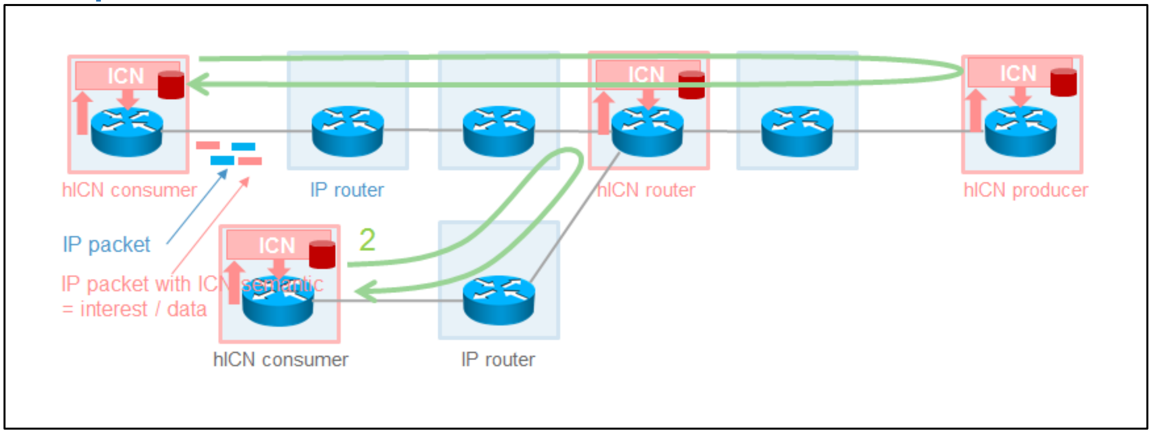


Fig.2 - Example of hybrid architecture

An hICN producer serves content to two hICN consumers (regular IP devices that have been hICN enabled (by installing application plugins). The underlying IP network has been enhanced by upgrading the router in the middle, aggregating traffic, with an hICN forwarding module (for ICN processing).
The operations are unchanged for regular IP traffic. HICN requests (interests) issued by consumers are named using IP addresses (name is put in IP destination address field) and IP-forwarded towards the producer.

Being the hICN interest a regular IP packet, it traverses the two first IP routers unmodified, to reach the intermediate hICN router, where it is ICN-processed and leaves a trace its source IP address and originating face in order to route data back to the consumer. Note that request aggregation is still possible exactly as in pure ICN networks by leveraging the same soft state associated to pending requests.

In case the content is not found in the intermediate cache, the interest is forwarded on the output interface with the same name in IP destination address field and the name of the hICN router as IP source address. Once it reaches the producer, a data packet is sent back with the same name in the IP source address field and the previous hICN router name (encoded as IP address) in IP destination address field. Requests for the same data initiated by the second consumer will terminate at the first hICN junction point where they are answered either from the local cache (asynchronous multicast) or from the data.

Mobile Video architecture with hICN

ICN-enhanced DAS

The ICN-enabled DAS client. To integrate (h)ICN stack at the **client**, we have developed a Qt/QML player for MPEG-DASH standard as well as HLS, in order to support application-level selection of the network stack (i.e., hICN vs TCP/IP), along with state-of-the-art adaptive-bitrate control algorithms, all exploiting packet granular network information coming from hICN data plane.
The ICN-enabled DASH server can stream videos over TCP/IP or ICN directly: it consists of an HTTP server that can use the ICN socket API in addition to standard TCP. Both TCP/IP and hICN stacks serve MPEG-DASH compliant 4K videos.
hICN DAS client/server architecture is reported below and applications take advantage of the ICN socket API made available as a library.

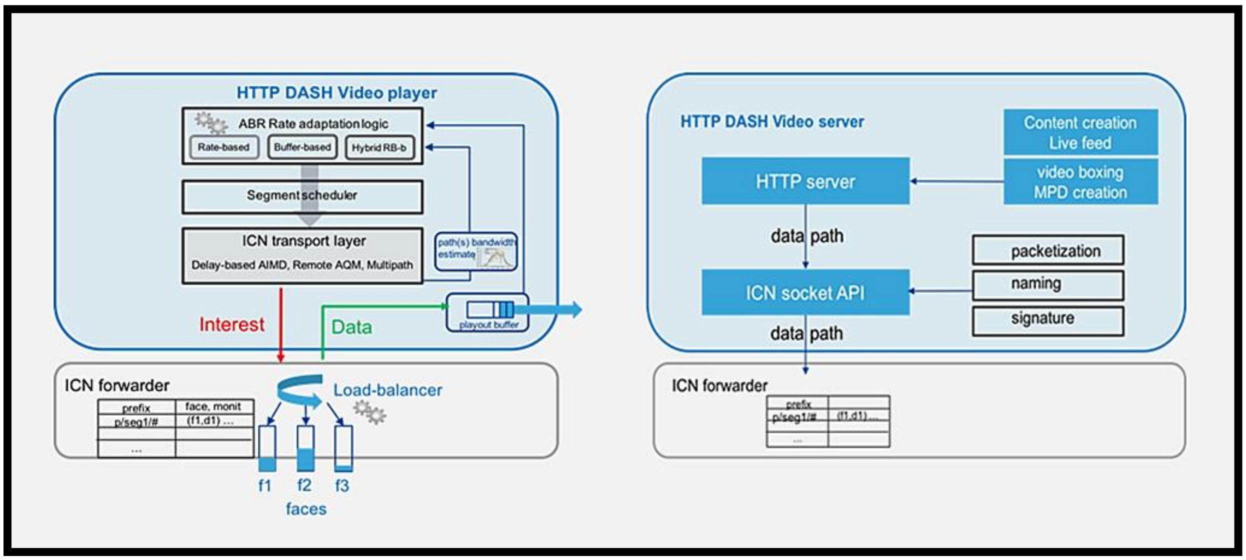


Figure 3: hICN stack

hICN stack

hICN enhancements relate to both video rate adaptation at the client and to intrinsic hICN features (in-network control, multipath transport, dynamic load-balancing, in path caching).

ICN enhancements

MPEG-DASH operates at the application layer, while ICN operates at the network/transport layer. We enhance ABR streaming by leveraging the following ICN network/transport capabilities:

1. **Receiver-based** transport model coherent with client-driven DASH model (less throughput oscillations, higher reactivity),
2. **Fine granular per-packet in-network control and monitoring** to feed rate adaptation logic and to drive dynamic load-balancing
3. **In-network loss detection/recovery** (smaller retransmission delays, especially on wireless access)
4. **Mobility-robust and multipath-capable** transport layer with no knowledge a-priori of sources/paths (seamless communication over heterogeneous access
5. **In-path caching and hop-by-hop forwarding** strategies (leveraging application-specific metrics)
6. **Unified unicast/multicast** communication model enabling both synchronous (via Interest aggregation in PIT) and asynchronous (via in-path caching) multicast opportunistically.

1)-2) Packet-level vs segment-level bandwidth estimation at the receiver: DAS adaptation logic experiences different behaviors according to the used stack: (h)ICN offers a finer-grained estimation of the bandwidth available directly from packet arrivals at receiver, as opposed to the much coarse application-layer estimate done at video-segment granularity with TCP. Not only changing the granularity of TCP estimate is complex (i.e., kernel level modification required), but also making this estimate available to the client is not trivial (i.e., estimate is available at server side, which would require specific protocols for piggybacking this information). (h)ICN increases reactivity to take rate selection decisions.

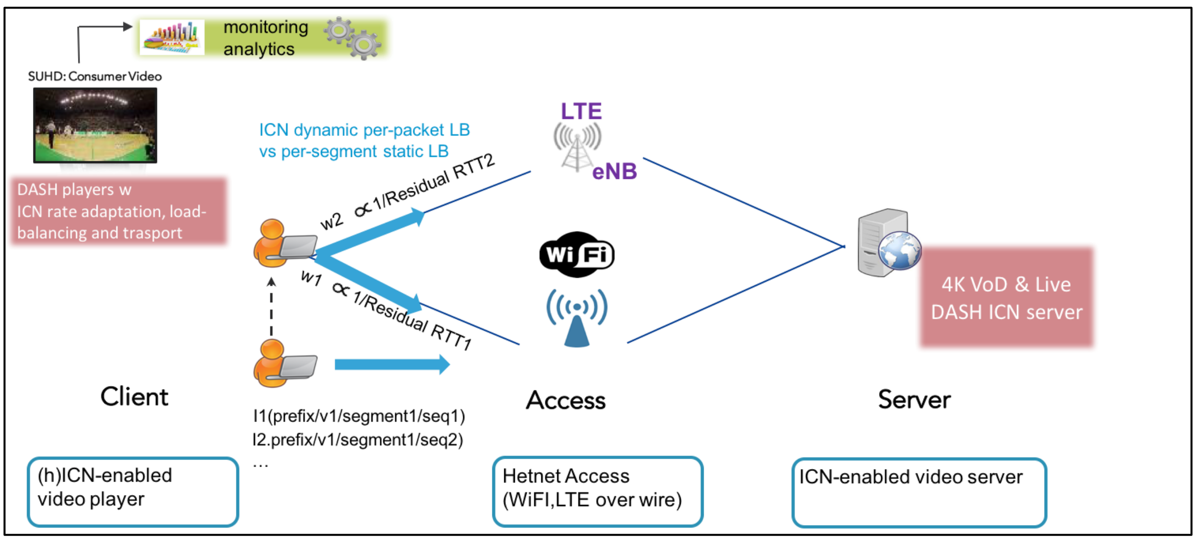


Figure-4: ICN Enhancements

3)WLDR, in network Wireless Loss Detection and Recover. In-network loss detection and recovery is a result of the connectionless request-reply transport model and a powerful feature to overcome standard transport limitations causing inefficient video delivery: (i) TCP/IP congestion control poorly performs in presence of wireless losses and does not handle mobility events; (ii) end-to-end control closed loop is slow (at least one RTT – round-trip time).
Instead, (h)ICN enables sub-RTT loss detection and recovery by delegation at key network nodes (consumer/producer/access points) of wireless, mobility, congestion events. We in Cisco have designed

WLDR, MLDR (Wireless, Mobility Loss Detection and Recovery) mechanisms, the latter generalized to congestion management, for handling loss detection and recovery in the network with low latency [4].

4)-5) Multi-path support and dynamic load-balancing. The receiver-based transport of hICN natively supports multiple paths and packet level load balancing function among all available faces, applied directly by the client. This function remains transparent to the application, with the controller still operating on the aggregate rate. In the case where the client in is multi-homed, e.g. Wi-Fi and LTE: the hICN client can perform load balancing of interest requests (so that data in return will travel along the trail of interests and be load balanced as well). The decisions are dynamically taken based: hICN clients monitor over time the residual latency for each prefix associated to a face. The advantage over existing solutions (leveraging MTCP/QUIC) is that the packet-level load-balancing of hICN adapts fast to varying network conditions and exploits the aggregate bandwidth of both accesses. Instead, the load-balancing of connection-based transport protocols may only, as for the bandwidth estimation, be performed at video-segment level and hence results in an oscillating selection of one of the two paths over time (with a negative impact on the stability of DAS rate adaptation), worse than single path selection.

5)-6) In-network caching and application-centric forwarding

hICN transport is coupled with in-network caching: content can be locally stored at every node and a cache lookup performed upon reception of an interest may result in data response from cache with no further propagation of the interest. Also, caching/forwarding strategies can be defined per name-prefix. By defining video quality-aware policies hICN permits to reactively cache closer to the access the more popular qualities as based on the access-dependent request pattern (i.e. available per-user bandwidth, type of devices etc.), while storing overall popular qualities in backhaul/core where they may serve more aggregated demand not satisfied by caches closer to the access. It results a reduced transport cost in terms of bandwidth savings due to traffic localization at the edge and to multicast. The latter can be additionally enhanced by network-assisted selection of rates/qualities to prefer popular multi-casted values over higher user-requested ones when this does not affect user experience.

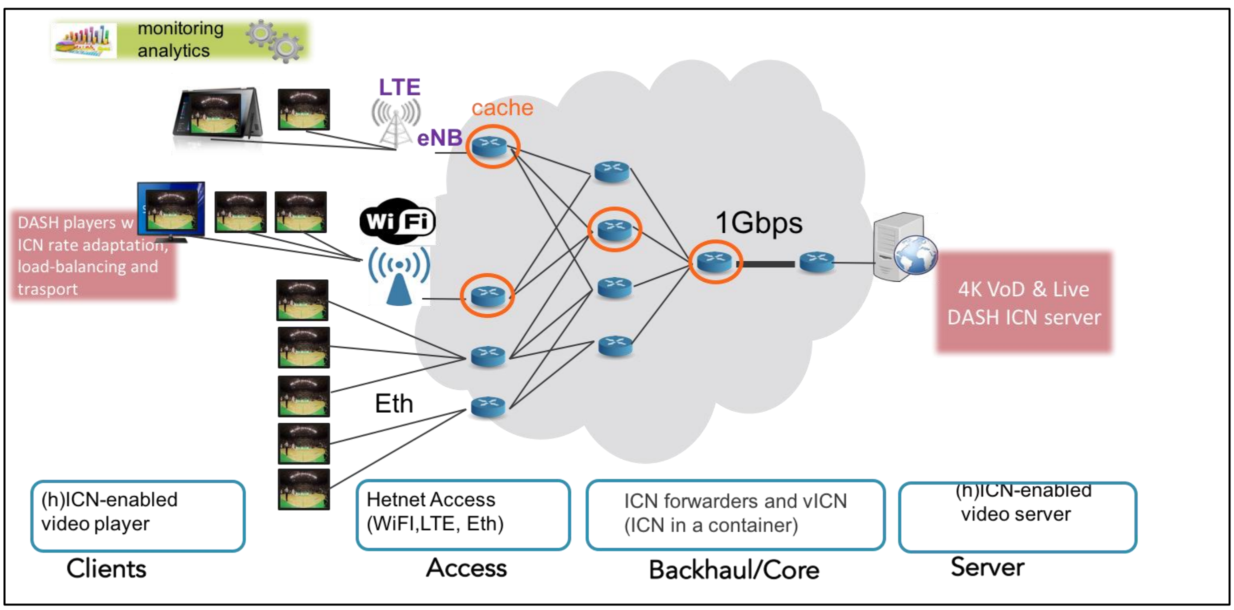
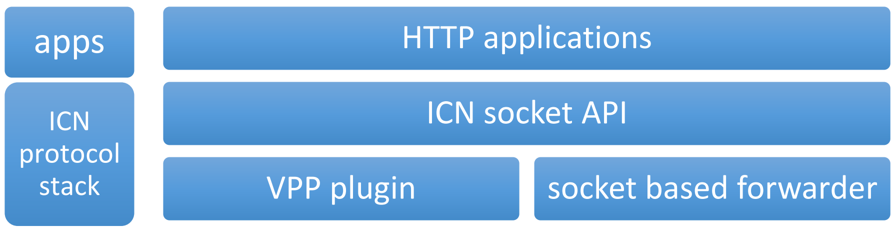


Figure 5: ICN Multicasting

Deployment

Reference Implementation

The reference implementation for the ICN stack is based on the CCNx 1.0 protocol specifications, which are IRTF drafts adopted by the ICNRG and currently under development in the same group. The overall system is built by assembling a number of software components, namely a packet forwarder, a socket API and supporting libraries.



The main forwarder is implemented as a VPP plugin which can be loaded at runtime in a VPP (https://wiki.fd.io/view/VPP) instance to enable ICN network functions. The VPP plugin is the I/O bottleneck free, user space software router that can run in commodity hardware at high speed, supporting DPDK and Linux AF_PACKET drivers. If the former driver is optimized for high speed forwarding with hardware accelerations the latter is used to interconnect to local applications (e.g. an HTTP server) or to forwarder instances that run inside non DPDK environments.

Several devices are unable to host VPP, namely end devices and low resources equipment. This is why the ICN software distribution includes a purely socket based forwarder, highly portable, that can be embedded in current mobile and desktop operating systems as an app.

The same philosophy has been used to develop the ICN socket API (ICNET) which is the application toolkit to enable ICN networking inside applications. ICNET is distributed as a C++ library with boost dependencies and implements different ICN socket types: reliable, stream unreliable and datagram.

An ICN socket implements named-data reassembly at the consumer and data segmentation at the producer as well as data naming and signing. While producer authenticity and data integrity are mandatory features of the socket API, the producer end can optionally provide confidentiality services by encryption. The reliable socket type implements flow and congestion control, as well as loss detection and recovery with network assisted capabilities, like hop-by-hop congestion control.

Among the features that distinguish an ICN socket API to an IP socket API one notable is the presence of data caching inside the socket itself on both consumer and producer. Data naming naturally enables this features which is integrated with no effort into the set of enhancements provided to current and future applications.

Among the many applications running on top of the Internet today, we have focused on HTTP that carries most of the traffic nowadays, especially video. An example of HTTP server and DAS video player is included in the software distribution to demonstrate how HTTP can run over ICN flawlessly with a number of new features that are gracefully enable by the underlying transport network: multicast, multi-source, caching and mobility among others.

Virtualization and orchestration (vICN)

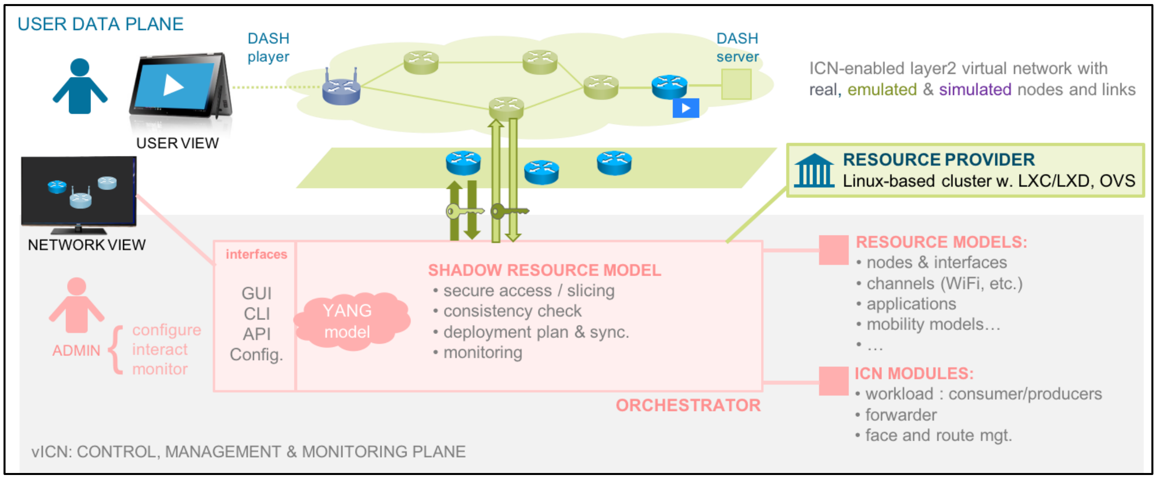


Figure 6: hICN Virtualized Infrastructure

Fast network configuration, deployment and management can be achieved by using recent advances in virtualization and orchestration of network resources. vICN leverages available techniques in the field and adapt them to the ICN context. The vICN system is based on two main components: virtualization and orchestration.

Virtualization embraces a number of resources like compute and networking and is implemented using Linux containers (LXC) and a hypervisor (LXD) to enable orchestration. Linux containers allow for lightweight virtualization with fast bootstrap and management as well as higher density with respect to other technologies. Orchestration is obtained using a centralized controller and a number of local agents to care of enforcing configurations and policies. The controller is a python implementation that communicates with the local agent using a number of interfaces like SSH execution and REST API. The **orchestrator**, called LURCH, is fed with a network model including different kind of resources that compose the overall network: switches, routers, servers, clients, channels etc. LURCH is also fed with network workload that can be used to run emulated experiments. LURCH is designed to host ICN networks by deploying a network topology the switching, name based routing for ICN as well as IP routing for management. It can also deploy different kind of network services like the DNS, HTTP servers and instrumentation tools like ICN ping.

The overall system is designed for large scale deployment, taking care of parallel computation and concurrency while deploying resources with different kind of dependencies to build the final ICN service.

Open Source and Standards

The whole software has been recently open sourced under the umbrella of the Linux Foundation in the FD.io (https://fd.io/) project. The open source initiative called Community Information-Centric Networking (cicn) will gather researchers and engineers, industry and academia in a common environment driven by running code and open standards with and experimental driven philosophy. The project will support most relevant use cases for today Internet as mobile video, 5G and IoT.

Conclusions

The designed hybrid ICN/IP solution unleashes all benefits of ICN communication with small upgrades of of the existing IP infrastructure. By deploying a virtualized hICN network at scale we showcased ICN advantages for mobile video over heterogeneous access (considering WiFi/LTE but directly portable to any access technology).

The demonstrated benefits result from key features of ICN communication model we have contributed to define and from video-specific enhancements of Dynamic Adaptive Streaming solutions. At the access, hICN brings improved DAS rate adaptation and dynamic load-balancing over multiple media leading to better user experience. In the network, video quality-aware forwarding/caching strategies maximize traffic localization and bandwidth savings in backhaul/core via enhanced multicast and network-assisted rate adaptation.

Contacts:

Giovanna Carofiglio, Cisco Distinguished Engineer, gcarofig@cisco.com

Printed in USA CXX-XXXXXX-XX 10/11