ICNRG
Internet Draft
Intended status: Informational
Expires: April 2016

J. Hong
ETRI
W. Chun
HUFS
H. Jung
ETRI

October 19, 2015

Bloom Filter-based Flat Name Resolution System for ICN
draft-hong-icnrg-bloomfilterbased-name-resolution-05.txt

Status of this Memo

Abstract

In information-centric networking (ICN), uniquely identifiable and location independent names are assigned directly to the named data which raises scalability issues and they get even worse with flat names. Accordingly, name resolution system required for lookup-by-name routing in ICN has to be designed to scale, also considering mobility support. In this draft, a bloom filter-based flat name resolution system (B-NRS) is proposed where the bloom filter as an aggregated form of names and hierarchical structure of the B-NRS are exploited to address the scalability issues.

Table of Contents

1. Introduction 4
2. NRS Requirements 5
   2.1. Scalability 5
   2.2. Fast resolution.. 5
   2.3. Fast update 6
   2.4. Resilience 6
   2.5. Security 6
3. Bloom Filter-based Flat Name Resolution System (B-NRS) 6
   3.1. System structure 6
   3.2. B-NRS Server Components 8
      3.2.1. Name Lookup Table 8
      3.2.2. Bloom Filter 8
   3.3. Key Operations 9
      3.3.1. Name Registration 9
      3.3.2. Locator Update 10
      3.3.3. Locator Lookup 11

4. Comparison of B-NRS with Other NRSs 12
5. Implementation Issues 14
5.1. False Positive 14
5.2. Membership Deletion 15
5.2.1. Use case 16
6. Implementation of B-NRS 16
6.1. Protocol Message 17
7. Security Considerations 19
8. IANA Considerations 19
9. References 20
9.1. Normative References 20
9.2. Informative References 20
A.1. Authors' Addresses 22

1. Introduction

In contrast to the host-centric networking in the current Internet, the primary communication object in information-centric networking (ICN) is named data, where uniquely identifiable and location independent name is assigned directly to the named data. This shift raises scalability issues to a new level. The current Internet is addressing on the order of 10^9 nodes, whereas the number of addressable ICN objects is expected to be several orders of magnitude higher [ICNRG charter]. Accordingly, name resolution system required both for lookup-by-name routing in ICN [ICN Challenges] and for ICN-IoT architecture [ICN-IoT] has to be designed to scale, also considering mobility support.

In this draft, we propose a bloom filter-based flat name resolution system (B-NRS) which maintains and resolves the binding between names and locators, i.e. B-NRS takes a name as its input and produces the locator sets that the name is currently associated with. We assume that the locator independent names are flat since the flat names provide some advantages compared to hierarchical ones, such as higher flexibility, simpler name allocation and benefits in terms of persistency and privacy [Ghodsi, ITU]. On the other hand, scalability becomes the most important challenge on designing the NRS supporting flat names. It is because of the ever increasing number of names in the network and no possible way to compactly represent the flat names such as the aggregation in IP addresses.

In order to address the scalability issue in designing the NRS for flat name, we need to aggregate names in any shape of type. One popular technique for flat name is Distributed Hashing Table (DHT) based approach [Hanka, Luo, Ahlgren, Mathy], where multiple servers form circular linked list and the bindings are stored in the appropriate server. However, the DHT technique has some drawbacks; the binding must be stored in a server other than the owner's, which causes a serious trust problem related to the authority issue and lookup message may be propagated through the long paths.

In this draft, to overcome the drawbacks of DHT, we exploit the bloom filter as an aggregated form of names and hierarchically construct the B-NRS. One of the major benefits of the bloom filter is a fixed constant time of insertion and search which is completely independent of the number of names already in the set. Another important and powerful property of bloom filter is the efficient support for union of bloom filters with the same size and set of hash functions which can be implemented with bitwise OR. However, bloom filter also has some drawbacks; false positive and no member deletion. Although there is no way to get rid of the false positive, it can be minimized by choosing the right parameters. The deletion problem is also taken care by periodic reconstruct of the bloom filters or by using variants of the bloom filter such as the counting bloom filter.

We note that the B-NRS in this draft does not require any specific mechanism for registering names, since names have no structure and can be registered to any B-NRS server with no constraint. Thus, the B-NRS needs only lookup mechanism. Whereas in the DHT-based system, the lookup message for a name is forwarded by the same way how to register the name.

2. NRS Requirements

Name resolution system (NRS) may become the bottleneck of the network when the signaling overhead of the location update and lookup becomes very large. Thus, the NRS must provide fast update and lookup for good performance since its basic functionality is to return the current locator for a given name. The NRS also must be secure and resilient because there is no way to respond to the querying message if the NRS is attacked. Obviously, the NRS must be scalable to the number of the ever-increasing ICN objects, i.e. names. Therefore, in this section, we discuss such requirements of the NRS.

2.1. Scalability

In ICN, the primary communication object is named data, where uniquely identifiable and location independent name is assigned directly to the named data. This raises scalability issues to a new level. The current Internet is addressing even on the order of 10^9 nodes, whereas the number of addressable ICN objects is expected to be several orders of magnitude higher considering sensor data, vehicular, Internet of things, etc. Accordingly, the NRS should be able to fully cover the ever-increasing number of ICN objects.

2.2. Fast resolution

A fundamental problem with any global query server network is that the requestor who sends the name resolving request may significantly delay or drop the initial packet of a new session if the resolution time gets too long. Thus, the resolution time should be sufficiently low so it does not affect much the overall system performance.

2.3. Fast update

When a named date moves and changes its point of attachment to Internet or a multi-homed device shuts down one of its physical interface, it needs to update the old information with the new one or delete the deprecated information in NRS. Thus, the NRS should adapt quickly with such changes.

2.4. Resilience

If the NRS fails, there is mostly no way for the requestor to reach other end information since the requester knows only its names. Therefore, the NRS must not fail.

2.5. Security

The NRS can be a potential target for attacks such as denial-of-service attacks. These types of attacks are difficult to prevent. Thus, updates to the NRS or responses from NRS server should be authenticated.

3. Bloom Filter-based Flat Name Resolution System (B-NRS)

We propose a bloom filter-based name resolution system (B-NRS) for supporting flat name which maintains and resolves the binding between names and locators.

3.1. System structure

We construct the B-NRS hierarchically by defining a network of B-NRS servers, which consists of a forest by several disjoint trees. The network of B-NRS servers is defined by both parent-child and peering relationships.

Figure 1 is an example of the B-NRS structure which consists of 8 B-NRS servers forming a tree, where there exists the peering relationship between S2 and S3. The peering relationship is allowed for better performance by reducing the overhead for the B-NRS at the top of the tree. A leaf B-NRS server knows every single name/locator pair that it manages but nothing else. The intermediate B-NRS servers know the name/locator pair for all names that are directly registered to them and also possess only information about the names that their descendant and peer B-NRS servers manage. Although there is a single tree in figure 1, if we assume there are several trees forming a forest, then the B-NRS servers are fully peered at the top of the trees. This means that each server shares its knowledge of all names that it manages with its peers.

We note that we have been very careful in distinguishing between the name/locator pair information and the name information. This distinction is necessary to provide a different level of information abstraction, which is naturally achieved through the hierarchical B-NRS structure and the use of bloom filters.

```
                    +----+
                    | S1 |
                    +----+
                   /      \
                  /        \
                 /          \
                /            \
               /              \
              /                \
             /                  \
      +----+                   +----+
      | S2 |*******************| S3 |
      +----+                   +----+
        / | \                      /\
       /  |  \                    /  \
      /   |   \                  /    \
     /    |    \                /      \
    /     |     \              /        \
   /      |      \            /          \
  /       |       \          /            \
+----+  +----+  +----+    +----+        +----+
| S4 |  | S5 |  | S6 |    | S7 |        | S8 |
+----+  +----+  +----+    +----+        +----+

Legend:

+---+
| S |  B-NRS Server
+---+
-----  Parent-child relationship

*****  Peering relationship
```

Figure 1. An example of B-NRS structure

3.2. B-NRS Server Components

A B-NRS server consists of a name lookup table and multiple bloom filters.

3.2.1. Name Lookup Table

Name lookup table stores the binding between names and locators for all names which are directly registered to the BRS server. The associated locator for a certain name can be more than one. So, the locator information is stored as a set shown in table 1. Name lookup table takes a name as the input and produces its associated locator sets as the output.

Table 1. Lookup table

Name	Locators
N1	LOC1
N2	LOC2-1, LOC2-2
N3	-
N4	LOC4-1, LOC4-2, LOC4-3

3.2.2. Bloom Filter

We utilize bloom filters as an aggregated form of names at each B-NRS server. B-NRS servers announce their name set to the other B-NRS servers. Instead of announcing the whole list of names, bloom filter as an aggregated form of names is announced. When announcing its name set to its peers or parents, the B-NRS server announces the union of name sets of all child B-NRS servers. Union of child name sets can be built by using the characteristic of bloom filer that bloom filter for union of sets can be built merely by bitwise 'OR' operation on all the sets.

Thus, each B-NRS server stores bloom filters for itself, from children, and from peers depicted in figure 2. The B-NRS server stores n+m+1 bloom filters in figure 2, where n is the number of child B-NRS servers and m is the number of peer B-NRS servers.

We note that the forest of B-NRS servers retains the loop-free property for the use of bloom filter.

```
             /  -------------------------   \
            /  | BF for its own         |    \
            /                            --------------------------     \ Bitw
ise OR
  +----------------+ /                            --------------------------
  / To Parents and Peers
  | B-NRS Server   |     | BFs from Child 1 to n   |   /
  +----------------+ \   --------------------------   /
            \   --------------------------
            \  | BFs from Peer 1 to m    |
             \ ---------------------------
```

Figure 2. B-NRS server components

3.3. Key Operations

3.3.1. Name Registration

When a communication entity attempts to join the network, it must register itself in at least one B-NRS server. In this draft, it is allowed that the communication entity can be registered in any arbitrary B-NRS server since names have no structure.

Upon receiving the registration request from the communication entity, the B-NRS server registers the name to its lookup table. The locators for the name are stored in the table when the communication entity for the name is actually present into the network. We separate this as the operation of locator update from the name registration.

The name registration is along with bloom filter update. When a communication entity is registered in a B-NRS server, the registration information is extracted from its name using the hash functions for its bloom filter and inserted into its own bloom filter first and then the B-NRS server updates bloom filters for its parents and peers, where this recursion holds until bloom filters at the top of trees are completely updated.

Figure 3 shows an example of the name registration and bloom filter updates, where a new name is registered at the B-NRS server, S4. It inserts information of the new name first into its own bloom filter and updates its parent, S2. Then, S2 updates its parent, S1 and its peer, S3.

When names are deleted from the lookup table, we need to adopt a certain mechanism to update the bloom filters for the deletion since bloom filter cannot handle the deletion by itself. Thus, we use the periodic refresh technique that bloom filters with registered names are rebuilt periodically and followed by bloom filter updates.

```
          (3)BF
          Update    +----+
           -------->| S1 |
          |         +----+
          |        /      \
          |       /        \
          |      /          \
          |     /            \
          |    /              \
          |   /                \
(2)BF     |  /                  \
 Update +----+  (3)BF Update     +----+
  ----> | S2 |------------------>| S3 |
 |     +----+*******************+----+
 |      / | \                      /\
 |     /  |  \                    /  \
 |    /   |   \                  /    \
 |   /    |    \                /      \
 |  /     |     \              /        \
 | /      |      \            /          \
 |/       |       \          /            \
+----+  +----+  +----+    +----+        +----+
| S4 |  | S5 |  | S6 |    | S7 |        | S8 |
+----+  +----+  +----+    +----+        +----+
  ^^
  ||
  || (1)Name registration
  ||
```

Figure 3. Name registration and BF update

3.3.2. Locator Update

When a communication entity actually presents in the network, the locator update is occurred, where the gateway sends the locator update message to the correspondent B-NRS server and the locator associated with the name is stored in the lookup table. If the name has multiple locators, then they are stored as a set of locators for the name. Through the bloom filter test of the name, the locator update messages are forwarded into the lookup table where the name is actually stored.

When the communication entity depresents from the network, the locators for the name is deleted from the lookup table by the locator update message as well. Table 1 shows the depresence of entity for the name, N3. We note that changing locators has no effect on the structure of the B-NRS and mobility is easily supported.

3.3.3. Locator Lookup

The lookup operation is to find the locator information for a given name. The simplest case is when the source object tries to communicate with the destination object registered in the same B-NRS server. B-NRS server always searches for the destination name in its own lookup table first so the locator information is acquired at the first lookup in such a case.

A harder, but more interesting, case is when the destination object is registered in the other B-NRS server with the source object. In this case, the B-NRS server would quickly learn that the destination object is not registered in the same B-NRS server by a simple search of its lookup table. Then, it searches bloom filters for its child and peer B-NRS servers. If none of the bloom filters return a positive answer, the lookup request message is forwarded to its parent B-NRS server. On the other hand, if any of bloom filters return a positive answer, the lookup request message is forwarded to every B-NRS server that corresponds to the bloom filters with positive answers. We note that because of the false positives of the bloom filter, multiple bloom filters may return positive answers.

This search is done recursively, and the locator information for the destination name can eventually be found. Once the locator information is found, it is delivered to the source object by the lookup reply message which takes the reverse path of the lookup request message.

Figure 4 is an example of lookup and registration processes where the lookup message for a name which is registered at S8 is received by S4. Then, the lookup message is forwarded to S2. Since S2 is peered with S3, S2 forwards it to S3 not to S1. S3 forwards it to

S8. The reply message takes the reverse path of the lookup request message, i.e., S8->S3->S2->S4.

```
                             +----+
                             | S1 |
                              +----+
                             /      \
                            /        \
                           /          \
                          /            \
                         /              \
                        /                \
                       /                  \
        (2)Lookup +----+  (3)Lookup        +----+ (4)Lookup
            ----> | S2 |<----------------->| S3 |<------
           |                         +----+*******************+----+       |
           |      /  |  \   (6)Reply           /\         |
           |     /   |   \                    /  \       |
 (7)Reply  |    /    |    \                  /    \      |(5)Reply
           |   /     |     \                /      \     |
           |  /      |      \              /        \    |
           | /       |       \            /          \   |
           v/        |        \          /            \  v
(1)Lookup +----+  +----+  +----+    +----+        +----+
<-------->| S4 |  | S5 |  | S6 |    | S7 |        | S8 |
(8)Reply  +----+  +----+  +----+    +----+        +----+
```

Figure 4. Lookup and reply

4. Comparison of B-NRS with Other NRSs

One of the critical challenges in designing NRS is scalability due to the ever increasing number of names. In order to overcome this issue, names need to be distributed and also aggregated in any shape of type especially for flat names. One popular technique to distribute and aggregate names is to use DHT (Distributed Hash Table). However, DHT has several drawbacks such as ownership, deployment, locality, etc. Thus, we exploit the bloom filter as an aggregated form of names and hierarchically construct the NRS.

As illustrated in figure 5, NRS can be roughly divided into two types: centralized vs. distributed. Then, the distributed type can be divided again into two approaches: DHT-based vs. all else. DMap (Direct Mapping) [DMap] and MDHT (Multiple DHT) [MDHT] are examples of DHT-based approach. DMap is proposed by MF (MobilityFirst) which is one of the Future Internet architecture projects funded by NSF in US and MDHT is by SAIL (Scalable and Adaptive Internet Solutions) which is an EU-funded project. B-NRS belongs to the distributed type but not DHT-based approach.

```
  *=================== NRS ===================*
 |                                             |
 |    *=======================*                |
 |   *       Distributed       *  Centralized  |
 |  *                           *              |
 | *   *==========*             *              |
 | *  *  DHT-based *            *              |
 | *  * o MF-DMap   *  o B-NRS  *              |
 | *  * o SAIL-MDHT *           *              |
 |  *  *===========*            *              |
 |   *                         *               |
 |    *=======================*                |
 |                                             |
 *=============================================*
```

Figure 5. A simple Venn diagram categorizing NRS

Table 2 presents the comparison of B-NRS with DMap and MDHT in respect of scalability, lookup latency and locator update.

For scalability, we compare how many names can be scalable for each NRS. DMap assumes that the number of names is 5*10^9, whereas MDHT and B-NRS assume that it is 10^15.

We define the lookup latency as the multiple of the number of hops, H, and the processing time per hop, T(N), which is proportional to the number of table entries, N. The lookup latencies for both DMap and MDHT are increasing proportionally to the number of hops and the number of table entries at each hop since the table lookup is processed at each hop. However, the lookup latency for B-NRS is dependent only to the number of hops since BF takes a fixed constant time, C, for searching. Even though each B-NRS server has several bloom filters, they are independent to each other and can be parallelized in a hardware implementation.

For locator update, we look at the staleness. Both DMap and MDHT do the location update periodically so the staleness occurs during it is not updated. However, the staleness for B-NRS occurs with probability 0 since it does the location update in real time.

Table 2. Comparison of B-NRS with DMap and MDHT (N and H are the number of table entries and hops, respectively and C is a constant.)

Design goal	Scalability	Lookup latency	Locator update
Metric	number of names	number of hops * processing time per hop	Staleness
MF-DMap	~5*10^9	H*T(N)	periodic update: occur
SAIL-MDHT	~10^15	H*T(N)	periodic update: occur
B-NRS	~10^15	H*C	real time update: occur with probability 0

5. Implementation Issues

Bloom filter has the well-known drawbacks such as false positive and no membership deletion. However, the false positive can be minimized by choosing the right parameters and the deletion problem can also be taken care by adopting a certain mechanism to update the bloom filters for the deletion such as the counting bloom filter, periodic reconstruct of bloom filter, etc.

5.1. False Positive

The width of a bloom filter is directly related to the false positive rate for fixed number of hash functions, the length of the bloom filter is inversely proportional to the false positive rate. Although a lengthier bloom filter is ideal for minimizing the false positive rate but increasing the B-NRS search efficiency, it creates a burden when filter information are exchanged among B-NRS servers.

In addition, since a leaf B-NRS server has a smaller number of names that it needs to manage, it makes sense to use a smaller bloom filter than the B-NRS servers at the higher level of the B-NRS hierarchy. However, the variable bloom filter length approach must be done with care since the key property, union of bloom filter via bit-wise AND operation, may be lost when variable length bloom filters are used.

5.2. Membership Deletion

One of the main advantages of the bloom filter is that data insertion and search can be done in a constant time. However, its major drawback is that a bloom filter does not have an efficient method of supporting data deletion. Of course, there are variants of the bloom filter to overcome the deletion issue. For example, the counting bloom filter supports the deletion by associating a counter to every bit of the bloom filter, where data insertion corresponds to incrementing the counters associated with the bits; data deletion to decrementing the counters; and query to checking whether the counters are positive. However, since each counter needs to have sufficient number of bits to prevent overflow; thus, it is a less space efficient than the traditional bloom filter. The space efficiency is critical to our B-NRS since bloom filters are exchanged among B-NRS servers and it is directly proportional to the size of exchanged control messages.

Because of this drawback of deletion of bloom filter, B-NRS needs to be carefully designed to support dynamic registration and deregistration of communicating entity.

In one extreme case, even if the de-registration were to be completely ignored by the B-NRS, the B-NRS would eventually be able to find the locator for a given name. This method will generate the fewest number of control messages (bloom filter updates) but the query would become inefficient since this would significantly increase the false positive rates.

The other extreme case would be to update the entire B-NRS whenever there is a single de-registration. Although this method would have the lowest false positive rates, and thus, would have the lowest average number of queries to find the name/locator pair, it would have a very high control message load since there would be a lot of bloom filter exchanges among B-NRS servers.

Certainly, the B-NRS will operate within these two extreme bounds, and the optimal rate is a design parameter in building the B-NRS system.

B-NRS overcome the deletion issue by periodically rebuilding bloom filters using the shadow memory, so called periodic refresh. The refresh frequency can be a day, a week, a month, etc. When B-NRS is refreshed, names in a name lookup table are inserted into the new bloom filter at a time and the merged bloom filters by bitwise OR are announced to parent and peer B-NRS servers. For better performance, the lossless compressed bloom filter can be used to announce the merged bloom filter. We note that the false positive probability certainly increases until all bloom filters are replaced by new bloom filters.

5.2.1. Use case

What happened if the deleted name is requested before the bloom filters are refreshed? The lookup message for the deleted name will be forwarded to the B-NRS server which stored the name. Once it gets the server, it will learn that the name does not exist in the lookup table of the server. Then, the lookup message is processed as a false positive case so that it would eventually return a response that there is no such name registered in the system. Therefore, the requestor would get a correct corresponding response even when the bloom filters are not refreshed.

Now, what is the difference between before and after the bloom filter refreshment? The requestor for the deleted name will get the same response in both case but the response will be processed much sooner after the refreshment since the lookup will not be forwarded to the server which stored the name.

As a result, it may not be fatal in B-NRS that bloom filter cannot handle the membership deletion. However, the periodic refreshment of bloom filters are necessary for the better performance and management.

6. Implementation of B-NRS

We have created prototypes for B-NRS: NRS server, top server, and client. Although all B-NRS servers perform the same functions, we separate top server from the others for convenient implementation. We have utilized the parallel process of a graphics processor unit (GPU) to accelerate the performance of BF check at each B-NRS server resulting in low latency.

We have used an algorithm for the GPU usage. The main idea of the algorithm is to enable to extract only the corresponding bits for the given name check from all BFs at each server to GPU memory and check the extracted bits in parallel to see if any chunk gives 1 by bitwise 'AND' operation. In this implementation, we use 16Mb BF size and 11 hash functions to keep the false positive probability less
6 information at a maximum of 10 names. We have u
sed the static tree
structure of B-NRS which is managed by configuration files of each server. We have also implemented the B-NRG by using CPUs to see the effect of the GPU usage on performance. It showed that the search time of a number of bloom filters with GPU was almost constant up to the number of GPU cores. In other words, as expected, the search time with CPU was linearly increasing according to the number of bloom filters. The search time with GPU became shorter than the time with CPU when the number of bloom filters was greater than a certain amount, which value is dependent to the specification of GPU and CPU. Using GPU is also much more cost-effective compared to CPU. This results are powerful when the number of bloom filters in a B-NRS server is huge. A number of bloom filters in a B-NRS server means that the server has the amount of child servers including peers. Having a number of child server is desirable because it is the way to reduce the height of the B-NRS hierarchy resulting in reducing the number of B-NRS server accesses per a lookup.

6.1. Protocol Message

We keep the flat name size as 24 bytes and use the UDP communication with port number, 7979 in the implementation. Prot in protocol messages is the protocol type of 1 byte size. Locator is defined as a variable length string.

O Name registration

```
+-----------------------------------------+
| Prot |             Name                 |
+-----------------------------------------+
```

O Locator update

Locator update message is divided into three types: Add, Delete, and Replace.

```
+----------------------------------------------------------+
| Prot | Mode |  Name   |  Locator length  |    Locator    |
+----------------------------------------------------------+
```

Mode is the type of locator update.

O Locator lookup

```
+-----------------------------------------+
| Prot |              Name                |
+-----------------------------------------+
```

O Name deregistration

It deletes the name and the corresponding locators from name lookup table.

```
+-----------------------------------------+
| Prot |              Name                |
+-----------------------------------------+
```

O BF update

```
+-----------------------------------------+
| Prot |              Name                |
+-----------------------------------------+
```

O CMD_Lookup

It is the locator lookup message between B-NRS servers.

Prot	Name	Client IP	Up/Down	Depth

It keeps the IP address of the client who creates the locator lookup message so the locator information could be delivered directly to the client once it is found. Up denotes that the lookup message is to parent server and Down is to child servers. We increase the Depth by 1 whenever the message is forwarded to child. We keep the depth information because of the false positive of BF.

O CMD_Lookup NACK

When BF check fails, it is sent to parent server.

Prot	Name	Client IP	Up/Down	Depth

7. Security Considerations

False positive error is one of the well-known drawbacks of bloom filter and there is no way to get rid of it. Thus, it can be an attack point. For example, if an attacker puts wrong information into bloom filters of B-NRS in order to increase the false positive error rate resulting in getting traffics to go far away and consuming resource, then the performance degradation may occur until the B-NRS is refreshed. Once B-NRS is rebuilt, there will be only probabilistic false positive error rate not the deterministic one.

8. IANA Considerations

TBD

9. References

9.1. Normative References

9.2. Informative References

[ICNRG charter] http://irtf.org/icnrg

[ICN Challenges] D.Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "ICN Research Challenges ", draft-kutscher-icnrg-challenges-02, February 2014.

[ICN-IoT] Y. Zhang, D. Raychadhuri, L. Grieco, E. Baccelli, J. Burke, R. Ravindran, and G. Wang, "ICN based Architecture for IoT -Requirements and challenges", draft-zhang-iot-icn-challenges-01, December 2014.

[Ghodsi] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and Shenker, "Naming in Content-Oriented Architectures," In Proceedings of the SIGCOMM ICN'11, August 19, 2011, Toronto, Ontario, Canada.

[ITU]   International Telecommunication Union (ITU), "ITU-T Recommendation Y.3031 - Identification framework in future networks," available at: http://www.itu.int/rec/T-REC-Y.3031-201205-P/en, 2012.

[Hanka] O. Hanka, C. Spleiss, G. Kunzmann, and J. Eberspacher, "A novel DHTbased network architecture for the next generation internet," Eighth International Conference on Networks, Cancun, Mexico, March 2009.

[Luo]   H. Luo, Y. Qin, and H. Zhang, "A DHT-Based Identifier-to-Locator Mapping Scheme for a Scalable Internet," IEEE Transactions on Parallel and Distributed Systems, October 2009.

[Ahlgren] B. Ahlgren, J. Arkko, L. Eggert, and J. Rajahalme, "A node identity internetworking architecture," in INFOCOM 2006. 25th IEEE International Conference on Computer Communications Proceedings. Washington, DC, USA: IEEE Computer Society, April 2006, pp. 1-6.

[Mathy] L. Mathy and L. Iannone, "LISP-DHT: Towards a DHT to map identifiers onto locators," in ReArch'08. Madrid, Spain: ACM, December 2008.

[Fab1999] Faber, T., Touch, J. and W. Yue, "The TIME-WAIT state in TCP and Its Effect on Busy Servers", Proc. Infocom 1999 pp. 1573-1583.

[DMap] T. Vu, A. Baid, Y. Zhang, T. Nguyeny, J. Fukuyamaz, R. Martin, and D. Raychaudhuri, "DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet," Proceedings of the IEEE International Conference on Distributed Computing Systems, pp. 698-707, 2012.

[MDHT] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: A Hierarchical Name Resolution Service for Information-centric Networks," ICN'11, August 19, 2011, Toronto, Ontario, Canada.

A.1. Authors' Addresses

Jungha Hong
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: jhong@etri.re.kr

Woojik Chun
Hankuk University of Foreign Strudies
81, Oedae-ro, Mohyeon-myeon, Cheoin-gu, Yongin-si, Gyeonggi-do, Korea

Email: woojikchun@gmail.com

Heeyoung Jung
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: hyjung@etri.re.kr

C. Hur
J. Kim
H. Jung
J. Eun
ETRI
W. Chun
HUFS

October 28, 2014

Abstracted Network API for ICN
draft-hur-icnrg-abstracted-network-api-00.txt

Status of this Memo

Abstract

In information-centric networking (ICN), well designed API will play a pivotal role in adopting ICN to applications. Traditional network API - i.e., current Socket implementation - is coupled between applications and underlying protocols, so it is hard to change one without changing the other, thus it could be solved by redesigning the network API in a way that decouples application-specific functions from network specific functions. In this draft, we addressed the network API modeling issues which can be also applied to the design of ICN APIs and proposed iPlug and dSocket API which help injection of applications' intent and hide underlying network specific mechanisms, respectively.

Table of Contents

1. Introduction 3
2. Abstracted Network API 4
   2.1. Intent modeling.. 4
   2.2. Loosely coupled modeling............................... 5
3. IPlug and DSocket API....................................... 6
   3.1. Key operations 8
4. Analysis 8
5. Security Considerations..................................... 8
6. IANA Considerations 8
7. References 9
   7.1. Normative References................................... 9
   7.2. Informative References................................. 9
   A.1. Authors' Addresses..................................... 10

1. Introduction

In the information-centric networking, the key idea is to let named data to be identified by name and to be retrieved wherever they are. To enable these communications, the named data need network API to use the functionalities of the underlying network. We bring up an issue for such an interface that is used by any source. Since many studies in interface patch up traditional socket-based API which is already tightly coupled to location, it is inevitable to redesign new network interface not only to be in accordance with location independent name, but also to meet the needs such as multihoming, and mobility[ICN Challenges].

Recent ICN-based contributions have been presented network API, such as NetInf[ref], CCN/NDN[ref] and Publish/Subscribe Networking. There is common ground on naming and network primitives for accessing any object, regardless of location. NetInf defines set of APIs to handle Named Data Object (NDO), such as PUBLISH, INDICATION, REQUEST, RESPONSE, etc. NetInf API is used to retrieve NDO which is identified by NI naming [RFC 6920], instead of IP and Port in traditional socket API. It employs receiver-driven transport for carrying chunks of an object, and provides convergence layer to bridge with other protocols, such as HTTP and DTN. CCN/NDN defines CCNx API [Mosko] which adopted URL in content name representation. All communication between the core protocol implementation, network, and applications is accomplished through a Face abstraction (e.g., link layer face, network-layer face, and transport-layer face). They also note that abstracted API is important for flexibility and extensibility.

In this regard, we need a common ICN API that enables applications to be independent, interworked and evaluated throughout heterogeneous network architecture. In addition, network architectures can evolve and converged with others. For this, it is necessary to define common primitive API which abstracts the essential behavior.

This draft introduces modeling consideration for API of ICN as well as other network architectures. We abstract the network API by applying Model Driven Engineering (MDE) and Separation of Concerns (SoC) principle [Morin, Moreira] which are principles in Object-Oriented design. The network API, in common with many other architectures, needs to be simple but powerful enough to help applications to focus on their own concerns regardless wherever the objects are located in and how to get the objects in the underlying network, but also assist separating ID/Locator in the Internet architecture. The Internet applications using the network API can focus on only what to communicate with, not where or how.

2. Abstracted Network API

This draft describes abstracted network API to inject communication intents from application to network. For this abstraction requires API modeling concepts at different point of view. The abstracted network API can overcome the problems in the application point of view in the following.

First of all, network application uses API that depends on some parts of network mechanism. Therefore, it has to concern with unnecessary aspects of network, such as name resolution, and congestion control.

Second, when new network mechanism such as ICN is introduced, existing applications are no longer available, but modification takes a lot of efforts. In addition, new architectural changes are hardly deployed because of difficulty of application migration. It often led to overwrap the existing network API.

Therefore, in network API, abstractions for both vertical and horizontal separation of concerns should be provided for reducing complexity. Vertical separation of concerns itemizes application's intent to be composable, and horizontal separation of concern reduces complexity by decoupling network-specific aspects from application, as shown in Figure 1.

2.1. Intent modeling

In the ID/Locator splitted network such as ICN, application is only required to handle whom to communicate, but not how to communicate. Given this network, this draft adopts the MDE for API to be modeled at two different levels of abstraction. Network Independent Model (NIM) and Network Specific Model (NSM). When we define a network independent model and a network mechanism dependent model, one can find intent of network application. An intent is concern of application and an application can have multiple intent. For instance, an application can have intent to communicate with endpoints and demand reliable transmission. To enable the network application over the network, intent injection from NIM to NSM are needed. For example, if voice call application requires secure communication channel and certain quality of service, its intent can be carried in various ways such as IPSec and jitter control.

2.2. Loosely coupled modeling

Loosely coupled modeling has several advantages in the network API. First of all, it helps to separate concerns between application and network layer. Because separation of concerns in horizontal manner is to present integrated way to use underlying network, despite the network might include certain networking mechanisms (e.g., protocol, delivery types). It helps to realize applications' various communication intents over the heterogeneous network mechanisms. Second, loosely coupled relation reduces the dependency. Also, it makes possible to communicate between applications regardless the lower-level component and its implementation. For example, when applications retrieve Named Data Object, loosely coupled API is required in different kinds of architectures (e.g., ICN, CCN). Similarly, by performing the dynamic association between communication intent and network mechanism, both sides can be independent.

```
+=========================================================+
| Network Independent Model                               |
| +----------+   +----------+        +----------+         |
| | Intent 1 |   | Intent 2 |   ...  | Intent N |         |
| +----------+   +----------+        +----------+         |
+========================|================================+
Intent Injection         |
+========================@================================+
| Network Specific Model                                  |
| +-------------+ +-------------+     +-------------+     |
| | Mechanism 1 | | Mechanism 2 | ... | Mechanism N |     |
| +-------------+ +-------------+     +-------------+     |
|              |           |          /                   |
|             +---------------------+                     |
|             | Network Architecture |                    |
|            +---------------------+                      |
+=========================================================+
```

Network API modeling

Figure 1.

3. IPlug and DSocket API

This draft proposes iPlug and dSocket as an abstracted API model. The iPlug API takes charge of application-specific capabilities. Application needs functionalities such as flow control, authentication, which is supported by iPlug. The dSocket is a cut-off point of separation and hold responsibilities that network should handle, such as congestion control, multipath, multihoming.

In Figure 2, relation of API components is described. At first, application binds unique ID to iPlug. The plug-in and plug-out behavior help dynamic association between iPlug and dSocket. When detailed metadata of dSockets are delivered to applications, and then dSocket notifies the status to dSocket Manager to inform their relation change. Then, the iPlug can plug into the dSocket to inject application's intent, and plug out of the dSocket. Detailed procedures of each API are described in the following.

```
                             +--------------------+
                             |     Application    |---+
                             +--------------------+   |
                             |       Extension    |   |BIND
  +-----------------+ FIND   +--------------------+   |
  |    dSocket      |<-------|        iPlug       |<--+
  |    Manager      |        +====================+
  +-----------------+                     | PLUGIN
             ^                            | PLUGOUT
             | REGISTER       +====================+
             ----------------|       dSocket      |
                              +--------------------+
                             |        Domain      |
                              +--------------------+
                          Components of API
```

Figure 2.

3.1. Key operations

The abstracted network API has main operations for dynamic convergence. The sequences and states of each operations are described in the folloing.

```
         iPlug                                                   dSocket

            |---+                                          |

            |   | BIND NAME                                |

IDENTIFIED  |<--+                                          |

            |                                              |

            |--------------     PLUG IN   ------------->| PLUGIN-RECEIVED

PLUGGED IN  |                                           | PRESENTED

            |                                              |

            |-------------      PLUG OUT  ------------->| PLUGOUT-RECEIVED

PLUGGED OUT |                                           | DEPRESENTED

            |                                              |
```

4. Analysis

TBD

5. Security Considerations

TBD

6. IANA Considerations

TBD

7. References

7.1. Normative References

7.2. Informative References

[ICNRG charter] http://irtf.org/icnrg

[ICN Challenges] D.Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "ICN Research Challenges ", draft-kutscher-icnrg-challenges-02, February 2014.

[RFC 6920]S. Farrell, C. Dannewitz, P. Hallam-Baker, D. Kutscher, and B. Ohlman, "Naming Things with Hashes." [Online]. Available: https://tools.ietf.org/html/rfc6920. [Accessed: 24-Oct-2014].

[Mosko]   M. Mosko, "CCNx 1.0 Protocol Introduction," Apr. 2014.

[NETINF] Scalable and Adaptable Internet Solutions (SAIL), "D.B.1: The Network of Information: Architecture and applications", available at: http://www.sail-project.eu/deliverables/, 2011

[CLICK] "The Click Modular Router Project." [Online]. Available: http://read.cs.ucla.edu/click/. [Accessed: 12-Jun-2014].

[Morin] B. Morin, F. Fleurey, N. Bencomo, J.-M. Jezequel, A. Solberg, V. Dehlen, and G. Blair, "An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability," in Model Driven Engineering Languages and Systems, K. Czarnecki, I. Ober, J.-M. Bruel, A. Uhl, and M. Volter, Eds. Springer Berlin Heidelberg, 2008, pp. 782-796.

[Moreira] A. Moreira, A. Rashid, and J. Araujo, "Multi-dimensional separation of concerns in requirements engineering," in 13th IEEE International Conference on Requirements Engineering, 2005. Proceedings, 2005, pp. 285-296.

[Fielding]R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine, 2000.

A.1. Authors' Addresses

Cinyoung Hur
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: cyhur@etri.re.kr

JongHwan Kim
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: ditto@etri.re.kr

Heeyoung Jung
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: hyjung@etri.re.kr

Jeesook Eun
ETRI
218 Gajeong-ro, Yuseong-gu, Daejeon, Korea

Email: jseun@etri.re.kr

Woojik Chun
Hankuk University of Foreign Strudies
81, Oedae-ro, Mohyeon-myeon, Cheoin-gu, Yongin-si, Gyeonggi-do, Korea

Email: woojikchun@gmail.com

ICNRG C. Westphal, Ed.
Internet Draft Huawei
Intended status: Informational S. Lederer
Expires: October 26, 2016 D. Posh
C. Timmerer
Alpen-Adria University Klagenfurt
Aytac Azgin
S. Liu
Huawei
C. Mueller
Bitmovin
A.Detti
University of Rome Tor Vergata
D. Corujo
University of Aveiro
J. Wang
City University of Hong-Kong
Marie-Jose Montpetit
Niall Murray
Athlone Institute of Technology

April 27, 2016

Adaptive Video Streaming over ICN
draft-irtf-icnrg-videostreaming-08.txt

Status of this Memo

Copyright Notice

Abstract

This document considers the consequences of moving the underlying network architecture from the current Internet to an Information-Centric Network (ICN) architecture on video distribution. As most of the traffic in future networks is expected to be video, we consider how to modify the existing video streaming mechanisms. Several important topics related to video distribution over ICN are presented, covering a wide range of scenarios: we look at how to evolve DASH to work over ICN, and leverage the recent ISO/IEC Moving Picture Experts Group (MPEG) Dynamic Adaptive Streaming over HTTP (DASH) standard; we consider layered encoding over ICN; Peer-to-Peer

(P2P) mechanisms introduce distinct requirements for video and we look at how to adapt PPSP for ICN; Internet Protocol Television (IPTV) adds delay constraints, and this will create more stringent requirements over ICN as well. As part of the discussion on video, we discuss Digital Rights Management (DRM) in ICN. Finally, in addition to considering how existing mechanisms would be impacted by ICN, this document lists some research issues to design ICN specific video streaming mechanisms.

Table of Contents

1. Introduction.. 4
2. Conventions used in this document............................... 5
3. Use case scenarios for ICN and Video Streaming.................. 5
4. Video download.. 7
5. Video streaming and ICN... 7
 5.1. Introduction to client-driven streaming and DASH 7
 5.2. Layered Encoding 8
 5.3. Interactions of Video Streaming with ICN 9
   5.3.1. Interaction of DASH and ICN 9
   5.3.2. Interaction of ICN with Layered Encoding 11
 5.4. Possible Integration of Video streaming and ICN architecture .. 12
   5.4.1. DASH over CCN 12
   5.4.2. Testbed, Open Source Tools, and Dataset 14
6. P2P video distribution and ICN.................................. 15
 6.1. Introduction to PPSP 15
 6.2. PPSP over ICN: deployment concepts 16
   6.2.1. PPSP short background 16
   6.2.2. From PPSP messages to ICN named-data 17
   6.2.3. Support of PPSP interaction through a pull-based ICN API .. 18
   6.2.4. Abstract layering for PPSP over ICN 18
   6.2.5. PPSP interaction with the ICN routing plane 19
   6.2.6. ICN deployment for PPSP 20
 6.3. Impact of MPEG DASH coding schemes 21
7. IPTV and ICN.. 22
 7.1. IPTV challenges 22
 7.2. ICN benefits for IPTV delivery 23
8. Digital Rights Managements in ICN............................... 25
 8.1. Broadcast Encryption for DRM in ICN 25
 8.2. AAA Based DRM for ICN Networks 28
   8.2.1. Overview 28
   8.2.2. Implementation 29

9. Future Steps for Video in ICN.................................... 30
9.1. Large Scale Live Events 30
9.2. Video Conferencing and Real-Time Communications 30
9.3. Store-and-Forward Optimized Rate Adaptation 30
9.4. Heterogeneous Wireless Environment Dynamics 31
9.5. Network Coding for Video Distribution in ICN 33
9.6. Synchronization Issues for Video Distribution in ICN 33
10. Security Considerations.. 34
11. IANA Considerations.. 34
12. Conclusions.. 34
13. References... 35
13.1. Normative References 35
13.2. Informative References 35
14. Authors' Addresses... 38
15. Acknowledgements... 39

1. Introduction

The unprecedented growth of video traffic has triggered a rethinking of how content is distributed, both in terms of the underlying Internet architecture and in terms of the streaming mechanisms to deliver video objects.

In particular, the IRTF ICNRG research group has been chartered to study new architectures centered upon information; the main contributor to Internet traffic (and information dissemination) is video, and this is expected to stay the same in the near future. If ICN is expected to become prominent, it will have to support video streaming efficiently.

As such, it is necessary to discuss along two directions:

- Can the current video streaming mechanisms be leveraged and adapted to an ICN architecture?

- Can (and should) new, ICN-specific video streaming mechanisms be designed to fully take advantage of the new abstractions exposed by the ICN architecture?

This document intends to focus on the first question, in an attempt to define the use cases for video streaming and some requirements.

This document focuses on a few scenarios, namely Netflix-like video streaming, peer-to-peer video sharing and IPTV, and identifies how the existing protocols can be adapted to an ICN architecture. In doing so, it also identifies the main issues with these protocols in this ICN context.

Some documents have started to consider the ICN-specific requirements of dynamic adaptive streaming [2][3][4][6].

In this document, we give a brief overview of the existing solutions for the selected scenarios. We then consider the interactions of such existing mechanisms with the ICN architecture and list some of the interactions any video streaming mechanism will have to consider. We then identify some areas for future research.

2. Conventions used in this document

In examples, "C:" and "S:" indicate lines sent by the client and server respectively.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

3. Use case scenarios for ICN and Video Streaming

For ICN specific descriptions, we refer to the other research group documents. For our purpose, we assume here that ICN means an architecture where content is retrieved by name and with no binding of content to a specific network location.

The consumption of multimedia content comes along with timing requirements for the delivery of the content, for both, live and on-demand consumption. Additionally, real-time use cases such as audio-/video conferencing [7], game streaming, etc., come along with more strict timing requirements. Long startup delays, buffering periods or poor quality, etc., should be avoided to achieve a good Quality of Experience (QoE) to the consumer of the content. (For a definition of QoE in the context of video distribution, please refer to [25]. The working definition is: "Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user’s personality and current state.")

Of course, these requirements are heavily influenced by routing decisions and caching, which are central parts of ICN and which have to be considered when streaming video in such infrastructures.

Due to this range of requirements, we find it useful to narrow the focus on four scenarios (more can be included later):

- a video download architecture similar to that of Apple iTtunes(R), where the whole file is being downloaded to the client and can be replayed there multiple times;
- a video streaming architecture for playing back movies; this is relevant for the naming and caching aspects of ICN, as well as the interaction with the rate adaptation mechanism necessary to deliver the best QoE to the end-user;
- a peer-to-peer architecture for sharing videos; this introduces more stringent routing requirements in terms of locating copies of the content, as the location of the peers evolves and peers join and leave the swarm they use to exchange video chunks (for Peer-to-Peer definitions and taxonomy, please refer to RFC5694);
- IPTV; this introduces requirements for multicasting and adds stronger delay constraints.

Other scenarios, such as video-conferencing and real-time video communications are not explicitly discussed in this document, while they are in scope. Also, events of mass-media distribution, such as a large crowd in a live event, are also adding new requirements to be included in later version.

We discuss how the current state-of-the-art protocols in an IP context can be modified for the ICN architecture. The remainder of this document is organized as follows. In the next section, we consider video download. Then in Section 5, we briefly describe DASH [1], and Layered Encoding (MDC, SVC). P2P is the focus of Section 6, where we describe PPSP. Section 7 highlights the requirements of IPTV, while Section 8 describes the issues of DRM. Section 9 lists some research issues to be solved for ICN-specific video delivery mechanisms.

Videoconferencing and real-time video communications will be detailed more in future versions of this document; as well as the mass distribution of content at live large-scale events (stadium, concert hall, etc) for which there is no clearly adopted existing protocol.

4. Video download

Video download, namely the fetching of a video file from a server or a cache down to the user's local storage, is a natural application of ICN. It should be supported natively without requiring any specific considerations.

This is supported now by a host of protocols (say, SCP, FTP, or over HTTP), which would need to be replaced by the protocols to retrieve content in ICNs.

However, current mechanisms are built atop existing transport protocols. Some ICN proposals (say, CCN or NDN for instance) attempt to leverage the work done upon these transport protocol and it has been proposed to use mechanisms such as the TCP congestion window (and the associated Adaptive Increase, Multiplicative Decrease - AIMD) to decide how many object requests ("interests" in CCN/NDN terminology) should be in flight at any point in time.

It should be noted that ICN intrinsically supports different transport mechanisms, which could achieve better performance than TCP, as they subsume TCP into a special case. For instance, one could imagine a link-by-link transport coupled with caching. This is enabled by the ICN architecture, and would facilitate the point-to-point download of video files.

5. Video streaming and ICN

5.1. Introduction to client-driven streaming and DASH

Media streaming over the hypertext transfer protocol (HTTP) and in a further consequence streaming over the transmission control protocol (TCP) has become omnipresent in today's Internet. Content providers such as Netflix, Hulu, and Vudu do not deploy their own streaming equipment but use the existing Internet infrastructure as it is and they simply deploy their own services over the top (OTT). This streaming approach works surprisingly well without any particular support from the underlying network due to the use of efficient video compression, content delivery networks (CDNs), and adaptive video players. Earlier video streaming research mostly recommended to use the user datagram protocol (UDP) combined with the real time transport protocol (RTP). It assumed it would not be possible to transfer multimedia data smoothly with TCP, because of its throughput variations and large retransmission delays. This point of view has significantly evolved today. HTTP streaming, and especially its most simple form known as progressive download, has become very popular over the past few years because it has some major benefits compared to RTP streaming. As a consequence of the consistent use of HTTP for this streaming method, the existing Internet infrastructure, consisting of proxies, caches and CDNs, could be used. Originally, this architecture was designed to support best effort delivery of files and not real time transport of multimedia data. Nevertheless, real time streaming based on HTTP could also take advantage of this architecture, in comparison to RTP, which could not leverage any of the aforementioned components. Another benefit that results from the use of HTTP is that the media stream could easily pass firewalls or network address translation (NAT) gateways, which was definitely a key for the success of HTTP streaming. However, HTTP streaming is not the holy grail of streaming as it also introduces some drawbacks compared to RTP. Nevertheless, in an ICN-based video streaming architecture these aspects also have to be considered.

The basic concept of DASH [1] is to use segments of media content, which can be encoded at different resolutions, bit rates, etc., as so-called representations. These segments are served by conventional HTTP Web servers and can be addressed via HTTP GET requests from the client. As a consequence, the streaming system is pull-based and the entire streaming logic is located on the client, which makes it scalable, and allows to adapt the media stream to the client's capabilities.

In addition to this, the content can be distributed using conventional CDNs and their HTTP infrastructure, which also scales very well. In order to specify the relationship between the contents' media segments and the associated bit rate, resolution, and timeline, the Media Presentation Description (MPD) is used, which is a XML document. The MPD refers to the available media segments using HTTP URLs, which can be used by the client for retrieving them.

5.2. Layered Encoding

Another approach for video streaming consist in using layered encoding. Namely, scalable video coding formats the video stream into different layers: a base layer which can be decoded to provide the lowest bit rate for the specific stream, and enhancement layers which can be transmitted separately if network conditions allow. The higher layers offer higher resolutions and enhancement of the video quality, while the layered approach allows to adapt to the network conditions. This is used in MPEG-4 scalable profile or H.263+. H264SVC is available, but not much deployed. JPEG2000 has a wavelet transform approach for layered encoding, but has not been deployed much either.

It is not clear if the layered approach is fine-grained enough for rate control.

5.3. Interactions of Video Streaming with ICN

5.3.1. Interaction of DASH and ICN

Video streaming, and DASH in particular, have been designed with goals that are aligned with that of most ICN proposals. Namely, it is a client-based mechanism, which requests items (in this case, chunks of a video stream) by name.

ICN and MPEG-DASH [1] have several elements in common:

- the client-initiated pull approach;
- the content being dealt with in pieces (or chunks);
- the support of efficient replication and distribution of content pieces within the network;
- the scalable, session-free nature of the exchange between the client and the server at the streaming layer: the client is free to request any chunk from any location;
- the support for potentially multiple source locations.

For the last point, DASH may list multiple source URLs in a manifest, and ICN is agnostic to the location of a copy it is receiving. We do not imply that current video streaming mechanisms attempt to draw the content from multiple sources concurrently. This is a potential benefit of ICN, but is not considered in the current approaches mentioned in this document.

As ICN is a promising candidate for the Future Internet (FI) architecture, it is useful to investigate its suitability in combination with multimedia streaming standards like MPEG-DASH. In this context, the purpose of this section is to present the usage of ICN instead of HTTP in MPEG-DASH

However, there are some issues that arise from using a dynamic rate adaptation mechanism in an ICN architecture (note that some of the issues are related to caching, and not necessarily unique to ICN):

o Naming of the data in DASH does not necessarily follow the ICN convention of any of the ICN proposals. Several chunks of the same video stream might currently go by different names that for instance do not share a common prefix. There is a need to harmonize the naming of the chunks in DASH with the naming conventions of the ICN. The naming convention of using a filename/time/encoding format could for instance be made compatible with the convention of CCN.

o While chunks can be retrieved from any server, the rate adaptation mechanism attempts to estimate the available network bandwidth so as to select the proper playback rate and keep its playback buffer at the proper level. Therefore, there is a need to either include some location semantics in the data chunks so as to properly assess the throughput to a specific location; or to design a different mechanism to evaluate the available network bandwidth.

o The typical issue of access control and accounting happens in this context, where chunks can be cached in the network outside of the administrative control of the content publisher. It might be a requirement from the owner of the video stream that access to these data chunks needs to be accounted/billed/monitored.

o Dynamic streaming multiplies the representations of a given video stream, therefore diminishing the effectiveness of caching: namely, to get a hit for a chunk in the cache, it has to be for the same format and encoding values. Alternatively, to get the same hit rate as for a stream using a single encoding, the cache size must be scaled up to include all the possible representations.

o Caching introduces oscillatory dynamics as it may modify the estimation of the available bandwidth between the end user and the repository where it is getting the chunks from. For instance, if an edge cache holds a low resolution representation near the user, the user getting this low resolution chunks will observe a good performance, and will then request higher resolution chunks. If those are hosted on a server with poor performance, then the client would have to switch back to the low representation. This oscillation may be detrimental to the perceived QoE of the user.

o The ICN transport mechanism needs to be compatible to some extent with DASH. To take a CCN example, the rate at which interests are issued should be such that the chunks received in return arrive fast enough and with the proper encoding to keep the playback buffer above some threshold.

- o The usage of multiple network interfaces is possible in ICN, enabling a seamless handover between them. For the combination with DASH, an intelligent strategy which should focus on traffic load balancing between the available links may be necessary. This would increase the effective media throughput of DASH by leveraging the combined available bandwidth of all links, however, it could potentially lead to high variations of the media throughput.

- o DASH does not define how the MPD is retrieved; hence, this is compatible with CCN. However, the current profiles defined within MPEG-DASH require the MPD to contain HTTP-URLs (incl. http and https URI schemes) to identify segments. To enable a more integrated approach as described in this document, an additional profile for DASH over CCN has to be defined, enabling ICN/CCN-based URIs to identify and request the media segments.

We describe in Section 5.4 a potential implementation of a dynamic adaptive video stream over ICN, based upon DASH and CCN [5].

5.3.2. Interaction of ICN with Layered Encoding

Issues of interest to an Information-Centric network architecture in the context of layered video streaming include:

- . Caching of the multiple layers. The caching priority should go to the base layer, and defining caching policy to decide when to cache enhancement layers;
- . Synchronization of multiple content streams, as the multiple layers may come from different sources in the network (for instance, the base layer might be cached locally while the enhancement layers may be stored in the origin server). Video and audio video streams must be synchronized, and this includes both intra-layer synchronization (for the layers of the same video or audio stream) and inter-stream synchronization (see Section 9 for other synchronization aspects to be included in the "Future Steps for Video in ICN");
- . Naming of the different layers: when the client requests an object, the request can be satisfied with the base layer alone, aggregated with enhancement layers. Should one request be sufficient to provide different streams? In a CCN architecture for instance, this would violate a one interest-one data packet principle and the client would need to specify each layer it would like to receive. In a Pub/Sub architecture, the rendezvous point would have to make a decision as to which layers (or which pointer to which layer's location) to return.

5.4. Possible Integration of Video streaming and ICN architecture

5.4.1. DASH over CCN

DASH is intended to enable adaptive streaming, i.e., each content piece can be provided in different qualities, formats, languages, etc., to cope with the diversity of todays' networks and devices. As this is an important requirement for Future Internet proposals like CCN, the combination of those two technologies seems to be obvious. Since those two proposals are located at different protocol layers - DASH at the application and CCN at the network layer - they can be combined very efficiently to leverage the advantages of both and potentially eliminate existing disadvantages. As CCN is not based on classical host-to-host connections, it is possible to consume content from different origin nodes as well as over different network links in parallel, which can be seen as an intrinsic error resilience feature w.r.t. the network. This is a useful feature of CCN for adaptive multimedia streaming within mobile environments since most mobile devices are equipped with multiple network links like 3G and WiFi. CCN offers this functionality out of the box which is beneficial when used for DASH-based services. In particular, it is possible to enable adaptive video streaming handling both bandwidth and network link changes. That is, CCN handles the network link decision and DASH is implemented on top of CCN to adapt the video stream to the available bandwidth.

In principle, there are two options to integrate DASH and CCN: a proxy service acting as a broker between HTTP and CCN as proposed in [6], and the DASH client implementing a native CCN interface. The former transforms an HTTP request to a corresponding interest packet as well as a data packet back to an HTTP response, including reliable transport as offered by TCP. This may be a good compromise to implement CCN in a managed network and to support legacy devices. Since such a proxy is already described in [6] this draft focuses on a more integrated approach, aiming at fully exploiting the potential of a CCN DASH Client. That is, we describe a native CCN interface within the DASH client, which adopts a CCN naming scheme (CCN URIs) to denote segments in the Media Presentation Description (MPD). In this architecture, only the network access component on the client has to be modified and the segment URIs within MPD have to be updated according to the CCN naming scheme.

Initially, the DASH client retrieves the MPD containing the CCN URIs of the content representations including the media segments. The naming scheme of the segments may reflect intrinsic features of CCN like versioning and segmentation support. Such segmentation support is already compulsory for multimedia streaming in CCN and, thus, can also be leveraged for DASH-based streaming over CCN. The CCN versioning can be adopted in a further step to signal different representations of the DASH-based content, which enables an implicit adaptation of the requested content to the clients' bandwidth conditions. That is, the interest packet already provides the desired characteristics of a segment (such as bit rate, resolution, etc.) within the content name (or potentially within parameters defined as extra types in the packet formats). Additionally, if bandwidth conditions of the corresponding interfaces or routing paths allow so, DASH media segments could be aggregated automatically by the CCN nodes, which reduces the amount of interest packets needed to request the content. However, such approaches need further research, specifically in terms of additional intelligence and processing power needed at the CCN nodes.

After requesting the MPD, the DASH client will start to request particular segments. Therefore, CCN interest packets are generated by the CCN access component and forwarded to the available interfaces. Within the CCN, these interest packets leverage the efficient interest aggregation for, e.g., popular content, as well as the implicit multicast support. Finally, the interest packets are satisfied by the corresponding data packets containing the video segment data, which are stored on the origin server or any CCN node, respectively. With an increasing popularity of the content, it will be distributed across the network resulting in lower transmission delays and reduced bandwidth requirements for origin servers and content providers respectively.

With the extensive usage of in-network caching, new drawbacks are introduced since the streaming logic is located at the client, i.e., clients are not aware of each other and the network infrastructure and cache states. Furthermore, negative effects are introduced when multiple clients are competing for a bottleneck and when caching is influencing this bandwidth competition. As mentioned above, the clients request individual portions of the content based on available bandwidth, which is calculated using throughput estimations. This uncontrolled distribution of the content influences the adaptation process of adaptive streaming clients. The impact of this falsified throughput estimation could be tremendous and leads to a wrong adaptation decision which may impact the Quality of Experience (QoE) at the client, as shown in [8]. In ICN, the client does not have the knowledge from which source the requested content is actually served or how many origin servers of the content are available, as this is transparent and depends on the name-based routing. This introduces the challenge that the adaptation logic of the adaptive streaming client is not aware of the event when the ICN routing decides to switch to a different origin server or content is coming through a different link/interface. As most algorithms implementing the adaption logic are using bandwidth measurements and related heuristics, the adaptation decisions are no longer valid when changing origin servers (or links) and potentially cause playback interruptions and, consequently, stalling. Additionally, ICN supports the usage of multiple interfaces. A seamless handover between these interfaces (and different sources for the content) comes together with changes in performance, e.g., due to switching between fixed and wireless, 3G/4G and WiFi networks, or between different types of servers (say with/without SSD, or with/without hardware acceleration), etc.

Considering these characteristics of ICN, adaptation algorithms merely based on bandwidth measurements are not appropriate anymore, as potentially each segment can be transferred from another ICN node or interface, all with different bandwidth conditions. Thus, adaptation algorithms taking into account these intrinsic characteristics of ICN are preferred over algorithms based on mere bandwidth measurements.

5.4.2. Testbed, Open Source Tools, and Dataset

For the evaluations of DASH over CCN, a testbed with open source tools and datasets is provided in [9]. In particular, it provides two client player implementations, (i) a libdash extension for DASH over CCN and (ii) a VLC plugin implementing DASH over CCN. For both implementations the CCNx implementation has been used as a basis.

The general architecture of libdash is organized in modules, so that the library implements a MPD parser and an extensible connection manager. The library provides object-oriented interfaces for these modules to access the MPD and the downloadable segments. These components are extended to support DASH over CCN and available in a separate development branch of the github project available at http://www.github.com/bitmovin/libdash. libdash comes together with a fully featured DASH player with a QT-based frontend, demonstrating the usage of libdash and providing a scientific evaluation platform. As an alternative, patches for the DASH plugin of the VLC player are provided. These patches can be applied to the latest source code checkout of VLC resulting in a DASH over CCN-enabled VLC player.

Finally, a DASH over CCN dataset is provided in form of a CCNx repository. It includes 15 different quality representation of the well-known Big Buck Bunny Movie, ranging from 100 kbps up to 4500 kbps. The content is split into segments of two seconds, and described by an associated MPD using the presented naming scheme in Section 4.1. This repository can be downloaded from [9], and is also provided by a public accessible CCNx node. Associated routing commands for the CCNx namespaces of the content are provided via scripts coming together with the dataset and can be used as a public testbed.

6. P2P video distribution and ICN

Another form of distributing content - and video in particular- which ICNs need to support is Peer-to-Peer distribution (P2P). We see now how an existing protocol such as PPSP can be modified to work in an ICN environment.

6.1. Introduction to PPSP

P2P video Streaming (PPS) is a popular approach to redistribute live media over Internet. The proposed P2PVS solutions can be roughly classified in two classes:

- Push/Tree based

- Pull/Mesh based

The Push/Tree based solution creates an overlay network among peers that has a tree shape [30]. Using a progressive encoding (e.g. Multiple Description Coding or H.264 Scalable Video Coding), multiple trees could be set up to support video rate adaptation. On each tree an enhancement stream is sent. The higher the number of received streams, the higher the video quality. A peer controls the video rate by fetching or not the streams delivered over the distribution trees.

The Pull/Mesh based solution is inspired by the BitTorrent file sharing mechanism. A Tracker collects information about the state of the swarm (i.e. set of participating peers). A peer forms a mesh overlay network with a subset of peers, and exchange data with them. A peer announces what data items it disposes and requests missing data items that are announced by connected peers. In case of live streaming, the involved data set includes only a recent window of data items published by the source. Also in this case, the use of a progressive encoding can be exploited for video rate adaptation.

Pull/Mesh based P2PVS solutions are the more promising candidate for the ICN deployment, since most of ICN approach provides a pull-based API [5][10][11][12]. In addition, Pull/Mesh based P2PVS are more robust than Push/Tree based one [13] and the Peer to Peer Streaming Protocol (PPSP) working group [14] is also proposing a Pull/Mesh based solution.

```
+------------------------------------------------+
|                                                |
|     +--------------------------------+         |
|     |            Tracker             |         |
|     +--------------------------------+         |
|        |     ^                   ^             |
|Tracker |     | Tracker           |Tracker      |
|Protocol|     | Protocol          |Protocol     |
|        |     |                   |             |
|        V     |                   |             |
|     +---------+    Peer     +---------+        |
|     |   Peer  |<----------->|   Peer  |        |
|     +---------+   Protocol  +---------+        |
|       | ^                                      |
|       | |Peer                                  |
|       | |Protocol                              |
|       V |                                      |
|     +---------------+                          |
|     |      Peer     |                          |
|     +---------------+                          |
|                                                |
+------------------------------------------------+
```

Figure 1: PPSP System Architecture (source [RFC6972])

Figure 1 reports the PPSP architecture presented in [RFC6972]. PEERs announce and share video chunks and a TRACKER maintains a list of PEERs participating in a specific audio/video channel or in the distribution of a streaming file. The tracker functionality may be centralized in a server or distributed over the PEERs. PPSP standardize the Peer and Tracker Protocols, which can run directly over UDP or TCP.

This document discusses some preliminary concepts about the deployment of PPSP on top of an ICN that exposes a pull-based API, meanwhile considering the impact of MPEG DASH streaming format.

6.2. PPSP over ICN: deployment concepts

6.2.1. PPSP short background

PPSP specifies peer protocol (PPSPP) [15] and tracker protocol (PPSP-TP)[16].

Some of the operations carried out by the tracker protocol are the followings. When a peer wishes to join the streaming session it contacts the Tracker (CONNECT message), obtains a PEER_ID and a list of PEER_IDs (and IP addresses) of other peers that are participating to the SWARM and that the tracker has singled out for the requesting peer (this may be a subset of the all peers of the SWARM). In addition to this join operation, a peer may contact the tracker to request to renew the list of participating peers (FIND message), to periodically update its status to the tracker (STAT_REPORT message), etc.

Some of the operations carried out by the peer protocol are the following. Using the list of peers delivered by the tracker, a peer establishes a session with them (HANDSHAKE message). A peer periodically announces to neighboring peers which chunks it has available for download (HAVE message). Using these announcements, a peer requests missing chunks from neighboring peers (REQUEST messages), which will send back them (DATA message).

6.2.2. From PPSP messages to ICN named-data

An ICN provides users with data items exposed by names. The bundle name and data item is usually referred as named-data, named-content, etc. To transfer PPSP messages though an ICN the messages should be wrapped as named-data items, and receivers should request them by name.

A PPSP entity receives messages from peers and/or tracker. Some operations require gathering the messages generated by another specific host (peer or tracker). For instance, if a peer A wishes to gain information about video chunks available from peer B, the former shall fetch the PPSP HAVE messages specifically generated by the later. We refer to these kinds of named-data as "located-named-data", since they should be gathered from a specific location (e.g. peer B).

For other PPSP operations, such as fetching a DATA message (i.e. a video chunk), as long as a peer receives the requested content, it doesn’t matter which endpoint generated the data.We refer to this information with the generic term "named-data".

The naming scheme differentiates named-data and located-named-data items. In case of named-data, the naming scheme only includes a content identifier (e.g. the name of the video chunk), without any prefix identifying who provides the content. For instance, a DATA message containing the video chunk #1 may be named as "ccnx:/swarmID/chunk/chunkID", where swarmID is a unique identifier of the streaming session, "chunk" is a keyword and chunkID is the chunk identifier (e.g. a integer number).

In case of located-named-data, the naming scheme includes a location-prefix, which uniquely identifies the host generating the data item. This prefix may be the PEER_ID in case the host was a peer or a tracker identifier in case the host was the tracker. For instance, a HAVE message generated by a peer B may be named as "ccnx:/swarmID/peer/PEER_ID/HAVE", where "peer" is a keyword, PEER_ID_B is the identifier of peer B and HAVE is a keyword.

6.2.3. Support of PPSP interaction through a pull-based ICN API

The PPSP procedures are based both on pull and push interactions. For instance, the distribution of chunks availability can be classified as a push-based operation, since a peer sends an "unsolicited" information (HAVE message) to neighboring peers. Conversely the procedure used to receive video chunks can be classified as pull-based, since it is supported by a request/response interaction (i.e. REQUEST, DATA messages).

As we said, we refer to an ICN architecture which provides a pull-based API. Accordingly, the mapping of PPSP pull-based procedure is quite simple. For instance, using the CCN architecture [5] a PPSP DATA message may be carried by a CCN Data message and a REQUEST message can transferred by a CCN Interest.

Conversely, the support of push-based PPSP operations may be more difficult. We need of an adaptation functionality that carries out a push-based operation using the underlying pull-based service primitives. For instance, a possible approach is to use the request/response (i.e. Interest/Data) four ways handshakes proposed in [7]. Another possibility is that receivers periodically send out request messages of the named-data that neighbors will push and, when available, sender inserts the pushed data within a response message.

6.2.4. Abstract layering for PPSP over ICN

```
+------------------------------------+
|      Application                   |
+------------------------------------+
|        PPSP (TCP/IP)               |
+------------------------------------+
|  ICN - PPSP Adaptation Layer (AL)  |
+------------------------------------+
```

```
                   |       ICN Architecture             |
                   +------------------------------------+
```

Figure 2: Mediator approach

Figure 2 provides a possible abstract layering for PPSP over ICN. The Adaptation Layer acts as a mediator (proxy) between legacy PPSP entities based on TCP/IP and the ICN architecture. In facts, the role the mediator is to use ICN to transfer PPSP legacy messages.

This approach makes possible to merely reuse TCP/IP P2P applications whose software includes also PPSP functionality. This "all-in-one" development approach may be rather common since the PPSP-Application interface is not going to be specified. Moreover, if the Operating System will provide libraries that expose a PPSP API, these will be initially based on a underlying TCP/IP API. Also in this case, the mediator approach would make possible to easily reuse both the PPSP libraries and the Application on top of an ICN.

```
                   +------------------------------------+
                   |      Application                   |
                   +------------------------------------+
                   |        ICN-PPSP                    |
                   +------------------------------------+
                   |       ICN Architecture             |
                   +------------------------------------+
```

Figure 3: Clean-slate approach

Figure 3 sketches a clean-slate layering approach in which the application directly includes or interacts with a PPSP version based on ICN. Likely such a PPSP_ICN integration could yield a simplier development, also because it does not require implementing a TCP/IP to ICN translation as in the Mediator approach. However, the clean-slate approach requires developing the application (in case of embedded PPSP functionality) or the PPSP library from scratch, without exploiting what might already exist for TCP/IP.

Overall, the Mediator approach may be considered as the first step of a migration path towards ICN native PPSP applications.

6.2.5. PPSP interaction with the ICN routing plane

Upon the ICN API a user (peer) requests a content and the ICN sends it back. The content is gathered by the ICN from any source, which could be the closest peer that disposes of the named-data item, an in-network cache, etc. Actually, "where" to gather the content is controlled by an underlying ICN routing plane, which sets up the ICN forwarding tables (e.g. CCN FIB [5]).

A cross-layer interaction between the ICN routing plane and the PPSP may be required to support a PPSP session. Indeed, ICN shall forward request messages (e.g. CCN Interest) towards the proper peer that can handle them. Depending on the layering approach, this cross-layer interaction is controlled either by the Adaptation Layer or by the ICN-PPSP. For example, if a peer A receives a HAVE message indicating that peer B disposes of the video chunk named "ccnx:/swarmID/chunk/chunkID", then former should insert in its ICN forwarding table an entry for the prefix "ccnx:/swarmID/chunk/chunkID" whose next hop locator (e.g. IP address) is the network address of peer B [17].

6.2.6. ICN deployment for PPSP

The ICN functionality that supports a PPSP session may be "isolated" or "integrated" with the one of a public ICN.

In the isolated case, a PPSP session is supported by an instance of an ICN (e.g. deployed on top of IP), whose functionalities operate only on the limited set of nodes participating to the swarm, i.e. peers and the tracker. This approach resembles the one followed by current P2P application, which usually form an overlay network among peers of a P2P application. And intermediate public IP routers do not carry out P2P functionalities.

In the integrated case, the nodes of a public ICN may be involved in the forwarding and in-network caching procedures. In doing so, the swarm may benefit from the presence of in-network caches so limiting uplink traffic on peers and inter-domain traffic too. These are distinctive advantages of using PPSP over a public ICN, rather than over TCP/IP. In addition, such advantages aren’t likely manifested in the case of isolated deployment.

However, the possible interaction between the PPSP and the routing layer of a public ICN may be dramatic, both in terms of explosion of the forwarding tables and in terms of security. These issues specifically take place for those ICN architectures for which the name resolution (i.e. name to next-hop) occurs en-route, like the CCN architecture.

For instance, using the CCN architecture, to fetch a named-data item offered by a peer A the on-path public ICN entities have to route the request messages towards the peer A. This implies that the ICN forwarding tables of public ICN nodes may contain many entries, e.g.

one entry per video chunk, and these entries are difficult to be aggregated since peers may have available only sparse parts of a big content, whose names have a same prefix (e.g. "ccnx:/swarmID"). Another possibility is to wrap all PPSP messages into a located-named-data. In this case the forwarding tables should contain "only" the PEER_ID prefixes (e.g. "ccnx:/swarmID/peer/PEER_ID"), so scaling down the number of entries from number of chunks to number of peers. However, in this case the ICN mechanisms recognize a same video chunk offered by different peers as different contents, so vanishing caching and multicasting ICN benefits. Moreover, in any case routing entries should be updated either the base of the availability of named-data items on peers or on the presence of peers, and these events in a P2P session is rapidly changing so possibly hampering the convergence of the routing plane. Finally, since peers have an impact on the ICN forwarding table of public nodes, this may open obvious security issues.

6.3. Impact of MPEG DASH coding schemes

The introduction of video rate adaptation may significantly decrease the effectiveness of P2P cooperation and of in-network caching, depending of the kind of the video coding used by the MPEG DASH stream.

In case of a MPEG DASH streaming with MPEG AVC encoding, a same video chunk is independently encoded at different rates and the encoding output is a different file for each rate. For instance, in case of a video encoded at three different rates R1,R2,R3, for each segment S we have three distinct files: S.R1, S.R2, S.R3. These files are independent of each other. To fetch a segment coded at R2 kbps, a peer shall request the specific file S.R2. Receiver-driven algorithms, implemented by the video client, usually handle the estimation of the best coding rate.

The independence among files associated to different encoding rates and the heterogeneity of peer bandwidths, may dramatically reduce the interaction among peers, the effectiveness of in-network caching (in case of integrated deployment), and consequently the ability of PPSP to offload the video server (i.e. a seeder peer). Indeed, a peer A may select a coding rate (e.g. R1) different from the one selected by a peer B (e.g. R2) and this prevents the former to fetch video chunks from the later, since peer B only has chunks available that are coded at a rate different from the ones needed by A. To overcome this issue, a common distributed rate selection algorithm could force peers to select the same coding rate [17]; nevertheless this approach may be not feasible in the in case of many peers.

The use of SVC encoding (Annex G extension of the H.264/MPEG-4 AVC video compression standard) should make rate adaptation possible, meanwhile neither reducing peer collaborations nor the in-network caching effectiveness. For a single video chunk, a SVC encoder produces different files for the different rates (roughly "layers"), and these files are progressively related each other. Starting from a base-layer which provides the minimum rate encoding, the next rates are encoded as an "enhancement layer" of the previous one. For instance, in case the video is coded with three rates R1 (base-layer), R2 (enhancement-layer n.1), R3 (enhancement-layer n.2), then for each DASH segment we have three files S.R1, S.R2 and S.R3. The file S.R1 is the segment coded at the minimum rate (base-layer). The file S.R2 enhances S.R1, so as S.R1 and S.R2 can be combined to obtain a segment coded at rate R2. To get a segment coded at rate R2, a peer shall fetch both S.R1 and S.R2. This progressive dependence among files that encode a same segment at different rates makes peer cooperation possible, also in case peers player have autonomously selected different coding rates. For instance, if peer A has selected the rate R1, the downloaded files S.R1 are useful also for a peer B that has selected the rate R2, and vice versa.

7. IPTV and ICN

7.1. IPTV challenges

IPTV refers to the delivery of quality content broadcast over the Internet, and is typically associated with strict quality requirements, i.e., with a perceived latency of less than 500 ms and a packet loss rate that is multiple orders lower than the current loss rates experienced in the most commonly used access networks (see [31]). We can summarize the major challenges for the delivery of IPTV service as follows.

Channel change latency represents a major concern for the IPTV service. Perceived latency during channel change should be less than 500ms. To achieve this objective over the IP infrastructure, we have multiple choices:

(i) receiving fast unicast streams from a dedicated server (most effective but not resource efficient);
(ii) connecting to other peers in the network (efficiency depends on peer support, effective and resource efficient, if also supported with a dedicated server);

(iii) connecting to multiple multicast sessions at once (effective but not resource efficient, and depends on the accuracy of the prediction model used to track user activity).

The second major challenge is the error recovery. Typical IPTV service requirements dictate the mean time between artifacts to be approximately 2 hours (see [31]). This suggests the perceived loss rate to be around or less than 10^-7. Current IP-based solutions rely on the following proactive and reactive recovery techniques: (i) joining the FEC multicast stream corresponding to the perceived packet loss rate (not efficient as the recovery strength is chosen based on worst-case loss scenarios), (ii) making unicast recovery requests to dedicated servers (requires active support from the service provider), (iii) probing peers to acquire repair packets (finding matching peers and enabling their cooperation is another challenge).

7.2. ICN benefits for IPTV delivery

ICN presents significant advantages for the delivery of IPTV traffic. For instance, ICN inherently supports multicast and allows for quick recovery from packet losses (with the help of in-network caching). Similarly, peer support is also provided in the shape of in-network caches that typically act as the middleman between two peers, enabling therefore earlier access to IPTV content.

However, despite these advantages, delivery of IPTV service over Information Centric Networks brings forth new challenges. We can list some of these challenges as follows:

- Messaging overhead: ICN is a pull-based architecture and relies on a unique balance between requests and responses. A user needs to make a request for each data packet. In the case of IPTV, with rates up to, and likely to be, above 15Mbps, we observe significant traffic upstream to bring those streams. As the number of streams increases (including the same session at different quality levels and other formats), so does the burden on the routers. Even if the majority of requests are aggregated at the core, routers close to the edge (where we observe the biggest divergence in user requests) will experience a significant increase in overhead to process these requests. The same is true at the user side, as the uplink usage multiplies in the number of sessions a user requests (for instance, to minimize the impact of bandwidth fluctuations).

- Cache control: As the IPTV content expires at a rapid rate (with a likely expiry threshold of 1s), we need solutions to effectively flush out such content to also prevent degradation impact on other cached content, with the help of intelligently chosen naming conventions. However, to allow for fast recovery and optimize access time to sessions (from current or new users), the timing of such expirations needs to be adaptive to network load and user demand. However, we also need to support quick access to earlier content, whenever needed, for instance, when the user accesses the rewind feature (note that in-network caches will not be of significant help in such scenarios due to overhead required to maintain such content).
- Access accuracy: To receive the up-to-date session data, users need to be aware of such information at the time of their request. Unlike IP multicast, since the users join a session indirectly, session information is critical to minimize buffering delays and reduce the startup latency. Without such information, and without any active cooperation from the intermediate routers, stale data can seriously undermine the efficiency of content delivery. Furthermore, finding a cache does not necessarily equate to joining a session, as the look-ahead latency for the initial content access point may have a shorter lifetime than originally intended. For instance, if the user that has initiated the indirect multicast leaves the session early, the requests from the remaining users need to experience an additional latency of one RTT as they travel towards the content source. If the startup latency is chosen depending on the closeness to the intermediate router, going to the content source in-session can lead to undesired pauses.

It should be noted that IPTV includes more than just multicast. Many implementations include "trick plays" (fast forward, pause, rewind) that often transform a multicast session into multiple unicast sessions. In this context, ICN is beneficial, as the caching offers an implicit multicast, but without tight synchronization constraints in between two different users. One user may rewind, and start playing forward again, drawing from a nearby cache of the content recently viewed by another user (whereas in a strict multicast session, the opportunity of one user lagging off behind would be more difficult to implement).

8. Digital Rights Managements in ICN

This section discusses the need for Digital Rights Management (DRM) functionalities for multimedia streaming over ICN. It focuses on two possible approaches: modifying AAA to support DRM in ICN, and using Broadcast Encryption.

It is assumed that ICN will be used heavily for digital content dissemination. It is vital to consider DRM for digital content distribution. In today's Internet there are two predominant classes of business models for on-demand video streaming. The first model is based on advertising revenues. Non-copyright protected (usually user-generated content, UGC) is offered by large infrastructure providers like Google (YouTube) at no charge. The infrastructure is financed by spliced advertisements into the content. In this context DRM considerations may not be required, since producers of UGC may only strive for the maximum possible dissemination. Some producers of UGC are mainly interested to share content with their families, friends, colleges or others and have no intention to make profit. However, the second class of business models requires DRM, because they are primarily profit oriented. For example, large on-demand streaming platforms like Netflix establish business models based on subscriptions. Consumers may have to pay a monthly fee in order to get access to copyright protected content like TV series, movies or music. This model may be ad-supported and free to the content consumer, like YouTube Channels or Spotify. But the creator of the content expects some remuneration for his work. From the perspective of the service providers and the copyright owners, only clients that pay the fee (explicitly or implicitly through ad placement) should be able to access and consume the content. Anyway, the challenge is to find an efficient and scalable way of access control to digital content, which is distributed in information-centric networks.

8.1. Broadcast Encryption for DRM in ICN

The section discusses Broadcast Encryption (BE) as a suitable basis for DRM functionalities in conformance to the ICN communication paradigm. Especially when network inherent caching is considered the advantage of BE will be highlighted.

In ICN, data packets can be cached inherently in the network and any network participant can request a copy of these packets. This makes it very difficult to implement an access control for content that is distributed via ICN. A naive approach is to encrypt the transmitted data for each consumer with a distinct key. This prohibits everyone other than the intended consumers to decrypt and consume the data. However, this approach is not suitable for ICN's communication paradigm since it would reduce the benefits gained from the inherent network caching. Even if multiple consumers request the same content the requested data for each consumer would differ using this approach. A better but still insufficient idea is to use a single key for all consumers. This does not destruct the benefits of ICN's caching ability. The drawback is that if one of the consumers illegally distributes the key, the system is broken and any entity in the network can access the data. Changing the key after such an event is useless since the provider has no possibility to identify the illegal distributer. Therefore this person cannot be stopped from distributing the new key again. In addition to this issue other challenges have to be considered. Subscriptions expire after a certain time and then it has to be ensured that these consumers cannot access the content anymore. For a provider that serves millions of daily consumers (e.g. Netflix) there could be a significant number of expiring subscriptions per day. Publishing a new key every time a subscription expires would require an unsuitable amount of computational power just to re-encrypt the collection of audio-visual content.

A possible approach to solve these challenges is Broadcast Encryption (BE) [22] as proposed in [23]. From this point on, this section will focus only on BE as an enabler for DRM functionality in the use case of ICN video streaming. This subsection continues with the explanation of how BE works and shows how BE can be used to implement an access control scheme in the context of content distribution in ICN.

BE actually carries a misleading name. One might expect a concrete encryption scheme. However, it belongs to the family of key-management schemes (KMS). KMS are responsible for the generation, exchange, storage and replacement of cryptographic keys. The most interesting characteristics of Broadcast Encryption Schemes (BES) are:

- A BES typically uses a global trusted entity called the licensing agent (LA), which is responsible for spreading a set of pre-generated secrets among all participants. Each participant gets a distinct subset of secrets assigned from the LA.
- The participants can agree on a common session key, which is chosen by the LA. The LA broadcasts an encrypted message that includes the key. Participants with a valid set of secrets can derive the session-key from this message.

- The number of participants in the system can change dynamically. Entities may join or leave the communication group at any time. If a new entity joins the LA passes on a valid set of secrets to that entity. If an entity leaves (or is forced to leave) the LA revokes the entity's subset of keys, which means that it cannot derive the correct session key anymore when the LA distributes a new key.
- Traitors (entities that reveal their secrets) can be traced and excluded from ongoing communication. The algorithms and preconditions to identify a traitor vary between concrete BES.

This listing already illustrates why BE is suitable to control the access to data that is distributed via an information-centric network. BE enables the usage of a single session key for confidential data transmission between a dynamically changing subset or network participants. ICN caches can be utilized since the data is encrypted only with a single key known by all legitimate clients. Furthermore, traitors can be identified and removed from the system. The issue of re-encryption still exists, because the LA will eventually update the session key when a participant should be excluded. However, this disadvantage can be relaxed in some way if the following points are considered:

- The updates of the session key can be delayed until a set of compromised secrets has been gathered. Note that secrets may become compromised because of two reasons. First, a traitor could have illegally revealed the secret. Second, the subscription of an entity expired. Delayed revocation temporarily enables some non-legitimate entities to consume content. However, this should not be a severe problem in home entertainment scenarios. Updating the session key in regular (not too short) intervals is a good tradeoff. The longer the interval last the less computational resources are required for content re-encryption and the better the cache utilization in the ICN will be. To evict old data from ICN caches that has been encrypted with the prior session key the publisher could indicate a lifetime for transmitted packets.
- Content should be re-encrypted dynamically at request time. This has the benefit that untapped content is not re-encrypted if the content is not requested during two session key updates and therefore no resources are wasted. Furthermore, if the updates are triggered in non-peak times the maximum amount of resource needed at one point in time can be lowered effectively, since in peak times generally more diverse content is requested.
- Since the amount of required computational resources may vary strongly from time to time it would be beneficial for any

streaming provider to use cloud-based services to be able to dynamically adapt the required resources to the current needs. Regarding to a lack of computation time or bandwidth the cloud service could be used to scale up to overcome shortages.

Figure 4 show the potential usage of BE in a multimedia delivery frameworks that builds upon ICN infrastructure and uses the concept of dynamic adaptive streaming, e.g., DASH. BE would be implemented on the top to have an efficient and scalable way of access control to the multimedia content.

```
+--------Multimedia Delivery Framework--------+
|                                             |
|     Technologies            Properties      |
|  +-----------------+    +-----------------+ |
|  |   Broadcast     |<--->|  Controlled    | |
|  |   Encryption    |     |    Access      | |
|  +-----------------+    +-----------------+ |
|  |Dynamic Adaptive|<--->|  Multimedia     | |
|  |   Streaming     |     |  Adaptation    | |
|  +-----------------+    +-----------------+ |
|  |      ICN        |<--->|   Cachable     | |
|  | Infrastructure  |     |  Data Chunks   | |
|  +-----------------+    +-----------------+ |
+---------------------------------------------+
```

Figure 4: A potential multimedia framework using BE.

8.2 . AAA Based DRM for ICN Networks

8.2.1. Overview

Recently, a novel approach to Digital Rights Management (DRM) has emerged to link DRM to usual network management operations, hence linking DRM to authentication, authorization, and accounting (AAA) services. ICN provides the abstraction of an architecture where content is requested by name and could be served from anywhere. In DRM, the content provider (the origin of the content) allows the destination (the end user account) to use the content. The content provider and content storage/cache are at two different entities in ICC and for traditional DRM only source and destination count and not the intermediate storage. The proposed solution allows the provider of the caching to be involved in the DRM policies using well known AAA mechanisms. It is important to note that this solution is compatible with the proposes the Broadcast Encryption

(BE) proposed earlier in this draft. The BE proposes a technology as this solution is more operational.

8.2.2. Implementation

With the proposed AAA-based DRM, when content is requested by name from a specific destination, the request could link back to both the content provider and the caching provider via traditional AAA mechanisms, and trigger the appropriate DRM policy independently from where the content is stored. In this approach the caching, DRM and AAA remain independent entities but can work together through ICN mechanisms. The proposed solution enables extending the traditional DRM done by the content provider to jointly being done by content provider and network/caching provider.

The solution is based on the concept of a "token". The content provider authenticates the end user and issues an encrypted token to authenticate the a named content ID or IDs that the user can access. The token will be shared with the network provider and used as the interface to the AAA protocols. At this point all content access is under the control of the network provider and the ICN. The controllers and switches can manage the content requests and handle mobility. The content can be accessed from anywhere as long as the token remains valid or the content is available in the network. In such a scheme the content provider does not need to be contacted every time a named content is requested. This reduces the load of the content provider network and creates a DRM mechanism that is much more appropriate for the distributed caching and peer-to-peer storage characteristic of ICN networks. In particular, the content requested by name can be served from anywhere under the only condition that the storage/cache can verify that the token is valid for content access.

The solution is also fully customizable to both content and network provider’s needs as the tokens can be issued based on user accounts, location and hardware (MAC address for example) linking it naturally to legacy authentication mechanisms. In addition, since both content and network providers are involved in DRM policies pollution attacks and other illegal requests for the content can be more easily detected. The proposed AAA-based DRM is currently under full development.

9. Future Steps for Video in ICN

The explosion of online video services, along with their increased consumption by mobile wireless terminals, further exacerbates the challenges of Video Adaptation leveraging ICN mechanisms. The following sections present a series of research items derived from these challenges, further introducing next steps for the subject.

9.1. Large Scale Live Events

An active area of investigation and a potential use case where ICN would provide significant benefits, is that of distributing content, and video in particular, using local communications in large scale events such as sports event in a stadium, a concert or a large demonstration.

Such use-case involves locating content that is generated on the fly and requires discovery mechanisms in addition to sharing mechanisms. The scalability of the distribution becomes important as well.

9.2. Video Conferencing and Real-Time Communications

Current protocols for video-conferencing have been designed, and this document needs to take input from them to identify the key research issues. Real-time communications add timing constraints (both in terms of delay and in terms of synchronization) to the scenario discussed above.

AR and VR (and immersive multimedia experiences in general) are clearly an area of further investigation, as they involve combining multiple streams of data from multiple users into a coherent whole. This raises issues of multi-source multi-destination multimedia streams that ICN may be equipped to deal with in a more natural manner than IP that is inherently unicast.

9.3. Store-and-Forward Optimized Rate Adaptation

One of the benefits of ICN is to allow the network to insert caching in the middle of the data transfer. This can be used to reduce the overall bandwidth demands over the network by caching content for future re-use. But it provides more opportunities for optimizing video streams.

Consider for instance the following scenario: a client is connected via an ICN network to a server. Let's say the client is connected wirelessly to a node that has a caching capability, which is connected through a WAN to the server. Assume further that the capacity of each of the links (both the wireless and the WAN logical links) vary with time.

If the rate adaptation is provided in an end-to-end manner, as in current mechanisms like DASH, then the maximal rate that can be supported at the client is that of the minimal bandwidth on each link.

For instance, if during time period 1, the wireless capacity is 1 and the wired capacity is 2, and during time period 2, the wireless is 2 due to some hotspot, and the wired is 1 due to some congestion in the network, then the best end-to-end rate that can be achieved is 1 during each period.

However, if the cache is used during time period 1 to pre-fetch 2 units of data, then during period 2, there is 1 unit of data at the cache, and another unit of data, which can be streamed from the server, and the rate that can be achieved is therefore 2 units of data. In this case, the average bandwidth rises from 1 to 1.5 over the 2 periods.

This straw man example illustrate a) the benefit of ICN for increasing the throughput of the network, and b) the need for the special rate adaptation mechanisms to be designed so as to take advantage of this gain. End-to-end rate adaptation cannot take advantage of the cache availability.

9.4. Heterogeneous Wireless Environment Dynamics

With the ever-growing increase in online services being accessed by mobile devices, operators have been deploying different overlapping wireless access networking technologies. In this way, in the same area, user terminals are within range of different cellular, Wi-Fi or even WiMAX networks. Moreover, with the advent of the Internet of Things (e.g., surveillance cameras feeding video footage), this list can be further complemented with more specific short-range technologies, such as Bluetooth or ZigBee.

In order to leverage from this plethora of connectivity opportunities, user terminals are coming equipped with different wireless access interfaces, providing them with extended connectivity opportunities. In this way, such devices become able to select the type of access which best suits them according to different criteria, such as available bandwidth, battery consumption, access do different link conditions according to the user profile or even access to different content. Ultimately, these aspects contribute to the Quality of Experience perceived by the end-user, which is of utmost importance when it comes to video content.

However, the fact that these users are mobile and using wireless technologies, also provides a very dynamic setting, where the current optimal link conditions at a specific moment might not last or be maintained while the user moves. These aspects have been amply analyzed in recently finished projects such as FP7 MEDIEVAL [18], where link events reporting on wireless conditions and available alternative connection points were combined with vide requirements and traffic optimization mechanisms, towards the production of a joint network and mobile terminal mobility management decision. Concretely, in [19] link information about the deterioration of the wireless signal was sent towards a mobility management controller in the network. This input was combined with information about the user profile, as well as of the current video service requirements, and used to trigger the decrease or increase of scalable video layers, adjusting the video to the ongoing link conditions. Incrementally, the video could also be adjusted when a new better connectivity opportunity presents itself.

In this way, regarding Video Adaptation, ICN mechanisms can leverage from their intrinsic multiple source support capability and go beyond the monitoring of the status of the current link, thus exploiting the availability of different connectivity possibilities (e.g., different "interfaces"). Moreover, information obtained from the mobile terminal’s point of view of its network link, as well as information from the network itself (i.e., load, policies, and others), can generate scenarios where such information is combined in a joint optimization procedure allowing the content to be forward to users using the best available connectivity option (e.g., exploiting management capabilities supported by ICN intrinsic mechanisms as in [20]).

In fact, ICN base mechanisms can further be exploited in enabling new deployment scenarios such as preparing the network for mass requests from users attending a large multimedia event (i.e., concert, sports), allowing video to be adapted according to content, user and network requirements and operation capabilities in a dynamic way.

The enablement of such scenarios require further research, with the main points highlighted as follows:

. Development of a generic video services (and obviously content) interface allowing the definition and mapping of their requirements (and characteristics) into the current capabilities of the network;

- How to define a scalable mechanism allowing either the video application at the terminal, or some kind of network management entity, to adapt the video content in a dynamic way;

- How to develop the previous research items using intrinsic ICN mechanisms (i.e., naming and strategy layers);

- Leverage intelligent pre-caching of content to prevent stalls and poor quality phases, which lead to bad Quality of Experience of the user. This includes in particular the usage in mobile environments, which are characterized by severe bandwidth changes as well as connection outages, as shown in [21];

- How to take advantage of the multi-path opportunities over the heterogeneous wireless interfaces.

9.5. Network Coding for Video Distribution in ICN

An interesting research area for combining heterogeneous sources is to use network coding [24]. Network coding allows to asynchronously combine multiple sources by having each of them send information that is not duplicated by the other but can be combined to retrieve the video stream.

However, this creates issues in ICN in terms of defining the proper rate adaptation for the video stream; securing the encoded data; caching the encoded data; timeliness of the encoded data; overhead of the network coding operations both in network resources and in added buffering delay, etc.

Network coding has shown promise in reducing buffering events in unicast, multicast and P2P setting. [26] considers strategies using network coding to enhance QoE for multimedia communications. Network coding can be applied to multiple streams, but also within a single stream as an equivalent of a composable erasure code. Clearly, there is a need for further investigation of network coding in ICN, potentially as a topic of activity in the research group.

9.6. Synchronization Issues for Video Distribution in ICN

ICN de-couples the fetching of video chunks from the location of these chunks. This means an audio chunk may be received from one network element (cache/storage/server) while a video chunk may be received from another one while another chunk (say, the next one, or another layer from the same video stream) may come from a third element. This introduces disparity in the retrieval times and locations of the different elements of a video stream that need to be played at the same (or almost same) time. Synchronization of such delivery and playback may require specific synchronization tools for video delivery in ICN.

Other synchronization aspects involve:

- synchronizing within a single stream, for instance the consecutive chunks of a single stream, or the multiple layers of a layered scheme, when sources and transport layers may be different. Re-ordering the packets of a stream distributed over multiple sources at the video client, or ensuring that multiple chunks coming from multiple sources arrive within an acceptable time window;
- synchronizing multiple streams, such as the audio and video components of a video stream, which can be received from independent sources;
- synchronizing multiple streams from multiple sources to multiple destinations, such as mass distribution of live events. For instance, for live video streams or video-conferencing, some level of synchronization is required so that people watching the stream view the same events at the same time.

Some of these issues were addressed in [27] in the context of social video consumption. Network coding, with traffic engineering, is considered as a potential solution for synchronization issues. Other approaches could be considered that are specific for ICN as well.

Traffic engineering in ICN [28,29] may be required to provide proper synchronization of multiple streams.

10. Security Considerations

This is informational. There are no specific security considerations outside of those mentioned in the text.

11. IANA Considerations

This document does not require any IANA action.

12. Conclusions

This draft proposed adaptive video streaming for ICN, identified potential problems and presented the combination of CCN with DASH as a solution. As both concepts, DASH and CCN, maintain several elements in common, like, e.g., the content in different versions being dealt with in segments, combination of both technologies seems useful. Thus, adaptive streaming over CCN can leverage advantages such as, e.g., efficient caching and intrinsic multicast support of CCN, routing based on named data URIs, intrinsic multi-link and multi-source support, etc.

In this context, the usage of CCN with DASH in mobile environments comes together with advantages compared to today's solutions, especially for devices equipped with multiple network interfaces. The retrieval of data over multiple links in parallel is a useful feature, specifically for adaptive multimedia streaming, since it offers the possibility to dynamically switch between the available links depending on their bandwidth capabilities, transparent to the actual DASH client.

13. References

13.1. Normative References

[RFC6972] Y. Zhang, N. Zong, "Problem Statement and Requirements of the Peer-to-Peer Streaming Protocol (PPSP)", RFC6972, July 2013

13.2. Informative References

[1] ISO/IEC DIS 23009-1.2, Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats

[2] Lederer, S., Mueller, C., Rainer, B., Timmerer, C., Hellwagner, H., "An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks", in Proceedings of the IEEE International Conference on Multimedia and Expo 2013, San Jose, USA, July, 2013

[3] Liu, Y., Geurts, J., Point, J., Lederer, S., Rainer, B., Mueller, C., Timmerer, C., Hellwagner, H., "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis", in Proceedings of the IEEE international Conference on Communication (ICC) 2013 - Next-Generation Networking Symposium, Budapest, Hungary, June, 2013

[4] Grandl, R., Su, K., Westphal, C., "On the Interaction of Adaptive Video Streaming with Content-Centric Networks", eprint arXiv:1307.0794, July 2013.

[5]   V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking named content", in Proc. of the 5th int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT ’09). ACM, New York, NY, USA, 2009, pp. 1-12.

[6]   A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano and A. Bragagnini, "Offloading cellular networks with Information-Centric Networking: The case of video streaming", In Proc. of the Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM ’12), IEEE, San Francisco, CA, USA, 1-3, 2012.

[7]   V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice over content-centric networks," in ACM ReArch Workshop, 2009

[8]   Christopher Mueller, Stefan Lederer and Christian Timmerer, A proxy effect analysis and fair adaptation algorithm for multiple competing dynamic adaptive streaming over HTTP clients, In Proceedings of the Conference on Visual Communications and Image Processing (VCIP) 2012, San Diego, USA, November 27-30, 2012.

[9]   DASH Research at the Institute of Information Technology, Multimedia Communication Group, Alpen-Adria Universitaet Klagenfurt, URL: http://dash.itec.aau.at

[10] A. Detti, N. Blefari-Melazzi, S. Salsano, and M. Pomposini CONET: A content centric inter-networking architecture," in ACM Workshop on Information-Centric Networking (ICN), 2011.

[11] W. K. Chai, N. Wang, I. Psaras, G. Pavlou, C. Wang, G. C. de Blas, F. Ramon-Salguero, L. Liang, S. Spirou, A. Beben, and E. Hadjioannou, "CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services," IEEE Communications Magazine, vol. 49, no. 3, pp. 112-120, March 2011

[12] NetInf project Website http://www.netinf.org

[13] N. Magharei, R. Rejaie, Yang Guo, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , vol., no., pp.1424,1432, 6-12 May 2007

[14] PPSP WG Website https://datatracker.ietf.org/wg/ppsp/

[15] A. Bakker, R. Petrocco, V. Grishchenko, "Peer-to-Peer Streaming Peer Protocol (PPSPP)", draft-ietf-ppsp-peer-protocol-08

[16] Rui S. Cruz, Mario S. Nunes, Yingjie Gu, Jinwei Xia, Joao P. Taveira, Deng Lingli, "PPSP Tracker Protocol-Base Protocol (PPSP-TP/1.0)", draft-ietf-ppsp-base-tracker-protocol-02

[17] A.Detti, B. Ricci, N. Blefari-Melazzi,"Peer-To-Peer Live Adaptive Video Streaming for Information Centric Cellular Networks", IEEE PIMRC 2013,London, UK, 8-11 September 2013

[18] http://www.ict-medieval.eu

[19] B. Fu, G. Kunzmann, M. Wetterwald, D. Corujo, R. Costa, "QoE-aware Traffic Management for Mobile Video Delivery", Proc. 2013 IEEE ICC, Workshop on Immersive & Interactive Multimedia Communications over the Future Internet (IIMC), Budapest, Hungary, Jun 2013.

[20] Corujo D., Vidal I., Garcia-Reinoso J., Aguiar R., "A Named Data Networking Flexible Framework for Management Communications", IEEE Communications Magazine, Vol. 50, no. 12, pp. 36-43, Dec 2012

[21] Crabtree B., Stevens T., Allan B., Lederer S., Posch D., Mueller C., Timmerer C., Video Adaptation in Limited or Zero Network Coverage, CCNxConn 2013,PARC, Palo Alto, pp. 1-2, 2013

[22] Fiat A., Naor M., "Broadcast Encryption", in Advances in Cryptology (Crypto'93), volume 773 of Lecture Notes in Computer Science, pages 480-491. Springer Berlin / Heidelberg, 1994.

[23] Posch D., Hellwagner H., Schartner P., "On-Demand Video Streaming based on Dynamic Adaptive Encrypted Content Chunks", in Proceedings of the 8th International Workshop on Secure Network Protocols (NPSec'13), Los Alamitos, IEEE Computer Society Press, October, 2013.

[24] Montpetit M.J., Westphal C., Trossen D., "Network Coding Meets Information Centric Networks," in Proceedings of the workshop on Name-Oriented Mobility (NOM), jointly with ACM MobiHoc 2013, Hilton Head, SC, June 2013.

[25] Le Callet P., Moeller S. and Perkis A. (eds.), Qualinet White Paper on Definitions of Quality of Experience (2012). European Network on Quality of Experience in Multimedia

Systems and Services (COST Action IC 1003), Lausanne, Switzerland, Version 1.2, March 2013.

[26] Medard M., Kim M., ParandehGheibi M., Zeng W., Montpetit M.J., Quality of Experience for Multimedia Communications: Network Coding Strategies, Technical Report, MIT, 2012/3

[27] Montpetit M.J., Holtzman H., Chakrabarti K., Matijasevic M., Social video consumption: Synchronized viewing experiences across devices and networks, IEEE International Conference on Communications Workshops (ICC), 2013, 286-290

[28] Su K., Westphal C., On the Benefit of Information Centric Networks for Traffic Engineering, IEEE ICC Conference, June 2014

[29] Chanda A., Westphal C., Raychaudhuri D., Content Based Traffic Engineering in Software Defined Information Centric Networks, in IEEE INFOCOM Workshop NOMEN'13, April, 2013

[30] Castro M., Druschel P., Kermarrec A.-M., Nandi A., Rowstron A., Singh A. SplitStream: Highbandwidth multicast in cooperative environments. In Proceedings of ACM SOSP (2003).

[31] ATIS IPTV Interoperability Forum, ATIS IFF, http://www.atis.org/iif/deliv.asp

14. Authors' Addresses

Stefan Lederer, Christian Timmerer, Daniel Posch
Alpen-Adria University Klagenfurt
Universitaetsstrasse 65-67, 9020 Klagenfurt, Austria

Email: {firstname.lastname}@itec.aau.at

Cedric Westphal, Aytac Azgin. Shucheng (Will) Liu
Huawei
2330 Central Expressway, Santa Clara, CA95050, USA

Email: {cedric.westphal,aytac.azgin,liushucheng}@huawei.com

Christopher Mueller
bitmovin GmbH
Lakeside B01, 9020 Klagenfurt, Austria

Email: christopher.mueller@bitmovin.net

Andrea Detti
Electronic Engineering Dept.
University of Rome Tor Vergata
Via del Politecnico 1, Rome, Italy

Email: andrea.detti@uniroma2.it

Daniel Corujo,
Advanced Telecommunications and Networks Group
Instituto de Telecomunicacoes
Campus Universitario de Santiago
P-3810-193 Aveiro, Portugal

Email: dcorujo@av.it.pt

Jianping Wang
City University of Hong Kong
Hong Kong, China

Email: jianwang@cityu.edu.hk

Marie-Jose Montpetit

Email: marie@mjmontpetit.com

Niall Murray
Dept. of Electronic, Computer and Software Engineering
Athlone Institute of Technology
Dublin Rd., Athlone, Ireland

Email: nmurray@research.ait.ie

15. Acknowledgements

This work was supported in part by the EC in the context of the SocialSensor (FP7-ICT-287975) project and partly performed in the Lakeside Labs research cluster at AAU. SocialSensor receives research funding from the European Community's Seventh Framework Programme. The work for this document was also partially performed in the context of the FP7/NICT EU-JAPAN GreenICN project, http://www.greenicn.org. Apart from this, the European Commission has no responsibility for the content of this draft. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The authors would like to thank Shucheng (Will) Liu from Huawei for supporting Dr. Wang's research and Marie-Jose Montpetit of MIT for her help in writing the AAA for DRM section.

ICNRG J. Seedorf
Internet-Draft NEC
Intended status: Informational June 25, 2014
Expires: December 27, 2014

Binding Self-certifying Names to Real-World Identities with a Web-of-Trust
draft-seedorf-icn-wot-selfcertifying-00

Abstract

Self-certifying names are one way of binding a given public key to a certain name in Information Centric Networking. However, an additional binding of a self-certifying name to a Real-World identity is needed in most cases, so that a recipient of some information cannot only verify that the publisher was in possession of the correct corressponding private key for the requested name, but that in addition the name itself is the intended one. This draft specifies how such a binding of Real-World identities with self-certifying ICN names can be done, taking existing IETF specifications into account.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. High-Level Design 3
3. Standardisation Considerations 4
4. Conclusion 4
5. References 5
5.1. Normative References 5
5.2. Informative References 5
Appendix A. Acknowledgment 6
Author's Address 6

1. Introduction

Self-certifying names provide the useful property that any entity in a distributed system can verify the binding between a corresponding public key and the self-certifying name without relying on a trusted third party [Aura2003]. Self-certifying names thus provide a decentralized form of data origin authentication. This feature makes self-certifying names a prime candidate for addressing the security requirements in Information Centric Networking (ICN) (which are inherently different from IP networks): a source can digitally sign data associated with a self-certifying name, and any intermediate entity (e.g. ICN-router/Cache) or receiving entity (i.e. issuer of a request for the name) can verify the signature, without the need to verify the identity of the host that caches the object, nor relying on a trusted third party, or a Public Key Infrastructure (PKI). However, as noted in [Ghodsi2011] and elsewhere, self-certifying names lack a binding with a corresponding real-world identity (RWI): the concept enables to verify that whoever signed some data was in possession of the private key associated with the self-certifying name, but it does not provide any means to verify what real-world identity corresponds to the public key, i.e. who actually signed the data [Ghodsi2011] [Nom2014].

In principle, this binding between a public key and an RWI could be provided by a PKI, or alternatively by a Web-of-Trust (WoT) [Ghodsi2011]. Several ICN approaches use a PKI [Survey] . However, until recently, there have not been concrete proposals for a WoT-based approach for binding a public key (or a self-certifying name) with an RWI in content-oriented architectures. A concrete approach on how this can be done has been proposed in [Nom2014]. This document has the objective of providing the corresponding necessary standards specification to enable this approach (or similar ones) in principle in an interoperable way.

2. High-Level Design

On a high level, binding of self-certifying names and a Web-of-Trust can be achieved in the following way (see [Nom2014] for a detailed example of such an approach): The WoT key-ID is equivalent to the self-certifying name part used in the naming scheme. This ties the self-certifying name with the ID of the corresponding public key in the WoT.

For instance, in the existing PGP Web-of-Trust, the V4 key ID is the lower 64 bits of the fingerprint of the public key, where the fingerprint is essentially the 160-bit SHA-1 hash of the public key [RFC2440]. So if a self-certifying name would be based on the same lower 64-bits of the fingerprint of a given public key, this public key would be tied to the self-certifying name and at the same time be tied to the real-world identity used in the WoT, e.g. an email-address or the real (i.e. non-self-certifying) name of a given ICN publisher.

Thus, if a user requests the content for a self-certifying name in a given ICN architecture, he/she would retrieve the content which contains a digital signature and the corresponding public key for the self-certifying name. The user can then verify that the content retrieved indeed belongs to the name by first hashing the public key and confirm that the hash (or part of it) matches the requested name, and second using the public key to verify the signature over the content. This is in principle the general way of using self-certifying names for data origin authentication in distributed systems. If, in addition, (part of) the self-certifying name is equivalent to a WoT key-ID, the user can use any WoT infrastructure (e.g. PGP keyservers) to retrieve certificates for the key ID that contain/confirm the binding between the corresponding (to the WoT key ID) public key with a real-world identity, such as an email address. This binding provides the requesting user with assurance that the self-certifying name indeed is owned by the intended publisher, i.e. is the correct, intended name from the requestor's perpective.

The current PGP specification [RFC2440] considers only a bitlength of 64-bit for forming the key-ID, which is not very collision-resistant (collision-resistance among different key-IDs was not a design goal for PGP [RFC2440]). For securely binding a self-certifying name to a WoT key-ID, collision-resistance is a design goal, because otherwise attckaers could potentially forge a binding of their public key with a given self-certifying name. Thus, either a longer bitlength of the hash of the public key (or its fingerprint) must be used, or hash extension techniques [Aura] must be used, which effectively make collision attacks harder for constant bitlengths at the price of the time needed to create a public/private key pair. Future versions of this document will take these design considerations into account.

3. Standardisation Considerations

Future versions of this document will outline a concrete protocol specification for binding self-certifying names to a Web-of-Trust as outlined on a high level in the previous Section. Below some initial standardisation considerations are highlighted. Also, future versions of this document will look in more detail into existing IETF specifications, e.g. regarding ICN naming ([RFC6920]) and Web-of-Trust ([RFC2440]), and inspect to what extend such existing specifications can be used directly or in a modified form.

An initial list of details that need to be specified is the following:

- (List of) Asymmetric cryptography algorithm(s) and corresponding bit-length(s)
- (List of) Hash algorithm(s) and corresponding bit-length(s)
- Rules that define what part of the hash is used for forming the self-certifying part of the name
- Rules for forming a self-certifying name based on a public key
- Semantics of a signature in the Web-of-Trust
- Defintion of the web-of-trust key-ID and how it relates to the self-certifying name
- Defintion of how many bits are used in case of hash extension techniques [Aura]

4. Conclusion

One option for binding self-certifying names to real-world identities is using a Web-of-Trust. This document aims at a concrete specification for providing such a binding, taking existing IETF specification into account. Future versions of this document will provide a more detailed specification.

5. References

5.1. Normative References

[RFC2440] Callas, J., Donnerhacke, L., Finney, H., and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, April 2013.

5.2. Informative References

[Aura] Aura, T. and M. Roe, "Strengthening Short Hash Values", http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.7681, .

[Aura2003] Aura, T., "Cryptographically Generated Addresses (CGA)", 6th International Conference on Information Security (ISC), 2003, .

[Ghodsi2011] Ghodsi, A., Koponen, T., Rajahalme, J., Sarolahti, P., and S. Shenker, "Naming in Content-oriented Architectures", ACM SIGCOMM Workshop on Information-centric Networking, 2011, .

[I-D.seedorf-icn-disaster] Arumaithurai, M., Seedorf, J., Tagami, A., Ramakrishnan, K., and N. Blefari-Melazzi, "Using ICN in disaster scenarios", draft-seedorf-icn-disaster-01 (work in progress), October 2013.

[Nom2014] Seedorf, J., Kutscher, D., and F. Schneider, "Decentralised Binding of Self-Certifying Names to Real-World Identities for Assessment of Third-Party Messages in Fragmented Mobile Networks", 2nd Workshop on Name Oriented Mobility (NOM), 2014, .

[Survey] Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., and G. Polyzos, "A Survey of Information-Centric Networking Research", IEEE Communications Surveys and Tutorials, Vol. 16, No. 2, pp 1024-1049, 2014, .

Appendix A. Acknowledgment

This document has been supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the GreenICN project, the European Commission, or NICT.

Author's Address

Jan Seedorf
NEC
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 221
Fax: +49 6221 4342 155
Email: seedorf@neclab.eu

ICN Research Group Y. Zhang
Internet-Draft D. Raychadhuri
Intended status: Informational WINLAB, Rutgers University
Expires: February 29, 2016 L. Grieco
Politecnico di Bari (DEI)
E. Baccelli
INRIA
J. Burke
UCLA REMAP
R. Ravindran (Ed)
G. Wang
Huawei Technologies
August 28, 2015

ICN based Architecture for IoT - Requirements and Challenges
draft-zhang-iot-icn-challenges-02

Abstract

The Internet of Things (IoT) promises to connect billions of objects to Internet. After deploying many stand-alone IoT systems in different domains, the current trend is to develop a common, "thin waist" of protocols forming a unified, defragmented IoT platform. Such a platform will make objects accessible to applications across organizations and domains. Towards this goal, quite a few proposals have been made to build a unified host centric IoT platform as an overlay on top of today's Internet. Such overlay solutions, however, are inadequate to address the important challenges posed by a heterogeneous, global scale deployment of IoT, especially in terms of mobility, scalability, and communication reliability, due to the inherent inefficiencies of the current Internet. To address this problem, we propose to build a common set of protocols and services, which form an IoT platform, based on the Information Centric Network (ICN) architecture, which we call ICN-IoT. ICN-IoT leverages the salient features of ICN, and thus provides seamless mobility support, scalability, and efficient content and service delivery.

This draft describes representative IoT requirements and ICN challenges to realize a unified ICN-IoT framework. Towards this, we first identify a list of important requirements which a unified IoT architecture should have to support tens of billions of objects. Though we see most of the IoT requirements can be met by ICN, we discuss specific challenges ICN has to address to satisfy them. Then we discuss important and popular IoT scenarios including the "smart" home, campus, grid, transportation infrastructure, healthcare, Education, and Entertainment.

Status of This Memo

Copyright Notice

Table of Contents

1. IoT Motivation 3
2. IoT Architectural Requirements 4
  2.1. Naming 4
  2.2. Scalability 4
  2.3. Resource Constraints 4
  2.4. Traffic Characteristics 5
  2.5. Contextual Communication 6
  2.6. Handling Mobility 6
  2.7. Storage and Caching 6
  2.8. Security and Privacy 7
  2.9. Communication Reliability 7
  2.10. Self-Organization 8
  2.11. Ad hoc and Infrastructure Mode 8
  2.12. Open API 8

3. State of the Art 8
3.1. Silo IoT Architecture 9
3.2. Overlay Based Unified IoT Solutions 9
3.2.1. Weaknesses of the Overlay-based Approach 10
4. ICN Challenges for IoT 11
4.1. Naming and Name Resolution 12
4.2. Caching/Storage 14
4.3. Routing and Forwarding 15
4.4. Contextual Communication 17
4.5. In-network Computing 18
4.6. Security and Privacy 19
4.7. Energy Efficiency 21
5. Popular Scenarios 21
5.1. Homes 21
5.2. Enterprise 22
5.3. Smart Grid 22
5.4. Transportation 23
5.5. Healthcare 24
5.6. Education 24
5.7. Entertainment, arts, and culture 25
6. Informative References 25
Authors' Addresses 30

1. IoT Motivation

During the past decade, many standalone Internet of Things (IoT) systems have been developed and deployed in different domains. The recent trend, however, is to evolve towards a globally unified IoT platform, in which billions of objects connect to the Internet, available for interactions among themselves, as well as interactions with many different applications across boundaries of administration and domains. Building a unified IoT platform, however, poses great challenges on the underlying network and systems. To name a few, it needs to support 50-100 Billion networked objects [1], many of which are mobile. The objects will have extremely heterogeneous means of connecting to the Internet, often with severe resource constraints. Interactions between the applications and objects are often real-time and dynamic, requiring strong security and privacy protections. In addition, IoT applications are inherently information centric (e.g., data consumers usually need data sensed from the environment without any reference to the sub-set of motes that will provide the asked information). Taking a general IoT perspective, we begin by presenting IoT architectural requirements, then summarize how state-of-art approaches address these requirements. We then discuss IoT challenges from an ICN perspective and requirements posed towards its design. Final discussion focusses on IoT scenarios and their unique challenges.

2. IoT Architectural Requirements

A unified IoT platform has to support interactions among a large number of mobile devices across the boundaries of organizations and domains. As a result, it naturally poses stringent requirements in every aspect of the system design. Below, we outline a few important requirements that a unified IoT platform has to address.

2.1. Naming

The first step towards realizing a unified IoT platform is the ability to assign names that are unique within the scope and lifetime of each device, data items generated by these devices, or a group of devices towards a common objective. Naming has the following requirements: first, names need to be persistent (within one or more contexts) against dynamic features that are common in IoT systems, such as lifetime, mobility or migration; second, names need to be secure based on application requirements; third, names should provide advantages to application authors in comparison with traditional host address based schemes.

2.2. Scalability

Cisco predicts there will be around 50 Billion IoT devices such as sensors, RFID tags, and actuators, on the Internet by 2020 [1]. As mentioned above, a unified IoT platform needs to name every entity such as data, device, service etc. Scalability has to be addressed at multiple levels of the IoT architecture spanning naming, security, name resolution, routing and forwarding level. In addition, mobility adds further challenge in terms of scalability. Particularly with respect to name resolution the system should be able to register/update/resolve up a name within a short latency. To satisfy this requirement, decentralization of the name resolution can be the key.

2.3. Resource Constraints

IoT devices can be broadly classified into two groups: resource-sufficient and resource-constrained. In general, there are the following types of resources: power, computing, storage, bandwidth, and user interface.

Power constraints of IoT devices limit how much data these devices can communicate, as it has been shown that communications consume more power than other activities for embedded devices. Flexible techniques to collect the relevant information are required, and uploading every single produced data to a central server is undesirable. Computing constraints limit the type and amount of processing these devices can perform. As a result, more complex processing needs to be conducted at opportunistic points, example at the network edge, hence it is important to balance local computation versus communication cost.

Storage constraints of the IoT devices limit the amount of data that can be stored on the devices. This constraint means that unused sensor data may need to be discarded or stored in aggregated compact form time to time. Bandwidth constraints of the IoT devices limit the amount of communication. Such devices will have the same implication on the system architecture as with the power constraints; namely, we cannot afford to collect single sensor data generated by the device and/or use complex signaling protocols.

User interface constraints refer to whether the device is itself capable of directly interacting with a user should the need arise (e.g., via a display and keypad or LED indicators) or requires the network connectivity, either global or local, to interact with humans.

2.4. Traffic Characteristics

IoT traffic can be broadly classified into local area traffic and wide area traffic. Local area traffic is between nearby devices. For example, neighboring cars may work together to detect potential hazards on the highway, sensors deployed in the same room may collaborate to determine how to adjust the heating level in the room. These local area communications often involve data aggregation and filtering, have real time constraints, and require fast device/data/ service discovery and association. At the same time, the IoT platform has to also support wide area communications. For example, in Intelligent Transportation Systems, re-routing operations may require a broad knowledge of the status of the system, traffic load, availability of freights, whether forecasts and so on. Wide area communications require efficient data/service discovery and resolution services.

While traffic characteristics for different IoT systems are expected to be different, certain IoT systems have been analyzed and shown to have comparable uplink and downlink traffic volume in some applications such as [2], which means that we have to optimize the bandwidth/energy consumption in both directions. Further, IoT traffic demonstrates certain periodicity and burstiness [2]. As a result, when provisioning the system, the shape of the traffic volume has to be properly accounted for.

2.5. Contextual Communication

Many IoT applications shall rely on contextual information such as social, relationships of owners, administrative groupings, location, type of ecosystem (home, grid, transport etc.) of devices and data (which are referred to as contexts in this document) to initiate dynamic relationship and communication. For example, cars traveling on the highway may form a "cluster" based upon their temporal physical proximity as well as the detection of the same event. These temporary groups are referred to as contexts. IoT applications need to support interactions among the members of a context, as well as interactions across contexts.

Temporal context can be broadly categorized into two classes, long-term contexts such as those that are based upon social contacts as well as stationary physical locations (e.g., sensors in a car/ building), and short-term contexts such as those that are based upon temporary proximity (e.g., all taxicabs within half a mile of the Time Square at noon on Oct 1, 2013). Between these two classes, short-term contexts are more challenging to support, requiring fast formation, update, lookup and association.

2.6. Handling Mobility

There are several degrees of mobility in a unified IoT platform, ranging from static as in fixed assets to highly dynamic in vehicle-to-vehicle environments.

Mobility in the IoT platform can mean 1) the data producer mobility (i.e., location change), 2) the data consumer mobility, 3) IoT Network mobility (e.g., a body-area network in motion as a person is walking); and 4) disconnection between the data source and destination pair (e.g., due to unreliable wireless links). The requirement on mobility support is to be able to deliver IoT data below an application's acceptable delay constraint in all of the above cases, and and if necessary to negotiate different connectivity or security constraints specific to each mobile context.

2.7. Storage and Caching

Storage and caching plays a very significant role depending on the type of IoT ecosystem, also a function subjected to privacy and security guidelines. In a unified IoT platform, depending on application requirements, content caching may or may not be policy driven. If caching is pervasive, intermediate nodes don't need to always forward a content request to its original creator; rather, locating and receiving a cached copy is sufficient for IoT applications. This optimization can greatly reduce the content access latencies.

Furthermore considering hierarchical nature of IoT systems, ICN architectures enable a more flexible, heterogeneous and potentially fault-tolerant approach to storage providing persistence at multiple levels.

In network storage and caching, however, has the following requirements on the IoT platform. The platform needs to support the efficient resolution of cached copies. Further the platform should strive for the balance between caching, content security/privacy, and regulations.

2.8. Security and Privacy

In addition to the fundamental challenge of trust management, a variety of security and privacy concerns also exist in ICNs.

The unified IoT platform makes physical objects accessible to applications across organizations and domains. Further, it often integrates with critical infrastructure and industrial systems with life safety implications, bringing with it significant security challenges and regulatory requirements [11].

Security and privacy thus become a serious concern, as does the flexibility and usability of the design approaches. Beyond the overarching trust management challenge, security includes data integrity, authentication, and access control at different layers of the IoT platform. Privacy means that both the content and the context around IoT data need to be protected. These requirements will be driven by various stake holders such as industry, government, consumers etc.

2.9. Communication Reliability

IoT applications can be broadly categorized into mission critical and non-mission critical. For mission critical applications, reliable communication is one of the most important features as these applications have strong QoS requirements. Reliable communication requires the following capabilities for the underlying system: (1) seamless mobility support in the face of extreme disruptions (DTN), (2) efficient routing in the presence of intermittent disconnection, (3) QoS aware routing, (4) support for redundancy at all levels of a system (device, service, network, storage etc.).

2.10. Self-Organization

The unified IoT platform should be able to self-organize to meet various application requirements, especially the capability to quickly discover heterogeneous and relevant (local or global) devices/data/services based on the context. This discovery can be achieved through an efficient platform-wide publish-subscribe service, or through private community grouping/clustering based upon trust and other security requirements. In the former case, the publish-subscribe service must be efficiently implemented, able to support seamless mobility, in- network caching, name-based routing, etc. In the latter case, the IoT platform needs to discover the private community groups/clusters efficiently.

2.11. Ad hoc and Infrastructure Mode

Depending upon whether there is communication infrastructure, an IoT system can operate either in ad-hoc or infrastructure mode.

For example, a vehicle may determine to report its location and status information to a server periodically through cellular connection, or, a group of vehicles may form an ad-hoc network that collectively detect road conditions around them. In the cases where infrastructure is unavailable, one of the participating nodes may choose to become the temporary gateway.

The unified IoT platform needs to design a common protocol that serves both modes. Such a protocol should be able to provide: (1) energy-efficient topology discovery and data forwarding in the ad-hoc mode, and (2) scalable name resolution in the infrastructure mode.

2.12. Open API

General IoT applications involve sensing, processing, and secure content distribution occurring at various timescales and at multiple levels of hierarchy depending on the application requirements. This requires open APIs to be generic enough to support commonly used interactions between consumers, content producer, and IoT services, as opposed to proprietary APIs that are common in today’s systems. Examples include pull, push, and publish/subscribe mechanisms using common naming, payload, encryption and signature schemes.

3. State of the Art

Over the years, many stand-alone IoT systems have been deployed in various domains. These systems usually adopt a vertical silo architecture and support a small set of pre-designated applications. A recent trend, however, is to move away from this approach, towards a unified IoT platform in which the existing silo IoT systems, as well as new systems that are rapidly deployed. This will make their data and services accessible to general Internet applications (as in ETSI- M2M and oneM2M standards). In such a unified platform, resources can be accessed over Internet and shared across the physical boundaries of the enterprise. However, current approaches to achieve this objective are based upon Internet overlays, whose inherent inefficiencies due to IP protocol [8] hinders the platform from satisfying the IoT requirements outlined earlier (particularly in terms of scalability, security, mobility, and self-organization)

3.1. Silo IoT Architecture

```
                          [IoT Server]
                               |
                               |
                       ______|_______
                        {              }
    ________          {              }
   {        }        {              }
   {IoT Dev}\           {   Internet   }---[IoT Application]
   {________}  [IoTGW]---{              }
                      {              }
                        {______________}
```

Figure 1:Silo architecture of standalone IoT systems

A typical standalone IoT system is illustrated in Figure 1, which includes devices, a gateway, a server and applications. Many IoT devices have limited power and computing resources, unable to directly run normal IP access network (Ethernet, WIFI, 3G/LTE etc.) protocols. Therefore they use the IoT gateway to the server. Through the IoT server, applications can subscribe to data collected by devices, or interact with devices.

There have been quite a few popular protocols for standalone IoT systems, such as DF-1, MelsecNet, Honeywell SDS, BACnet, etc. However, these protocols are operating at the device-level abstraction, instead of information driven, leading to a highly fragmented protocol space with limited interoperability.

3.2. Overlay Based Unified IoT Solutions

The current approach to a unified IoT platform is to make IoT gateways and servers adopt standard APIs. IoT devices connect to the Internet through the standard APIs and IoT applications subscribe and receive data through standard control and data APIs. Building on top of today's Internet as an overlay, this is the most practical approach towards a unified IoT platform. There are ongoing standardization efforts including ETSI[3], oneM2M[4],and CORE[5]. Network operators can use standard API to build common IOT gateways and servers for their customers. Figure 2 shows the architecture adopted in this approach.

```
                 Publishing----[IoT Server]----Subscribing--
                     |        /    |       \               |
                     |       /     |        \              |
                   |      /_______|________  \            |
  ____________       |    /{               }  publishing    |
 {           }       |    | {              }     |         |
 {Smart Homes}\      |    | {   Internet   }---------[IoT Application]
 {___________}  [IoTGW]---{              }\    |    _______________
                          | {              } \  |   {               }
                          | {______________}  [IoTGW]-{Smart Healthcare}
                          |        |                 {________________}
                 Publishing [IoTGW]
                          |     ____|_____
                          |   {          }
                           ---{Smart Grid}
                              {__________}
```

Figure 2: Implementing an open IoT platform through standarized APIs on the IoT gateways and the server

3.2.1. Weaknesses of the Overlay-based Approach

The above overlay-based approach can work with many different protocols, but the system is built upon today's IP network, which has inherent weaknesses towards supporting a unified IoT system. As a result, it cannot satisfy some of the requirements we outlined in Section 2:

- o Naming. In current overlays for IoT systems the naming scheme is host centric, i.e., the name of a given resource/service is linked to the one of device that can provide it. In turn, device names are coupled to IP addresses, which are not persistent in mobile scenarios. On the other side, in IoT systems the same service/ resource could be provided by many different devices thus requiring a different design rationale.

- Trust. Trust management schemes are still relatively weak, focusing on securing communication channels rather than managing the data that needs to be secured directly.

- Mobility. The overlay-based approach uses IP addresses as names at the network layer, which hinders the support for device/service mobility or flexible name resolution. Further the Layer 2/3 management, and application-layer addressing and forwarding required to deploy current IoT solutions limit the scalability and management of these systems.

- Resource constraints. The overlay-based approach requires every device to send data to an aggregator or to the IoT server. Resource constraints of the IoT devices, especially in power and bandwidth, could seriously limit the performance of this approach.

- Traffic Characteristics. In this approach, applications are written in a host-centric manner suitable for point-to-point communication. IoT requires multicast support that is challenging in overlay systems today.

- Contextual Communications. This overlay-based approach cannot react to dynamic contextual changes in a timely fashion. The main reason is that context lists are kept at the IoT server in this approach, and they cannot help efficiently route requests information at the network layer.

- Storage and Caching. The overlay-based approach supports application-centric storage and caching but not what ICN envisions at the network layer, or flexible storage enabled via name-based routing or name-based lookup.

- Self-Organization. The overlay-based approach is topology-based as it is bound to IP semantics, and thus does not sufficiently satisfy the self-organization requirement. In addition to topological self-organization, IoT also requires data- and service-level self-organization [49], which is not supported by the overlay approach.

- Ad-hoc and infrastructure mode. As mentioned above, the overlay-based approach lacks self-organization, and thus does not provide efficient support for the ad-hoc mode.

4. ICN Challenges for IoT

ICN integrates content/service/host abstraction, name-based routing, compute, caching/storage as part of the network infrastructure connecting consumers and services which meets most of the requirements discussed above; however IoT requires special considerations given heterogeneity of devices and interfaces such as for constrained networking [31], data processing, and content distribution models to meet specific application requirements which we identify as challenges in this section. We also discuss scenario specific challenges discussed in Section 5.

4.1. Naming and Name Resolution

Inter-connecting numerous IoT entities, as well as establishing reachability to them, requires a scalable name resolution system considering several dynamic factors like mobility of end points, service replication, in-network caching, failure or migration [30] [33] [34] [47]. The objective is to achieve scalable name resolution handling static and dynamic ICN entities with low complexity and control overhead. In particular, the main requirements/challenges of a name space (and the corresponding Name Resolution System where necessary) are [26] [27]:

- o Scalability: The first challenge faced by ICN-IoT name resolution system is its scalability. Firstly, the approach has to support billions of objects and devices that are connected to the Internet, many of which are crossing administrative domain boundaries. Second of all, in addition to objects/devices, the name resolution system is also responsible for mapping IoT services to their network addresses. Many of these services are based upon contexts, hence dynamically changing, as pointed out in [30]. As a result, the name resolution should be able to scale gracefully to cover a large number of names/services with wide variations (e.g., hierarchical names, flat names, names with limited scope, etc.). Notice that, if hierarchical names are used, scalability can be also supported by leveraging the inherent aggregation capabilities of the hierarchy. Advanced techniques such as hyperbolic routing [43] may offer further scalability and efficiency.

- o Trust: We need to ensure the name of a network element is issued by a trustworthy issuer in the context of the application, such as a trusted organization in [44]. Further the validity of each piece of data published by an authorized entity in the namespace should be verifiable - e.g., by following a hierarchical chain-of-trust to a root that is acceptable for the application. See [44] for an example.

- o Deployability and interoperability: Graceful deployability and interoperability with existing platforms is a must to ensure a naming schema to gain success on the market [7]. As a matter of fact, besides the need to ensure coexistence between IP-centric and ICN-IoT systems, it is required to make different ICN-IoT realms, each one based on a different ICN architecture, to interoperate.

o Flexibility: Further challenges arise for hierarchical naming schema: referring to requirements on "constructable names" and "on-demand publishing" [23][24]. The former entails that each user is able to construct the name of a desired data item through specific algorithms and that it is possible to retrieve information also using partially specified names. The latter refers the possibility to request a content that has not yet been published in the past, thus triggering its creation.

o Latency: For real-time or delay sensitive M2M application, the name resolution should not affect the overall QoS. With reference to this issue it becomes important to circumvent too centralized resolution schema (whatever the naming style, i.e, hierarchical or flat) by enforcing in-network cooperation among the different entities of the ICN-IoT system, when possible [48]. In addition, fast name lookup are necessary to ensure soft/hard real time services [50][51][52]. This challenge is especially important for applications with stringent latency requirements, such as health monitoring, emergency handling and smart transportation [53].

o Locality and network efficiency: During name resolution the named entities closer to the consumer should be easily accessible (subject to the application requirements). This requirement is true in general because, whatever the network, if the edges are able to satisfy the requests of their consumers, the load of the core and content seek time decrease, and the overall system scalability is improved. This facet gains further relevance in those domains where an actuation on the environment has to be executed, based on the feedbacks of the ICN-IoT system, such as in robotics applications, smart grids, and industrial plants [49].

o Agility: Some data items could disappear while some other ones are created so that the name resolution system should be able to effectively take care of these dynamic conditions. In particular, this challenge applies to very dynamic scenarios (e.g., VANETs) in which data items can be tightly coupled to nodes that can appear and disappear very frequently.

o Control/scoping: Some information could be accessible only within a given scope. This challenge is very relevant for smart home and health monitoring applications, where privacy issues play a key role and the local scope of a home or healthcare environment may be well-defined. However, perimeter- and channel-based access control is often violated in current networks to enable over-thewire updates and cloud-based services, so scoping is unlikely to replace a need for data-centric security in ICN.

- o Confidentiality: As names can reveal information about the nature of the communication, mechanisms for name confidentiality should be available in the ICN-IoT architecture.

In addition to the above general requirements, we identify the following specific requirements for different IoT applications:

- o Smart homes require names that can enable local and wide area interactions; Also, security, privacy, and access control is particularly important for smart homes.

- o Smart grids require names and name resolution system that can enable networked control loops, real-time control, and security.

- o Smart transportation systems require names and name resolution system to be able to handle extreme mobility, short latency and security. In addition, the mobility patterns of transportation systems increase the likelyhood that a user migrates from one network realm to another one during the journey. In this case, names and NRS should be designed in such a way to enable interoperability between different heterogeneous ICN realms and/or ICN and IP realms [58].

- o Smart healthcare system requires names and name resolution system to enable real- time interactions, dependability, and security.

- o Smart campus systems usually consist of hetereogeneous IoT services, thus requring names and name resolution system to enable resource/ service ownership, and be application-centric.

4.2. Caching/Storage

In-network caching helps bring data closer to consumers, but its usage differs in constrained and infrastructure part of the IoT network. Caching in constrained networks is limited to small amounts in the order of 10KB, while caching in infrastructure part of the network can allow much larger chunks.

Caching in ICN-IoT faces several challenges:

- o The main challenge is to determine which nodes on the routing path should cache the data. According to [27], caching the data on a subset of nodes can achieve a better gain than caching on every en-route routers. In particular, the authors propose a "selective caching" scheme to locate those routers with better hit probabilities to cache data. According to [28], selecting a random router to cache data is as good as caching the content everywhere. In [45], the authors suggest that edge caching provides most of the benefits of in-network caching typically discussed in NDN, with simpler deployment. However, it and other papers consider workloads that are analogous to today's CDNs, not the IoT applications considered here. Further work is likely required to understand the appropriate caching approach for IoT applications.

- Another challenge in ICN-IoT caching is what to cache for IoT applications. In many IoT applications, customers often access a stream of sensor data, and as a result, caching a particular sensor data item may not be beneficial. In [29], the authors suggest to cache IoT services on intermediate routers, and in [30], the authors suggest to cache control information such as pub/sub lists on intermediate nodes. In addition, it is yet unclear what caching means in the context of actuation in an IoT system. For example, it could mean caching the result of a previous actuation request (using other ICN mechanisms to suppress repeated actuation requests within a given time period), or have little meaning at all if actuation uses authenticated requests as in [46].

Next we use specific IoT systems to explain the caching challenge:

- Smart homes may use in-network caching at gateway to enable efficient content access
- Smart grids may use in-network caching to back up valuable data
- Smart transportation may implement in-network caching on vehicles for efficient information dissemination
- Smart healthcare may use in-network caching for rapid information dissemination
- Smart campus systems may use in-network caching to enable social interactions and efficient content access.

4.3. Routing and Forwarding

Routing in ICN-IoT differs from routing in traditional IP networks in that ICN routing is based upon names instead of locators. Broadly speaking, ICN routing can be categorized into the following two categories: direct name-based routing and indirect routing using a name resolution service (NRS).

- In direct name-based routing, packets are forwarded by the name of the data [47][31][35] or the name of the destination node [36]. Here, the main challenge is to keep the ICN router state required to route/forward data low. This challenge becomes more serious when a flat naming scheme is used due to the lack of aggregation capabilities.

- In indirect routing, packets are forwarded based upon the locator of the destination node, and the locator is obtained through the name resolution service. In particular, the name-locator binding can be done either before routing (i.e., static binding) or during routing (i.e., dynamic binding). For static binding, the router state is the same as that in traditional routers, and the main challenge is the need to have fast name resolution, especially when the IoT nodes are mobile. For dynamic binding, ICN routers need to main a name-based routing table, hence the challenge of keeping the state information low. At the same time, the need of fast name resolution is also critical. Finally, another challenge is to quantify the cost associated with mobility management, especially static binding vs. dynamic binding.

During a network transaction, either the data producer or the consumer may move away and thus we need to handle the mobility to avoid information loss. ICN may differentiate mobility of a data consumer from that of a producer:

- When a consumer moves to a new location after sending out the request for Data, the Data may get lost, which requires the consumer to simply resend the request, a technique used by direct routing approach. Indirect routing approach doesn't differentiate between consumer and producer mobility [47], also network caching can improve data recovery for this approach.

- If the data producer itself has moved, the challenge is to control the control overhead while searching for a new data producer (or for the same data producer in its new position). To this end, flooding techniques could be used, but an intra-domain level only, otherwise the network stability would be seriously impaired. For handling mobility across different domains, more sophisticated approaches could be used, including the adoption of a SDN-based control plane.

Finally, in addition to the above requirements, specific IoT applications may impose specific challenges on routing and forwarding:

- In smart homes, we need local, intra-domain and inter-domain routing protocols.

o In smart grids, we often require very timely data delivery. Therefore, it is important to be able to locate the closest information. In addition, routing/forwarding robustness and resilience is also critical.

o In smart transportation, vehicle-to-vehicle ad-hoc communication is required for efficient information dissemination.

o In smart healthcare, timely and dependable routing and information forwarding is the key.

o In smart campus, inter-domain routing protocols are required which often need short latency.

4.4. Contextual Communication

Contextualization through metadata in ICN control or application payload allows IoT applications to adapt to different environments. This enables intelligent networks which are self-configurable and enable intelligent networking among consumers and producers [29]. For example, let us look at the following smart transportation scenario: "James walks on NYC streets and wants to find an empty cab closest to his location." In this example, the context is the relative locations of James and taxi drivers. A context service, as an IoT middleware, processes the contextual information and bridges the gap between raw sensor information and application requirements. Alternatively, naming conventions could be used to allow applications to request content in namespaces related to their local context without requiring a specific service, such as /local/geo/mgrs/4QFJ/123/678 to retrieve objects published in the 100m grid area 4QFJ 123 678 of the military grid reference system (MGRS). In both cases, trust providers may emerge that can vouch for an application's local knowledge.

However, extracting contextual information on a real-time basis is very challenging:

o We need to have a fast context resolution service through which the involved IoT devices can continuously update its contextual information to the application (e.g., each taxi's location and Jame's information in the above example). Or, in the namespace driven approach, mechanisms for continuous nearest neighbor queries in the namespace need to be developed.

o The difficulty of this challenge grows rapidly when the number of devices involved in a context as well as the number of contexts increases.

Next, in addition to the above requirements, specific IoT services may impose specific challenges on contextual communication:

- o In smart homes many control loops and actions are depend heavily on the context, and the contexts evolve with time, e.g., temperature, weather, number of occupants, etc

- o In smart grids, contextual information such as location, time, voltage fluctuations, depending on the specific segment of the grid, can be used to optimize several power distribution objectives.

- o In smart transportation, many different contexts exist, intertwined to each other and highly changing, which include location - both geographical and jurisdictional, time - absolute and relative to a schedule, traffic, speed, etc.

- o In smart healthcare several contexts can be used to delineate between levels of care and urgency, for example delineating between chronic, everyday, urgent, and emergency situations. Such contexts can evolve rapidly with significant impact to individuals health. Hence timely and accurate detection of contexts is critical.

- o In smart campus, due to the existence of many services, relevant contextual inputs can be used to improve the quality and efficiency of different services.

4.5. In-network Computing

In-network computing enables ICN routers to host heterogenous services catering to various network functions and applications needs. Contextual services for IoT networks require in-network computing, in which each sensor node or ICN router implements context reasoning [29]. Another major purpose of in-network computing is to filter and cleanse sensed data in IoT applications is critical as the data is noisy as is [37]. Named Function Networking [54] describes an extension of the ICN concept to named functions processed in the network, which could be used to generate data flow processing applications well-suited to, for example, time series data processing in IoT sensing applications.

- o In smart homes, local services can provide value-added contributions to a standardized home gateway network, through features such as reporting, context-based control, coordination with mobile devices, etc.

- In smart grids, we often rely on in-network computing to increase the scalability and efficiency of the system, putting computation closer to the data sources.

- In smart transportation, in-network computing is very useful to make vehicle become an active element of the system and to improve response time and scalability.

- In smart healthcare, in-network computing can help resolve contexts and ensure security and dependability, as well as provide low-latency responses to urgent situations.

- In smart campus, in-network computing services can be used to provide context for different applications.

4.6. Security and Privacy

Security and privacy is crucial to all the IoT applications including the use cases discussed in Section 5. In one recent demonstration, it was shown that passive tire pressure sensors in cars could be hacked and used as a gateway into the automotive system [38]. Though ICN includes data-centric security features the mechanisms have to be generic enough to satisfy multiplicity of policy requirements for different applications. Furthermore security and privacy concerns have to be dealt in a scenario-specific manner with respect to network function perspective spanning naming, name-resolution, routing, caching, and ICN-APIs. In general, we feel that security and privacy protection in IoT systems should mainly focus on the following aspects: confidentiality, integrity, authentication and non-repudiation, and availability.

Implementing security and privacy methods faces different challenges in the constrained and infrastructure part of the network.

- In the resource-constrained nodes, energy limitation is the biggest challenge. As an example, let us look at a typical sensor tag. Suppose the tag has a single 16-bit processor, often running at 6 MHz to save energy, with 512Bytes of RAM and 16KB of flash for program storage. Moreover, it has to deliver its data over a wireless link for at least 10,000 hours on a coin cell battery. As a result, traditional security/privacy measures are impossible to be implemented in the constrained part. In this case, one possible solution might be utilizing the physical wireless signals as security measures [39] [29].

- In the infrastructure part, we have several new threats introduced by ICN-IoT [42]:

1. We need to ensure the name of a network element is issued by a trustworthy organization entity such as in [41], or by its trusted delegate.

2. An intruder may gain access or gather information from a resource it is not entitled to. As a consequence, an adversary may examine, remove or even modify confidential information.

3. An intruder may mimic an authorized user or network process. As a result, the intruder may forge signatures, or impersonate a source address.

4. An adversary may manipulate the message exchange process between network entities. Such manipulation may involve replay, rerouting, mis-routing and deletion of messages.

5. An intruder may insert fake/false sensor data into the network. The consequence might be an increase in delay and performance degradation for network services and applications.

Finally, in addition to the above requirements, specific IoT applications may impose specific challenges on privacy that impact both applications and the ICN-IoT network:

- In smart homes, the access to networked information should be shielded to protect the privacy of people, for example, cross-correlation of device activity patterns to infer higher-level activity information.

- In smart grids, energy consumptions profiles should not been never disclosed at a fine granularity since from them it is possible to violate the privacy of users.

- In smart transportation, the habits of users can be inferred by looking at their movement patterns -- privacy protection is essential.

- In smart healthcare, personal medical data about patients should remain shielded to protect their privacy, implementing both regulatory requirements and current industry best practices.

- In smart campus, it is required to differentiate among different profiles and to allocate different rights and protection levels to them.

4.7. Energy Efficiency

All the optimizations for other components of the ICN-IoT system (described in earlier subsections) can lead to optimized energy efficiency. As a result, we refer the readers to read sections 4.1-4.6 for challenges associated with energy efficiency for ICN-IoT.

5. Popular Scenarios

Several types of IoT applications exists, where the goal is efficient and secure management and communication among objects in the system and with the physical world through sensors, RFIDs and other devices. Below we list a few popular IoT applications. We omit the often used term "smart", though it applies to each IoT scenario below, and posit that IoT-style interconnection of devices to make these environments "smart" in today's terms will simply be the future norm.

5.1. Homes

The home [10] is a complex ecosystem of IoT devices and applications including climate control, home security monitoring, smoke detection, electrical metering, health/wellness, and entertainment systems. In a unified IoT platform, we would inter-connect these systems through the Internet, such that they can interact with each other and make decisions at an aggregated level. Also, the systems can be accessed and manipulated remotely. Challenges in the home include topology independent service discovery, common protocol for heterogeneous device/application/service interaction, policy based routing/ forwarding, service mobility as well as privacy protection. Notably, the ease-of-use expectations and training of both users and installers also presents challenges in user interface and user experience design that are impacted by the complexity of network configuration, brittleness to change, configuration of trust management, etc. Finally, it is unlikely that there will be a single "home system", but rather a collection of moderately inter-operable collaborating devices. In addition, several IoT-enabled homes could form a smart district where it becomes possible to bargain resources and trade with utility suppliers.

Homes [12][13] faces the following challenges that are hard to address with IP-based overlay solutions: (1) context-aware control: home systems must make decisions (e.g., on how to control, when to collect data, where to carry out computation, when to interact with end-users, etc.) based upon the contextual information [14]; (2) inter-operability: home systems must operate with devices that adopt heterogeneous naming, trust, communication, and control systems; (3) mobility: home systems must deal with mobility caused by the movement of sensors or data receivers; (4) security: a home systems must be able to deal with foreign devices, handle a variety of user permissions (occupants of various types, guests, device manufacturers, installers and integrators, utility and infrastructure providers) and involve users in important security decisions without overwhelming them; (5) user interface / user experience: homes need to provide reasonable interfaces to their highly heterogeneous IoT networks for users with a variety of skill levels, backgrounds, cultures, interests, etc.

5.2. Enterprise

Enterprise building deployments, from university campuses [15] [55] [56] [57] to industrial facilities and retail complexes, drive an additional set of scalability, security, and integration requirements beyond the home, while requiring much of its ease of use and flexibility. Additionally, they bring requirements for integration with business IT systems, though often with the additional support of in-house engineering support.

Increasing number of enterprises are equipped with sensing and communication devices inside buildings, laboratories, and plants, at stadiums, in parking lots, on school buses, etc. A unified IoT platform must integrate many aspects of human interaction, H2M and M2M communication, within the enterprise, and thus enable many IoT applications that can benefit a large body of enterprise affiliates. The challenges in smart enterprise include efficient and secure device/data/resource discovery, inter-operability between different control systems, throughput scaling with number of devices, and unreliable communication due to mobility and telepresence.

Enterprises face the following challenges that are hard to address with IP-based overlay solutions: (1) efficient device/data/ resource discovery: enterprise devices must be able to quickly and securely discover requested device, data, or resources; (2) scalability: a enterprise system must be able to scale efficiently with the number and type of sensors and devices across not only a single building but multi-national corporations (for example); (3) mobility: a enterprise system must be able to deal with mobility caused by movement of devices; (4) security: security for IoT applications in the enterprise should integrate with other enterprise-wide security components.

5.3. Smart Grid

Central to the so-called "smart grid"[16] is data flow and information management, achieved by using sensors and actuators, which enables important capabilities such as substation and distribution automation. In a unified IoT platform, data collected from different smart grids can be integrated to reach more significant optimizations. The challenges for smart grid include reliability, real-time control, secure communications, and data privacy.

Deployment of the smart grid [17] [18] faces the following issues that are hard to address with IP-based overlay solutions: (1) scalability: tomorrow's electrical grids must be able to scale gracefully to manage a large number of heterogeneous devices; (2) real time: grids must be able to perform real-time data collection, data processing and control; (3) reliability: grids must be resilient to hardware/software/networking failures; (4) security: grids and associated systems are often considered critical infrastructure -- they must be able to defend against malicious attacks, detect intrusion, and route around disruption.

5.4. Transportation

We are currently witnessing the increasing integration of sensors into cars, other vehicles transportation systems [19]. Current production cars already carry many sensors ranging from rain gauges and accelerometers over wheel rotation/traction sensors, to cameras. While intended for internal vehicle functions, these could also be networked and leveraged for applications such as monitoring external traffic/road conditions. Further, we can build vehicle-to-infrastructure (V2I),Vehicle-to-Roadside (V2R), and vehicle-to-vehicle (V2V) communications that enable many more applications for safety, convenience, entertainment, etc. The challenges for transportation include fast data/device/service discovery and association, efficient communications with mobility, trustworthy data collection and exchange.

Transportation [19][20] faces the following challenges that are hard to address with IP-based overlay solutions: (1) mobility: a transportation system must deal with a large number of mobile nodes interacting through a combination of infrastructure and ad hoc communication methods; ; also, during the journey the user might cross several realms, each one implementing different stacks (whether ICN or IP); (2) real-time and reliability: transportation systems must be able to operate on real-time and remain resilient in the presence of failures; (3) in-network computing/filtering: transportation systems will benefit from in-network computing/filtering as such operations can reduce the end-to-end latency; (4) inter-operatibility: transportation systems must operate with heterogeneous device and protocols; (5) security: transportation systems must be resilient to malicious physical and cyber attacks.

5.5. Healthcare

As more embedded medical devices, or devices that can monitor human health become increasingly deployed, healthcare is becoming a viable alternative to traditional healthcare solutions [21]. Further, consumer applications for managing and interacting with health data are a burgeoning area of research and commercial applications. For future health applications, a unified IoT platform is critical for improved patient care and consumer health support by sharing data across systems, enabling timely actuations, and lowering the time to innovation by simplifying interaction across devices from many manufacturers. Challenges in healthcare include real-time interactions, high reliability, short communication latencies, trustworthy, security and privacy, and well as defining and meeting the regulatory requirements that should impact new devices and their interconnection. In addition to this dimension, assistive robotics applications are gaining momentum to provide 24/24 7/7 assistance to patients [49].

Healthcare [21][22] faces the following challenges that are hard to address with IP-based overlay solutions: (1) real-time and reliability: healthcare systems must be able to operate on real-time and remain resilient in the presence of failures; (2) inter-operability: healthcare systems must operate with heterogeneous devices and protocols; (3) security: healthcare systems must be resilient to malicious physical and cyber attacks and meet the regulatory requirement for data security and interoperability; (4) privacy: user trust in healthcare systems is critical, and privacy considerations paramount to garner adoption and continued user; (5) user interface / user experience: the highly heterogeneous nature of real-world healthcare systems, which will continue to increase through the introduction of IoT devices, presents significant challenges in interface design that may have architectural implications.

5.6. Education

IoT technologies enable the instrumentation of a variety of environments (from greenhouses to industrial plants, homes and vehicles) to support not only their everyday operation but an understanding of how they operate -- a fundamental contribution to education. The diverse uses of hobbyist-oriented micro-controller platforms (e.g., the Arduino) and embedded systems (e.g., the Raspberry PI) point to a burgeoning community that should be supported by the next generation IoT platform because of its fundamental importance to formal and informal education.

Educational uses of IoT deployments include both learning about the operation of the system itself as well as the systems being observed and controlled. Such deployments face the following challenges that are hard to address with IP-based overlay solutions: (1) relatively simple communications patterns are obscured by many layers of translation from the host-based addressing of IP (and layer 2 configuration below) to the name-oriented interfaces provided by developers; (2) security considerations with overlay deployments and channel-based limit access to systems where read-only use of data is not a security risk; (3) real-time communication helps make the relationship between physical phenomena and network messages easier to understand in many simple cases; (4) integration of devices from a variety of sources and manufacturers is currently quite difficult because of varying standards for basic communication, and limits experimentation; (5) programming interfaces must be carefully developed to expose important concepts clearly and in light of current best practices in education.

5.7. Entertainment, arts, and culture

IoT technologies can contribute uniquely to both the worldwide entertainment market and the fundamental human activity of creating and sharing art and culture. By supporting new types of human-computer interaction, IoT can enable new gaming, film/video, and other "content" experiences, integrating them with, for example, the lighting control of the smart home, presentation systems of the smart enterprise, or even the incentive mechanisms of smart healthcare systems (to, say, encourage and measure physical activity).

Entertainment, arts, and culture applications generate a variety of challenges for IoT: (1) notably, the ability to securely "repurpose" deployed smart systems (e.g., lighting) to create experiences; (2) low-latency communication to enable end-user responsiveness; (3) integration with infrastructure-based sensing (e.g., computer vision) to create comprehensive interactive environments or to provide user identity information; (4) time synchronization with audio/video playback and rendering in 3D systems (5) simplicity of development and experimentation, to enable the cost- and time-efficient integration of IoT into experiences being designed without expert engineers of IoT systems; (6) security, because of integration with personal devices and smart environments, as well as billing systems.

6. Informative References

[1] Cisco System Inc., CISCO., "Cisco visual networking index: Global mobile data traffic forecast update.", 2009-2014.

[2] Shafig, M., Ji, L., Liu, A., Pang, J., and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization.", Proceedings of the ACM Sigmetrics , 2012.

[3] The European Telecommunications Standards Institute, ETSI., "http://www.etsi.org/.", 1988.

[4] Global Intiative for M2M Standardization, oneM2M., "http://www.onem2m.org/.", 2012.

[5] Constrained RESTful Environments, CoRE., "https://datatracker.ietf.org/wg/core/charter/.", 2013.

[6] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and J. Wilcox, "Information-Centric Networking: Seeing the Forest of the Trees.", Hot Topics in Networking , 2011.

[7] Dong, L., Zhang, Y., and D. Raychaudhuri, "Enhance Content Broadcast Efficiency in Routers with Integrated Caching.", Proceedings of the IEEE Symposium on Computers and Communications (ISCC) , 2011.

[8] NSF FIA project, MobilityFirst., "http://www.nets-fia.net/", 2010.

[9] Kim, B., Lee, S., Lee, Y., Hwang, I., and Y. Rhee, "Mobiiscape: Middleware Support for Scalable Mobility Pattern Monitoring of Moving Objects in a Large-Scale City.", Journal of Systems and Software, Elsevier, 2011.

[10] Dietrich, D., Bruckne, D., Zucker, G., and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", IEEE Transactions on Industrial Electronics, 2010.

[11] Keith, K., Falco, F., and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security", NIST, Technical Report 800-82 Revision 1, 2013.

[12] Darianian, M. and Martin. Michael, "Smart home mobile RFID-based Internet-of-Things systems and services.", IEEE, ICACTE, 2008.

[13] Zhu, Q., Wang, R., Chen, Q., Chen, Y., and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things", IEEE/IFIP, EUC, 2010.

[14] Biswas, T., Chakrabort, A., Ravindran, R., Zhang, X., and G. Wang, "Contextualized information-centric home network", ACM, Siggcomm, 2013.

[15] Huang, R., Zhang, J., Hu, Y., and J. Yang, "Smart Campus: The Developing Trends of Digital Campus", 2012.

[16] Yan, Y., Qian, Y., Hu, Y., and J. Yang, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", IEEE Communications Survey and Tutorials, 2013.

[17] Miao, Y. and Y. Bu, "Research on the Architecture and Key Technology of Internet of Things (loT) Applied on Smart Grid", IEEE, ICAEE, 2010.

[18] Zhang, Y., Yu, R., Nekovee, M., Liu, Y., Xie, S., and S. Gjessing, "Cognitive Machine-to-Machine Communications: Visions and Potentials for the Smart Grid", IEEE, Network, 2012.

[19] Zhou, H., Liu, B., and D. Wang, "Design and Research of Urban Intelligent Transportation System Based on the Internet of Things", Springer Link, 2012.

[20] Zhang, M., Yu, T., and G. Zhai, "Smart Transport System Based on the Internet of Things", Applied Mechanics and Materials, 2012.

[21] Zhang, A., Yu, R., Nekovee, M., and S. Xie, "The Internet of Things for Ambient Assisted Living", IEEE, ITNG, 2010.

[22] Savola, R., Abie, H., and M. Sihvonen, "Towards metrics-driven adaptive security management in E-health IoT applications.", ACM, BodyNets, 2012.

[23] Jacobson, V., Smetters, D., Plass, M., Stewart, P., Thornton, J., and R. Braynard, "VoCCN: Voice-over Content-Centric Networks", ACM, ReArch, 2009.

[24] Piro, G., Cianci, I., Grieco, L., Boggia, G., and P. Camarda, "Information Centric Services in Smart Cities", ACM, Journal of Systems and Software, 2014.

[25] Ravindran, R., Biswas, T., Zhang, X., Chakrabort, A., and G. Wang, "Information-centric Networking based Homenet", IEEE/IFIP, 2013.

[26] Dannewitz, C., D’ Ambrosio, M., and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks", 2013.

[27] Chai, W., He, D., and I. Psaras, "Cache "less for more" in information-centric networks", ACM, IFIP, 2012.

[28] Eum, S., Nakauchi, K., Murata, M., Shoji, Yozo., and N. Nishinaga, "Catt: potential based routing with content caching for icn", IEEE Communication Magazine, 2012.

[29] Eum, S., Shvartzshnaider, Y., Francisco, J., Martini, R., and D. Raychaudhuri, "Enabling internet-of-things services in the mobilityfirst future internet architecture", IEEE, WoWMoM, 2012.

[30] Sun, Y., Qiao, X., Cheng, B., and J. Chen, "A low-delay, lightweight publish/subscribe architecture for delay-sensitive IOT services", IEEE, ICWS, 2013.

[31] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T., and M. Wahlisch, "Information Centric Networking in the IoT:Experiments with NDN in the Wild", ACM, ICN Siggcomm, 2014.

[32] Gronbaek, I., "Architecture for the Internet of Things (IoT): API and interconnect", IEEE, SENSORCOMM, 2008.

[33] Tian, Y., Liu, Y., Yan, Z., Wu, S., and H. Li, "RNS-A Public Resource Name Service Platform for the Internet of Things", IEEE, GreenCom, 2012.

[34] Roussos, G. and P. Chartier, "Scalable id/locator resolution for the iot", IEEE, iThings,CPSCom, 2011.

[35] Amadeo, M. and C. Campolo, "Potential of information-centric wireless sensor and actor networking", IEEE, ComManTel, 2013.

[36] Nelson, S., Bhanage, G., and D. Raychaudhuri, "GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet", ACM, MobiArch, 2011.

[37] Trappe, W., Zhang, Y., and B. Nath, "MIAMI: methods and infrastructure for the assurance of measurement information", ACM, DMSN, 2005.

[38] Rouf, I., Mustafa, H., Taylor, T., Oh, S., Xu, W., Gruteser, M., Trappe, W., and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study", USENIX, 2010.

[39] Liu, R. and W. Trappe, "Securing Wireless Communications at the Physical Layer", Springer, 2010.

[40] Xiao, L., Greenstein, L., Mandayam, N., and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels", IEEE Transactions on Wireless Communications, 2008.

[41] Sun, S., Lannom, L., and B. Boesch, "Handle system overview", IETF, RFC3650, 2003.

[42] Liu, X., Trappe, W., and Y. Zhang, "Secure Name Resolution for Identifier-to-Locator Mappings in the Global Internet", IEEE, ICCCN, 2013.

[43] Boguna, M., Fragkiskos, P., and K. Dmitri, "Sustaining the internet with hyperbolic mapping", Nature Communications, 2010.

[44] Shang, W., "Securing building management systems using named data networking", IEEE Network 2014.

[45] Fayazbakhsh, S. and et. et al, "Less pain, most of the gain: Incrementally deployable icn", ACM, Siggcomm, 2013.

[46] Burke, J. and et. et al, "Securing instrumented environments over Content-Centric Networking: the case of lighting control", INFOCOM, Computer Communications Workshop, 2013.

[47] Li, S., Zhang, Y., Dipankar, R., and R. Ravindran, "A comparative study of MobilityFirst and NDN based ICN-IoT architectures", IEEE, QShine, 2014.

[48] Grieco, L., Alaya, M., and K. Drira, "Architecting Information Centric ETSI-M2M systems", IEEE, Pervasive and Computer Communications Workshop (PERCOM), 2014.

[49] Grieco, L., Rizzo, A., Colucci, R., Sicari, S., Piro, G., Di Paola, D., and G. Boggia, "IoT-aided robotics applications: technological implications, target domains and open issues", Computer Communications, Volume 54, 1 December, 2014.

[50] Quan, Wei., Xu, C., Guan, J., Zhang, H., and L. Grieco, "Scalable Name Lookup with Adaptive Prefix Bloom Filter for Named Data Networking", IEEE Communications Letters, 2014.

[51] Wang, Yi., Pan, T., Mi, Z., Dai, H., Guo, X., Zhang, T., Liu, B., and Q. Dong, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage Bloom filters", INFOCOM, 2013.

[52] So, W., Narayanan, A., Oran, D., and Y. Wang, "Toward fast NDN software forwarding lookup engine based on Hash tables", ACM, ANCS, 2012.

[53] Amadeo, M., Campolo, C., Iera, A., and A. Molinaro, "Named data networking for IoT: An architectural perspective", IEEE, EuCNC, 2014.

[54] Sifalakis, M., Kohler, B., Christopher, C., and C. Tschudin, "An information centric network for computing the distribution of computations", ACM, ICN Sigcomm, 2014.

[55] Lu, R., Lin, X., Zhu, H., and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots", INFOCOM, 2009.

[56] Wang, H. and W. He, "A reservation-based smart parking system", The First International Workshop on Cyber-Physical Networking Systems, 2011.

[57] Qian, L., "Constructing Smart Campus Based on the Cloud Computing and the Internet of Things", Computer Science 2011.

[58] Project, BonVoyage., "From Bilbao to Oslo, intermodal mobility solutions, interfaces and applications for people and goods, supported by an innovative communication network", Call H2020-MG-2014, 2015-2018.

Authors' Addresses

Prof.Yanyong Zhang
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: yyzhang@winlab.rutgers.edu

Prof. Dipankar Raychadhuri
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: ray@winlab.rutgers.edu

Prof. Luigi Alfredo Grieco
Politecnico di Bari (DEI)
Via Orabona 4
Bari 70125
Italy

Email: alfredo.grieco@poliba.it

Prof. Emmanuel Baccelli
INRIA
Room 148, Takustrasse 9
Berlin 14195
France

Email: Emmanuel.Baccelli@inria.fr

Jeff Burke
UCLA REMAP
102 East Melnitz Hall
Los Angeles, CA 90095
USA

Email: jburke@ucla.edu

Ravishankar Ravindran
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ravi.ravindran@huawei.com

Guoqiang Wang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: gq.wang@huawei.com