ICN Publisher-Subscriber Models: Challenges and Group-based Communication

Boubakr Nour, Kashif Sharif, Fan Li, Song Yang, Hassine Moungla, Yu Wang

ICN Publisher-Subscriber Models: Challenges and Group-based Communication

Boubakr Nour, Kashif Sharif, Fan Li, Song Yang, Hassine Moungla, and Yu Wang

***Abstract*—Information centric networking (ICN) architectures use pull-based communication with single interest response pairs. Content retrieval using this model is very efficient when the content already exists in the network. Subscription services for content that is being dynamically generated or will be generated in future, do not work well with pull based systems. In this work we investigate ICN as publisher-subscriber communication enabler, and present its challenges and limitation. Based on the observations, we propose a group-based subscription architecture, which enables not only a seamless publisher-subscriber model, but also authentication, access control, and group management features, without modifying ICN principles. Compared to traditional pull-based subscription, we are able to achieve lesser control overhead, with added security and privacy features. The performance analysis also shows that with semi-persistent interest, the memory requirements of the core nodes can be kept at minimal levels. We also identify interesting research challenges which can enable a rich ICN ecosystem for different types of services.**

I. Introduction

The fundamental working principal for an Information Centric Network (ICN) [1] is based on data-oriented nature of communication. This is in contrast to the host-oriented communication, where the network itself is oblivious to the nature or location of content. The information centric nature of network is not only aware of the content, but also facilitates in optimizing the retrieval process. Hence, the future Internet architecture [2] will have a more interactive network layer, as compared to the existing one. The building block in ICN is content name, which should be unique, secure, and location-independent. Hence, ICN routers use name-based forwarding rules to retrieve and deliver the content. Although the physical communication channels need to be secure, more emphasis is placed on securing the content itself. Furthermore, in-network data caching is a principal network element in ICN.

A number of ICN architectures have been proposed in literature. The two major designs are Named-Data Networking (NDN) [3] and Content Centric Networking (CCN) [4], both of which implement pull-based Interest-Data paradigm. *Interest* packets are triggered from a consumer to request some content from the network. The important point to note is that the content is not requested from a specific host, but rather from any publisher or replica node. The content is then sent in a *Data* packet which takes the same path to the requester as that of request packet. This communication model of single-interest single-data is the fundamental working principal between consumer and producer/publisher. By using this communication model, different requests to the same data are aggregated by intermediate router, which decreases the load on network. It is also effective for rapid content retrieval, as multiple nodes in the network can cache the content and fulfill future requests. However, using the same model to ensure a unicast machine-to-machine communication is challenging and in violation of request aggregation principle. Furthermore, all interfaces that have been aggregated will receive the same content, regardless of their authentication. Solutions to most of these challenges are left for application layer to tackle. ICN core routers only maintain three data structures: Cache Store (CS) to manage in-network content caching, Pending Interest Table (PIT) to ensure a receive-driven design with request aggregation, and Forwarding Information Base (FIB) to forward the interest based on its name.

Recent years with rapid evolution of social media, has seen the drive towards content-oriented behavior of communication. Hence in ICN the core data retrieval focuses on content itself, and not on the location of provider. Unicast communication between requester and provider is replaced with anycast request and multicast response mechanism. Given this fundamental design constraint, traditional communication services which do not strictly follow request-response have to adapt. A wide range of common applications in Internet are subscription based, where information is generated by a provider over a long period of time, and periodically delivered to a set of subscribers. This is commonly known as the Publisher-Subscriber (pub-sub) design. Real applications that may use this model are live video/audio feeds, weather information services, monitoring services, event triggered information distribution, etc.

The contribution of our work is multifold. We first discuss the pub-sub communication models from ICN perspective, review the existing solutions, and highlight the current limitation and challenges. Then, we design and implement a clean publisher-subscriber model using group mechanics. Moreover, the model defines how the subscription is to be managed, and how the keys are to be distributed. We demonstrate that, with minimal changes to interest packet structure, semi-persistent PIT entries, and specialized table at intermediate nodes, a publisher driven key distribution mechanism can achieve efficient and secure access control for pub-sub communication in ICN. Furthermore, it can be seamlessly integrated with NDN or CCNx implementations, and has been verified by simulation experiments.

B. Nour, K. Sharif (co-corresponding author), F. Li (co-corresponding author), and S. Yang are with School of Computer Science, Beijing Institute of Technology, Beijing, China.

H. Moungla is with University of Paris Descartes, Paris, and Telecom SudParis, Scaly, France.

Y. Wang is with University of North Carolina at Charlotte, Charlotte, NC, USA.

II. Publisher-Subscriber Communication Requirements

In order to design a pub-sub communication model for ICN, different subscription mechanisms need to be explored. Varying requirements from these models may require changes to ICN primitives at the network layer. Different users may subscribe for a service offered by a publisher, where the information generated may be continuous (e.g. live video feed), periodic (e.g. weather updates), or when specific events occur (e.g. patient pulse rate change). Hence, we classify the traffic into following categories based on its behavior.

Single-Request Single-Response: A requester node sends one request packet asking for some content by using its content name. The original content provider or a replica-node replies and delivers the content to the requester. Any update to the same content can be obtained by sending another request. This model works perfectly when the content required already exists, or can be made available before the interest entry expires in the core network.

Single-Request Multiple-Responses: This model is used when a subscriber sends one request asking for data which may comprise of multiple responses spread over time. The number of responses can vary depending on application service.

- Periodic Delivery: A consumer sends one request packet asking for periodic data identified by name after a specific time interval, e.g. receiving sensor value every 10 minutes for next 2 hours. The amount of data is limited and bounded by time.
- *N* Responses: A subscriber node sends one request packet for a specific number of responses, e.g. next *n* pieces of information generated for a certain topic (next 10 frames of a video). Here, the amount of data is limited but not bounded by time.
- Conditional Delivery: A subscribing user sends a request packet to receive data from publisher only if certain conditions are met or events triggered, e.g. notify when mentioned in a tweet. The information flow is not bounded in time.

A vast majority of current Internet applications, especially mobile and Internet of Things (IoT) technologies, use these communication models for subscription services. Moreover, many of these applications also use the information centric communication patterns [5]. The requirement to harness the benefits of rapid content deliver of ICN along with pub-sub communication, presents a unique and interesting challenge. NDN and CCNx are primarily request-response (pull-based) systems. Although they are well designed and efficient in working, implementing a pub-sub model without modifying any of their primitives is non-trivial.

III. Challenges of Publisher-Subscriber Models

Publisher-Subscriber communication is an active research topic in ICN, and a handful of solutions have been proposed. These can be fundamentally classified into four categories, as shown in Figure 1, based on the available literature.

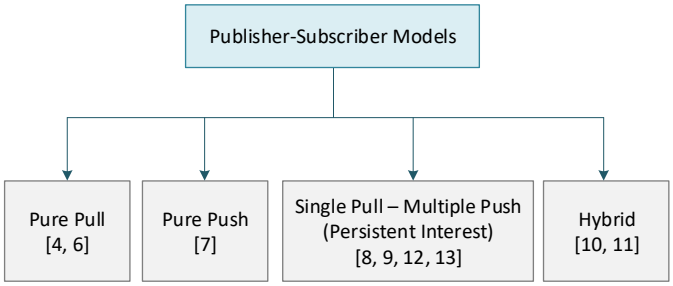

Fig. 1: Classification of ICN Publisher-Subscriber Solutions.

A. Existing Solutions

In [6] authors use a combination of push and pull mechanisms, where data is pushed to a common repository from publisher, and subscribers pull from the repository as they desire. However, from a consumer request perspective, it sends one request for one response to the cache, hence a pure pull system. Work in [7] addresses the pub-sub communication in mobile ad hoc networks in a pure push fashion, where FIB table is used to deliver the generated data by the publisher rather than PIT. In [8] authors propose the use of persistent interests, which remain in PIT for longer periods of time. Moreover they use hop-by-hop acknowledgments for reliability. The model in [9] merges persistent and reverse interest packets. The idea is to add a persistence value in interest packet which indicates the number of responses required, and intermediate nodes store this values in PIT table. [10] proposes multiple strategies to deliver push-based IoT traffic. The publisher sends an interest notification for periodical and event-triggered content, which includes the data in interest name components, or by sending unsolicited data. Authors in [11] propose a lightweight solution for IoT, which is an enhanced version of CCN-lite architecture that supports publisher-subscriber communication. It uses specialized packets and table structures for subscriptions. Work in [12] studies the use of persistent interests to support push-based communication in Interest-based ICN architectures, while [13] proposes an adaptive persistent interest forwarding scheme to overcome the long-lived path.

B. Limitations and Challenges

The limitations of exiting works are not based on their working principal, rather they are more related to security, scalability, and seamless integration into ICN implementations. Here, we dissect both pull-based and push based mechanisms, to explore the technical requirements for an optimal solution.

Pull-based Model: This model can be used to realize periodic data subscription, where subscribers are required to send periodic interest packets asking for the data. The publisher replies to each request individually. The relation between content name and amount of content is very important. For example, *local_weather_update* is the content name where the content will change over time. Every time this information is required by a mobile device, a new request has to be sent. This solution fundamentally creates an overhead for publisher, as the requests arrive at disjoint times. Intermediate node interest

aggregation also suffers due to the same reason. Content caching by intermediate nodes may provide some efficiency, but the overhead of request packet from every subscribing node still burdens the network. Once the content becomes stale, caches need to be expunged. Furthermore, this model has no authentication mechanism, as for who can access the information. Application layer authentication does not work, as ICN supports in-network caching and name aggregation. Existing pull-based models also lack security mechanisms. If an application layer public-private key mechanism is used, then the publisher has to obtain public key of each individual subscriber, and generate unique replies every time. This again results in caching and name aggregation failure in core network. If a common key is used for encryption and decryption, then key safekeeping, updation, and distribution becomes a challenge. Lastly, event triggered communication model cannot be optimally implemented.

Push-based Model: This model can be implemented in two ways. One method is that the publisher injects data into the network for subscribers, where the list of subscribers is known to the publisher. This method violates the interest-response mechanism. Data packets with no entry in PIT will be dropped in core network. Even if the core architecture is changed, the publisher will generate as many responses as the subscribers, and performs a unicast push for each. This increases the overhead, and takes away the advantage of ICN. The second method is to create semi-permanent entries (which are not removed after a single response packet), in core routers using a specialized interest packet. This method is interesting but has following challenges associated to it.

- Subscribers in the same domain may subscribe for same topic but use different frequency/rate to receive data. Due to PIT aggregation feature, all subscribers within the same domain will receive data with no regards to subscription frequency rate.
- Keeping PIT compact and fresh requires persistent entries to expire after some time. Once the timer expires, a burst of interest packets will be injected into the network from all down stream subscribers. This burst may force intermediate nodes to overflow.
- No access control mechanism has been proposed for this model. Once the data is injected into the network, anyone can access it. Encryption mechanisms suffer from same limitations as that of the earlier method.
- To achieve minimal levels of access control, each subscriber has to generate a unique interest packet, achievable only by a unique interest name (otherwise in-network aggregation will combine requests). This can be done by adding a subscriber ID to content name. Content name standardization itself is an open research challenge. At the same time, such uniqueness takes away the caching and aggregation benefits from ICN.

In light of these challenges, it is better to use a hybrid $(1 : N)$ publisher-subscriber group model for dynamically generated content. It is important to note that, the content already published can be retrieved individually by subscribers using native NDN interest-data model. Access control can be managed by group operations, and group based encryption mechanisms can be used for privacy. This makes the overall architecture secure, scalable, NDN & CCNx compliant, and efficient.

IV. Group-based Publisher-Subscriber Architecture

The proposed Group-based Publisher-Subscriber (GbPS) architecture can be divided into two parts. First part deals with the subscription management, and the second handles key computation. It is an inherently secure design, which treats members as a $(1 : N)$ pub-sub group. Without violating ICN's working principles, we introduce new types of interest and data packets, with an additional table at the intermediate nodes. Figure 2 depicts the overall communication process and the message structure for each step of communication. Both publisher and subscriber modules are shown with corresponding components, which are explained in the sections below.

A. Design

The publisher-subscriber model can essentially be viewed as a group management scenario, with specific management operations. In order to keep it simple, we propose the use of three operations, i.e. Join/Subscribe, Leave, and Evict. A subscriber sends a subscription request (interest packet) to join a topic (group) offered by a publisher. Similarly, it may request to leave, or the publisher can evict a subscriber for different reasons. The objective here is not to propose a new group management protocol but to add simple yet effective procedures which can enable secure group management communication. We use the term *group* and *topic* in a loosely interchangeable manner.

Naming Convention: The complete process of subscription and data delivery is shown in Figure 2. The top part shows the naming convention used in this work. The name of subscribeable content is different from the interest name used for group operations, where later is an extension of former with added semantics. The content/topic/group name is used to represent the actual content, which is used for content retrieval. On the other hand, the operation names contain added name-components to represent two elements: desired operation and requesting subscriber. This two-element name structure is used to enable unicast communication between subscriber and publisher. In its absence, the interest will be aggregated in intermediate network and no unique identification of subscriber will be possible.

Pub-Sub Modules: Subscriber and publisher systems contain specialized modules as shown in Figure 2. Subscriber implements a special *Subscription Controller*, which acts as a connecting point for different topics. Once the application requests subscription of a specific topic, the Subscription Controller creates and publishes a globally reachable service point. Signature can be used to verify the authenticity and authorization of a user. We assume that efficient and secure signature verification mechanisms are available which eliminate the risk of forging signatures [14]. This service point is represented by a name which is similar to content name.

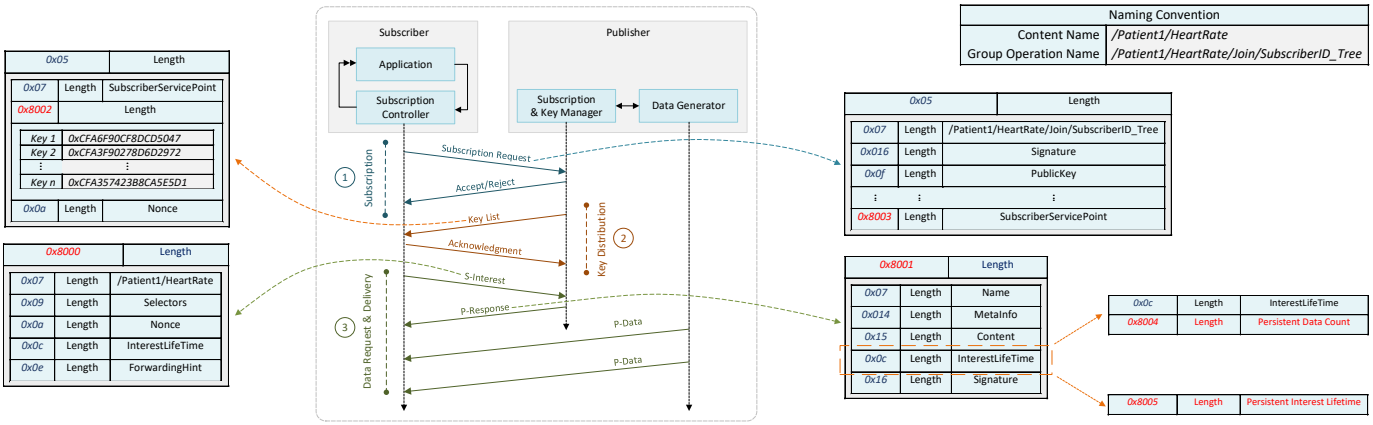

Fig. 2: GbPS communication (shown with NDN Type-Length-Value format).

On the publisher side, there are two sub-modules. a) *Data Generator* module is responsible for generating & publishing the actual content. b) *Subscription & Key Manager* module is responsible of subscriber authentication, as well as generation, management, and distribution of keys. It also provides the Data Generator with key that is used for encryption.

B. Subscription & Data Processing

Join Operation: The join operation is a three step process for subscribing to a specific topic, which are: Join request & authentication, Distribution of key set, and Data request & reply. The subscriber initiates a join request using the operation interest name as described earlier. ICN packets are a collection of *Type-Length-Value* (TLV) fields, hence signature, public keys, and subscriber service point information can be easily added to interest packets. The important point to note is that, the communication is unicast between *subscription controller* and *subscription manager*, as shown in Figure 2. The response to this request can contain additional TLVs to indicate approve or deny decision. It is assumed that existing mechanisms of signatures, password, biometric data, etc. are sufficient to authenticate a request. In case of an accepted subscription, the Subscription & Key Management module maintains the binding of subscriber to its keys and service point name. Following this, the group keys are updated and re-distributed in the group, which is a publisher initiated unicast process. The interest message for key distribution has piggy-backed data in form of encrypted key list, which can later be used by the subscriber to decrypt subscribed data.

Both of the above mentioned steps have to be unicast, otherwise authentication of individual subscribers and key distribution to specific group members is not possible. Additional TLVs do not change ICN request-response principle, and has no effect on the intermediate node processing. It is important to note that, if unicast delivery of content is achieved, then the benefit of aggregation and caching is lost. Hence, we use group keys to secure communication, and the keys themselves are generated/changed and distributed within the group.

Data Delivery: Once the new subscriber has received the group keys, it can generate an interest for data (with actual content name). This interest (*S-Interest*) is not different in structure as compared to a generic NDN or CCNx interest packet. However, the Type value used is *0x8000*. This enables the distinction between the two types, so that intermediate nodes can take appropriate action. Upon receiving S-Interest, the intermediate node creates an entry in *Subscription Interest Table* (SIT). Hence, each node in our model, has an additional table to keep track of interests related to subscriptions. The table, unlike PIT, retains entries for a longer period of time. Fundamentally, an intermediate node has two possible scenarios.

(a) S-Interest arrives with no related entry in SIT. In this case, an entry is created with interface ID, and packet is processed using existing forwarding strategy. It is important to note that individual timers are kept for each requesting interface. In case, when there are no entries along the entire path, S-Interest will reach the provider. Provider then, generates a persistent response (P-Response), which does not contain data, but contains specialized TLV as show in Figure 2. This TLV is then processed at each return-path node to update the persistent interest lifetime value.

(b) S-Interest arrives at intermediate nodes and an existing entry is present in SIT. The node first forwards the signature of requesting subscriber to the publisher's subscription manager and verifies that the node has been previously authenticated. On positive response, the subject node adds the interface ID to list, sets timer to minimum timer value of other associated interfaces, and generates a persistent response packet.

The persistent response packet (either from publisher or intermediate node) specifies the time/count of responses before the entry is expunged from SIT. As seen from Figure 2, we propose the use of a counter in combination with a maximum time limit, where expiration of either will remove the entry. We also propose that a maximum limit must be set by the network for any timer value (10 minutes in this work). Every intermediate node receiving the persistent response, sets the appropriate value for outgoing interface in SIT. Once it reaches the subscriber, it can record the specified time limit, and must rejoin the group after expiration.

Subscribe-able data is generated by the Data Generation

process at publisher. It is published using the generic name structure. Once the publishing node has subscribers in SIT, the data is forwarded onto the interface as long as it has a valid timer. This is the only deviation from pull-based principle, where PIT entry is removed once the data is forwarded back towards consumer. This process ensures that multiple data packets are forwarded before the entry expires, or is renewed by the consumer. Renewal process can be identical to join operation. The overall benefit of such a mechanism is that, multicast and in-network data replication still holds, while secure subscription management is also available.

Leave Operation: When the application no longer desires subscription, it initiates the leave operation through Subscription Controller. The publisher responds with an acknowledgment to the subscriber. Immediately after the request is received, the publisher removes the subscriber from its list, and updates the group keys. These keys are then unicasted to the remaining subscribers. It is not possible for the publisher to instruct intermediate nodes for SIT entry removal, as there may exist other active subscribers on the same downstream interface. Actually it is the key change that ensures that only active nodes are able to decrypt the information. Moreover, we assume that the trusted subscribers do not collude with non-subscribers by sharing the keys and data streams.

In case where publisher wants to revoke a subscriber access, it sends a unicast message to that subscriber's service point, with piggybacked revocation notice. Publisher immediately changes the keys, and performs redistribution operation.

C. Key Computation

In order to ensure that only authenticated subscribers can obtain data, it has to be encrypted. Using the public keys of subscriber is not a scalable solution, as the publisher will be required to generate as many encrypted packets for same content as the number of subscribers. It is more suitable to use a group key with secure group key management system. The objective of this paper in not to develop a new group key generation solution for ICN. Rather we use Logical Key Hierarchy mechanism in our pub-sub communication architecture which uses a tree structure. For each join or leave operation, all keys in the path from the subscriber location (leaf) to the root are changed. This change ensures both backward and forward secrecy requirements. The publisher is responsible to maintain list of subscribers, and implements the centralized key management process. Since LKH is based on a balanced tree to manage group membership, we prefer to use algorithm described in [15], which efficiently keeps the tree balanced in highly dynamic groups.

Join/Leave Operation: After receiving the join request, the node must be authenticated using the credentials provided in the interest packet. Authenticated subscribers are then provided with necessary keys using their subscriber service point name. In this work, key changes due to join operation are delivered after a time delay, where the delay is dependent on the number of join requests per unit time. The maximum delay is set to 2 seconds. This time delay reduces the key distribution communication overhead. When a subscriber leaves the group or is forcefully evicted, the keys are changed & distributed immediately without delay.

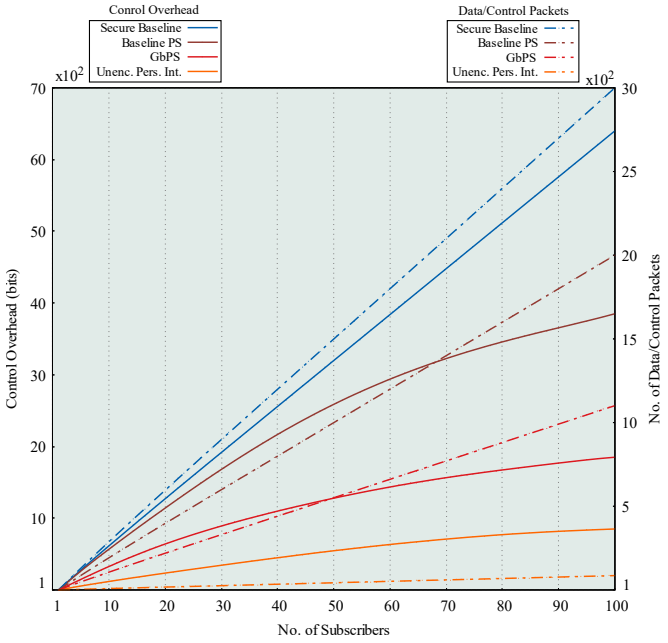

Fig. 3: Control overhead performance.

V. Performance & Compatibility

The performance of GbPS has been evaluated to determine relative control overhead created by group management requests, key exchanges, and entries made in PIT/SIT at intermediate nodes. It is important to note that native ICN design does not support secure group communication. Hence, we use Baseline Pub-Sub (PS) as a pull based model, where every subscriber sends an interest for each required content packet. In-network name aggregation is also utilized in this mechanism. Moreover, Secure Baseline uses an encrypted version of Baseline PS, which creates unique encryption for each subscriber. In control overhead measurements, we also show unencrypted persistent interest performance, which uses a single interest to obtain multiple data packets.

Control Overhead: In this experiment, we measure the amount of non-data bits generated by subscribers or publishers, against increasing number of subscribers. All subscribers request the same published content. Figure 3 shows the overhead from two perspectives. The solid lines represent the overhead performance measured in bits (left vertical scale), and dotted lines represent the number of packets (right vertical scale). With an average path length of 6 hops and data rate of 5 packets per second, we observe that, as the number of subscribers increases, the chance of finding existing entries in intermediate nodes also increases. Hence all different algorithms see a tapering-off of control information at higher subscriber numbers. Encrypted baseline individually identifies the subscriber, so that the publisher can encrypt the content for it (unicast), which creates a continuous growth in overhead. Removing encryption gives less overhead, but still requires subscribers to generate as many interest packets as

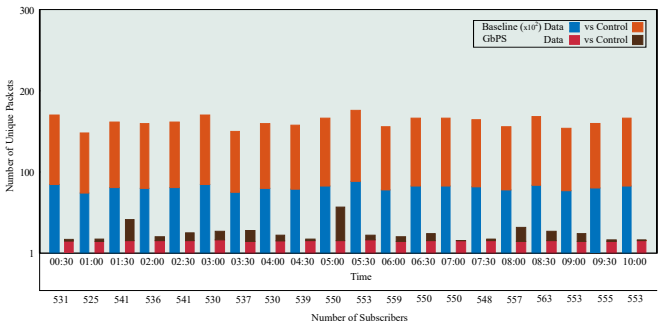

Fig. 4: Live video stream control and data packets.

data. Unencrypted persistent interest algorithm gives the least overhead as a single interest delivers multiple data packets. The dotted lines show the number of data/control packet in different algorithms. Baseline PS has a 1 to 1 ratio between data packet generated against interest packets. GbPS generates more number of control packets as compared to unencrypted persistent interest algorithm, and hence has less data-control ratio. But at the same time it is able to provide a well defined secure group operation. The benefits achieved are far more in significance than the ratio.

Live Video Stream Analysis: To show the effectiveness and performance of GbPS in real-world scenario, we observed the number of subscribers and data exchanges in NASA Live Channel on YouTube for 10 minutes. The viewers/subscribers join or leave at will in real-time. Each response packet contains 60 frames from producer (Youtube stats). Figure 4 shows the number of both unique data and control packets exchanged. In case of secure baseline, the publisher needs to perform as many encryption operations as that of subscribers, and send these unique packets to individual subscribers, even though the packets contain the same frames. By using GbPS, only one encryption operation is used, hence the data packets are very less. The spikes at 1:30 and 5:00 are due to changes in subscribers which requires new keys to be distributed.

Memory Requirements: This experiment analyzes the average memory required at each node to store the interest entries. Baseline PS continuously generates interest requests, hence the memory requirement increases as the number of subscribers grow. On the other hand, keeping subscribed interests for longer period of time combined with aggregation has far less memory requirements. Figure 5 shows the effect for GbPS only, where we evaluate it in relation to different number of uniquely published content and number of subscribers. Memory requirement increase is impacted more by the number of unique content which can be subscribed, rather than the number of subscribers.

Compatibility with NDN & CCNx: The fundamental designs of NDN and CCNx 0.x are almost identical. Later NDN protocol modifications have added newer features independently from CCNx 1.x. To ensure compatibility with either of the architectures, we summarize the technical design choices in Table I. GbPS can be implemented in both of them, and some of the design choices actually support the proposal put forward in this article.

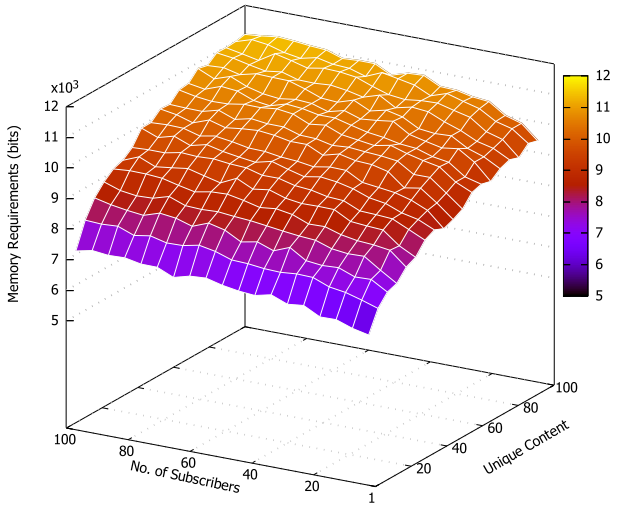

Fig. 5: Avg. storage req. for GbPS at intermediate nodes.

VI. Conclusion and Research Directions

Providing a publisher-subscriber model in ICN with security/access control, without violating the single request-response primitive is very challenging. However, by using a specialized table at intermediate nodes and semi-persistent interest, it is possible to enforce access control on a group of subscribers. Publisher-subscriber communication is an integral part of modern Internet, and by using grouping mechanisms along with key management, it can become an integrated part of NDN & CCN architecture.

Future Directions: There are a number of research directions which can be explored based on this work:

- Using *Interest* to carry data or control information will require standardization of type codes for TLVs.
- Although ICN enables multicast from forwarding perspective, but group management and related control messaging is an extremely important area, which should be made integral part of ICN.
- Group based key mechanisms also need to be explored for ICN, especially from key distribution overhead optimization perspective.
- An important direction specific to this work is to analyze the use of cache stores as distributors, and enforce access control in sub-groups through them.

Acknowledgments

The work of F. Li is partially supported by the National Natural Science Foundation of China No. 61772077, 61370192, and the Beijing Natural Science Foundation No. 4192051. The work of S. Yang is funded by National Natural Science Foundation of China No. 61802018, and Beijing Institute of Technology Research Fund Program for Young Scholars.

References

[1] G. Xylomenos, C. N. Ververidis *et al.*, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

TABLE I: NDN & CCNx 1.x Compatibility with GbPS.

	NDN	CCNx 1.x	GbPS
Semantics	- Unbounded name components - Partial name matching in hierarchy	- Unbounded name components - Full name matching	- Requires full name matching - Uses basic naming convention with name components
Encoding	- TLV format (outer & inner TLVs)	- XML Encoded → TLV format - Fixed header with TLVs within packets	- Uses TLVs, and can be stored in either of packet structure
Naming	- Proposal to have zero or multiple data packets for one Interest - Digest as last part of name	- Free to define name components - Implicit digest used as part of hash restriction for unique name	- Uses *Subscriber_ID* to enable uniqueness in name when desired - Digest can be used for same
Interest Aggregation	- Available	- Available	- Capitalizes on aggregation for encrypted content - Keeps individual timers for return interfaces
Opportunistic Caching	- Available	- Available	- Does not require caching due to nature of data
Forwarding	- Assumes loop freedom - Nonce for detection of duplicates - Hop limits	- Assumes loop freedom - Hop limits	- Additional SIT and semi-persistent Interest - This does not interfere with the forwarding process
Data-Centric Security	- TLV for signature	- Supports signature for Interests - CCNx Key Exchange Protocol	- Uses signatures for user authentication - Key Exchange Protocol can be used to encrypt, in addition to other key distribution methods

[2] J. Pan, S. Paul, R. Jain, "A survey of the research on future Internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, 2011.

[3] L. Zhang, D. Estrin *et al.*, "Named Data Networking (NDN) Project," *Technical Report NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[4] Project CCNx. [Online]. Available: http://www.ccnx.org/

[5] M. Amadeo, C. Campolo *et al.*, "Information-Centric Networking for the Internet of Things: Challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[6] T. Yagyu, K. Nakamura *et al.*, "Content-based Push/Pull Message Dissemination for Disaster Message Board," in *ACM International Conference on Information-Centric Networking*, 2015, pp. 205–206.

[7] A. Detti, D. Tassetto *et al.*, "Exploiting content centric networking to develop topic-based, publish-subscribe MANET systems," *Ad hoc networks*, vol. 24, pp. 115–133, 2015.

[8] C. Tsilopoulos and G. Xylomenos, "Supporting diverse traffic types in information centric networks," in *ACM SIGCOMM workshop on Information-Centric Networking*, 2011, pp. 13–18.

[9] B. Nour, K. Sharif *et al.*, "A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study," in *IEEE Global Communications Conference*, 2017, pp. 1–6.

[10] M. Amadeo, C. Campolo *et al.*, "Internet of Things via Named Data Networking: The support of push traffic," in *IEEE International Conference on Network of the Future*, 2014, pp. 1–5.

[11] H. Wang, S. S. Adhatarao *et al.*, "COPSS-lite: Lightweight ICN Based Pub/Sub for IoT Environments," *CoRR*, vol. abs/1706.03695, June 2017.

[12] P. Moll, D. Posch *et al.*, "Investigation of push-based traffic for conversational services in named data networking," in *IEEE International Conference on Multimedia & Expo Workshops*, 2017, pp. 315–320.

[13] P. Moll, J. Janda *et al.*, "Adaptive forwarding of persistent interests in named data networking," in *ACM Conference on Information-Centric Networking*, 2017, pp. 180–181.

[14] Y. Yu, Y. Li *et al.*, "Content Protection in Named Data Networking: Challenges and Potential Solutions," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 82–87, 2018.

[15] D.-W. Kwak, S. J. Lee *et al.*, "An efficient LKH tree balancing algorithm for group key management," *IEEE Communications Letters*, vol. 10, no. 3, pp. 222–224, 2006.