An SDN-aided Information Centric Networking Approach to Publish-Subscribe with Mobile Consumers

Paolo Benedetti, Agnese V. Ventrella, Giuseppe Piro, and Luigi Alfredo Grieco
Dept. of Electrical and Information Engineering (DEI),
Politecnico di Bari, v. Orabona 4, 70125, Bari, Italy;
Email: {paolo.benedetti, agnese.ventrella, giuseppe.piro, alfredo.grieco}@poliba.it
CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni

***Abstract*—Information Centric Networking represents a fundamental technology for the Future Internet. Its baseline functionalities can be extended to support publish-subscribe communication schema. But, in case of consumer mobility, its benefits slam against two main drawbacks. On one hand, available handover management solutions temporarily leave wrong forwarding information within network routers and do not take care of data dissemination across stale paths. On the other hand, mobile consumers inevitably lose content updates during the handover and definitively become unaware about the latest version of the content to request. By leveraging the potentials of the Software-Defined Network paradigm, this paper formulates new methodologies willing to solve these problems. Specifically, it proposes protocols for (1) dynamically updating forwarding functionalities through the control plane when the consumer detaches from the network and (2) restoring the synchronization between consumer and producer when the former one attaches to a new network attachment point. It addition, it develops preliminary analytical models for evaluating the average number of control information these protocols require to exchange per unit of time. The resulting analysis illustrates the pros and cons they achieve in different network configurations.**

***Index Terms*—SDN, ICN, publish-subscribe, consumer mobility**

I. INTRODUCTION

Information Centric Networking (ICN) emerged as one of the most promising communication paradigms for the Future Internet [1]. Its baseline networking primitives have been conceived to reach a data-centric usage of the current Internet infrastructure [2], [3]. In addition, content dissemination could be further improved thanks to the adoption of publish-subscribe communication mechanisms [4]–[8]. In case of mobility, however, the implementation of publish-subscribe mechanisms brings to two important drawbacks. During the handover, the consumer generally establishes a new multi-hop communication path with the remote producer, while the previous one (simply named stale path) remains still active for a given amount of time. Inevitably, content updates delivered across stale paths generate an unpleasant communication overhead on the data plane [6]. To make things worse, the consumer loses the synchronization with the producer and misses some content updates during the handover [9]. As a result, network elements and communicating peers should be configured on demand.

Due to its key ability to decouple control and data planes, the emerging Software-Defined Network (SDN) approach offers good opportunities to face this goal. The scientific literature already demonstrated how SDN could handle producer mobility in ICN networks [9]–[11]. However, currently, management of consumer mobility remains an open issue to solve.

In order to provide preliminary answers in this direction, this work focuses on the pull-based publish-subscribe mechanism presented in [6] and provides a three-folded contribution. First, to fix the presence of stale paths, two different protocols that dynamically remove wrong forwarding information when the consumer detaches from the network, namely Forwarding Information Base (FIB) inspection and neighbor inspection, are conceived. Second, two re-synchronization mechanisms to implement when the consumer attaches to a new network element, based on both SDN and ICN communication primitives, are designed. Third, preliminary analytical models have been developed for evaluating the upper bound of the average number of control information they need to exchange per unit of time, simply referred to as *control overhead*, and highlighting pros and cons they achieve in different network configurations. The proposed performance evaluation demonstrates that the overall control overhead increases with network size, publish-subscribe window size, and consumer speed. Moreover, the joint adoption of neighbor inspection and NDN-based re-sync protocols achieves the best performance.

The rest of the paper is organized as follows: Section II reviews the current state of the art on both ICN and SDN, by paying particular attention to publish-subscribe mechanisms and mobility. Section III presents the designed protocols addressing the publish-subscribe flaws due to consumer mobility. Section IV discusses the performance evaluation. Finally, Section V concludes the paper and draws future works.

II. RELATED WORK AND ESSENTIAL BACKGROUND

Without loss of generality, this work focuses on Named Data Networking (NDN), that is a concrete widely accepted ICN implementation [1]. The proposed approaches can be extended to other data-centric architectures as well.

A. NDN and publish-subscribe implementation

In NDN, a content is uniquely identified by a name and the baseline communication process follows a request-response mechanism. Only two types of messages are supported: Interest and Data packets. At the same time, each router manages three local data structures, that are Pending Interest Table (PIT), FIB, and Content Store (CS).

To retrieve a given content from the network, the consumer issues an Interest packet carrying, among the other parameters, the content name. The Interest packet is delivered according to a routing-by-name algorithm, towards the first node able to satisfy the request. When a router receives an Interest packet, it may potentially perform three different tasks. First, it verifies that the requested content is already available within the local CS. If not, the router checks the presence of a similar request (i.e., a request received from another face or another consumer, but referring to the same content name) within the PIT table. If so, the router updates the retrieved PIT entry by adding the network face from which the request was received. This operation is known as Interest aggregation. Otherwise, a new entry is created and the FIB table is examined for identifying the local face through which the Interest packet has to be forwarded. The network entity able to satisfy the request (i.e., the remote producer or an internal router with a cached content) generates a Data packet, which is sent back to the consumer according to information stored within PIT tables of network routers.

NDN can also be extended to enable publish-subscribe communication strategies [5], [7], [8], [12], [13]. Available solutions are based on pull-based and push-based mechanisms. In the pull-based approach, a sliding window mechanism controls the delivery of consumer's requests. The consumer initially sends a window of Interest packets to retrieve a certain number of consecutive content updates belonging to a given topic-name. Then, every time a new Data packet is received, the sliding window moves forward and a new Interest packet is released. Instead, the pull-based approach assumes that the consumer creates a semi-permanent communication channel with the producer, through which content updates are delivered without further solicitations. In both cases, however, mobility is not natively supported.

The works in [14] and [15] propose the Content-Oriented Pub/Sub System (COPSS) architecture, offering publish-subscribe functionalities also in mobile conditions. Nevertheless, COPSS drastically extends NDN communication primitives, by introducing two new messages (i.e. subscribe and publish), a new logical node (i.e. Rendezvous-Point), and a new table in intermediate routers (i.e. Subscription table). More conservative mechanisms, instead, are discussed in [6], where baseline pull-based and push-based approaches are extended for supporting the mobility of both consumers and producers. In this case, the key rationale is that the consumer issues a new window of Interest packets (for the pull-based case) or a new semi-permanent Interest packet (for the push-based case) every timeout set according to the producer mobility, or every time it changes network attachment point (please, see [6] for further technical details).

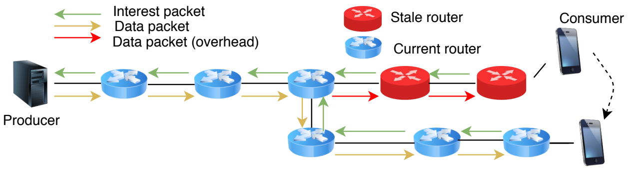


Fig. 1. Communication overhead due to the stale paths.

B. Publish-subscribe flaws due to consumer mobility

This contribution focuses the attention on the pull-based publish-subscribe mechanism presented in [6]. In this case, consumer mobility generates two important flaws. First, when a consumer changes its network attachment point, a new multi-hop communication path between the consumer itself and the remote producer is established. At the same time, however, the path established just before the handover, namely stale path, remains still active (this problem refers to both pull-based and push-based approaches). As shown in Figure 1, any new version of contents will also be delivered across stale disjointed links (that are links of the stale path not overlapped with the new path established between consumer and producer after the handover), till their expiration. The presence of wrong forwarding information in NDN routers brings to an inevitable growth of the communication overhead. Second, during the handover, the consumer misses some content updates and (if the pull-based approach is used) it is not able to correctly set the parameters of the sliding window mechanism. Therefore, when a new network access point is reached, the consumer should restore the synchronization with the producer, while optionally retrieving missed contents.

C. SDN and solutions for mobility

SDN is emerging as a key enabling technology for decoupling control and data planes, thus removing control functionalities from network devices, that now simply forward packets [16]. The Software-Defined Controller (SDC) manages network resources and runs complex applications for handling mobility, traffic engineering, security, and monitoring [16]. It can also slice network resources and run distinct virtual machines on programmable nodes by means of Network Hypervisors. This way, the SDC may deploy any desired network function (e.g., firewall, Dynamic Host Configuration Protocol server, Network Attached Storage) through virtualization on any node. The OpenFlow protocol provides a wide abstraction of network functionalities and enables the interaction between SDC and remote network elements [17]. As already discussed in [18], [19], SDN offers all the instruments for an easier integration and faster deployment of innovative data-centric architectures in the Future Internet. To address producer mobility, the works presented in [9]–[11] conceive proactive handover schemes that intend to advertise the SDC every time the producer changes its network attachment point. This way, the SDC is able to change FIB tables of network routers, thus being able to redirect pending and new Interest packets to the new producer's location. Consumer mobility, instead, has not been investigated yet.

III. THE PROPOSED APPROACHES

Figure 2 shows the reference scenario investigated in this work. It embraces NDN routers, a centralized SDC, a producer, and mobile consumers. NDN routers are access points offering wireless connectivity to both consumers and producers. Connected to each other, they make a data-centric overlay network [20]. A centralized SDC interacts with all the NDN routers through the control plane. Thus, it is assumed that it knows the topology of the data-centric overlay network, the available producers, and the position of mobile consumers. For simplicity, it is assumed that a routing protocol, like the one proposed in [21], fills FIB tables of NDN routers on demand.

The example reported in Figure 2 considers one producer, P, and two consumers, C_1 and C_2. The producer P generates contents under a specific topic-name, each one uniquely identified with an incremental identifier. Without loss of generality, the resulting content name is set to *ndn://[topic-name]/#id*. According to the pull-based publish-subscribe strategy [6], the number of Interest packets managed by the sliding-window mechanism is set to W. Let's now suppose that the consumer C_1 moves from R_1 to R_2. The handover process embraces detachment and attachment procedures. To solve publish-subscribe flaws already discussed before, these procedures are addressed differently.

A. Implementation of the detachment process

The network access point involved in the detachment process (i.e., the node R_1 in the example presented in Figure 2), sends a *handover initiation message* to the SDC. This message stores the consumer's identifier and the pending W requested content names. In turn, SDC updates its topology knowledge and starts updating forwarding information within the NDN network. This latter task is performed according to two possible protocols, namely *FIB inspection* and *neighbor inspection*.

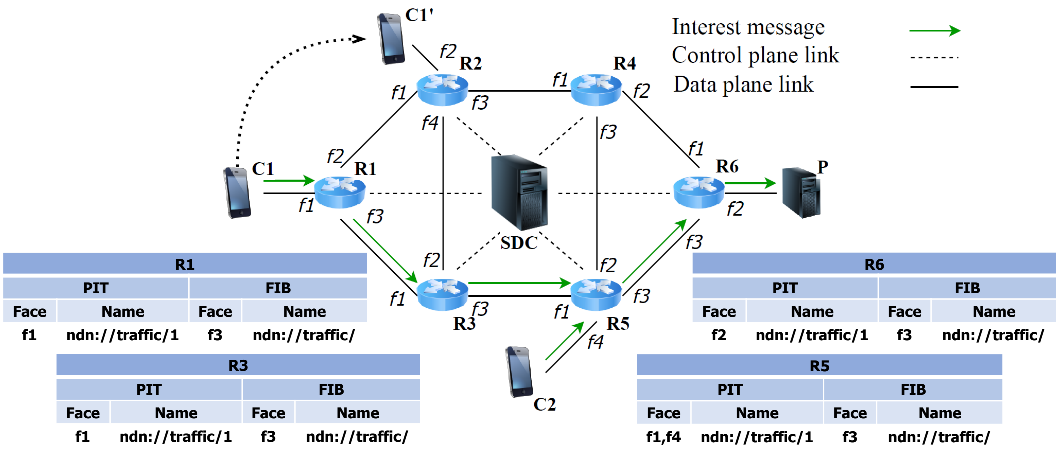
1) FIB inspection protocol: the main rationale of the FIB inspection protocol is shown in Figure 3. It assumes that SDC acquires the path established between consumer and producer before the detachment, inspects FIB and PIT tables of related NDN routers, and updates (if needed) their PIT tables by deleting wrong forwarding information associated to stale paths.

The inspection starts from the network access point that triggered the detachment processes and potentially ends when it reaches the network access point which the producer is attached to. In every round, SDC focuses on a given router of the path and retrieves entries from its PIT and FIB tables corresponding to content names stored within the handover initiation message. To this end, SDC and router exchange *FIB inspection request* and *FIB inspection response* messages. SDC immediately investigates the list of faces stored within each single PIT entry. If the investigated router is the first of the path (i.e., the one that triggered the process), SDC deletes the face associated with the mobile consumer. Otherwise, it deletes the face (if it exists) that connects the investigated router with the previous one of the path. The updated PIT entries are delivered to the router through the *FIB inspection update* message. Among the new set of PIT entries, the presence of an entry with at least one other face demonstrates that the corresponding content has been requested by another consumer, connected to the router through another path. In this case, the protocol stops looking for the corresponding content name. This way, selected stale paths previously created by the detached consumer are definitively erased. But the links belonging to other paths established with other consumers requesting the same contents through other logical faces are kept (preserving the Interest aggregation mechanism natively allowed by NDN). On the contrary, if there is an updated PIT entry with no faces, SDC extracts from the FIB entries the neighbors through which the investigated router could forward Interest packets issued by the consumer and moves inspecting their PIT and FIB tables, as just explained.

To provide further insight, Figure 2 is used to discuss a practical example. The consumer C_1 issues an Interest packet for the content *ndn://traffic/1*(the example just assumes that W=1). Then, it moves before receiving the corresponding Data packet. According to the envisaged approach, R_1 sends a handover initiation message to the SDC. Now, SDC retrieves the entries related to the *ndn://traffic/1* content from PIT and FIB tables of R_1. The iterative process realizes that C_1 is the only consumer requesting the considered content through R_1. Therefore, SDC deletes the related PIT entry in R_1, inspects the FIB entry, and moves to the next NDN router of the path, that is R_3. Similarly to what happened for R_1, also in this case SDC deletes a PIT entry in R_3 and moves inspecting R_5. Here, it recognizes that the content is also requested by C_2. Therefore, the protocol updates the PIT entry of R_5 and stops running. As a result, Data packets related to the *ndn://traffic/1* content will not be sent, through a stale path, to the location left by the consumer C_1 because of the absence of breadcrumbs in PITs. Nevertheless, Data packets will still be delivered to C_2, whose path with the producer has not been modified.

2) Neighbor inspection protocol: the main rationale of the neighbor inspection protocol is shown in Figure 3. Also in this case, the inspection starts from the network access point that triggered the detachment processes and potentially ends when the network access point which the producer is attached to is reached. But, unlike the previous case, the neighbor inspection protocol intends to scavenge forwarding information stored within all the neighbors of routers belonging to the communication path established between consumer and producer before the handover. Indeed, the information stored within the FIB table is not retrieved and used.

In every round, SDC focuses on a given router of the path and retrieves entries from its PIT tables that correspond to



R1			
PIT		FIB	
Face	Name	Face	Name
f1	ndn://traffic/1	f3	ndn://traffic/

R6			
PIT		FIB	
Face	Name	Face	Name
f2	ndn://traffic/1	f3	ndn://traffic/

R3			
PIT		FIB	
Face	Name	Face	Name
f1	ndn://traffic/1	f3	ndn://traffic/

R5			
PIT		FIB	
Face	Name	Face	Name
f1,f4	ndn://traffic/1	f3	ndn://traffic/

Fig. 2. Reference scenario.

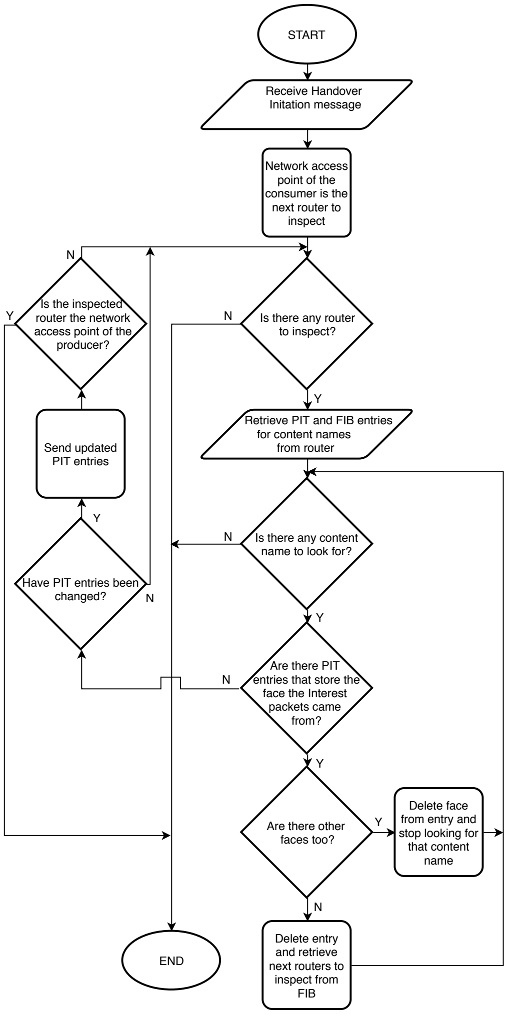


Fig. 3. FIB inspection protocol

content names stored within the handover initiation message. To this end, SDC and router exchange *neighbor inspection request* and *neighbor inspection response* messages. In line with the FIB inspection protocol, also in this case SDC investigates the list of faces stored within each single PIT entry. If the investigated router is the first of the path (i.e., the one that triggered the process), SDC deletes the face associated with the mobile consumer. Otherwise, it deletes the face (if it exists) that connects the investigated router with the previous one of the path. The updated PIT entries are delivered to the router through the *neighbor inspection update* message. Among the new set of PIT entries, the presence of a PIT entry with at least one other face demonstrates that the corresponding content has been requested by another consumer, connected to the router through another path. In this case, the protocol stops looking for that content name for the same reasons discussed before. On the contrary, if there is an updated PIT entry whose face field is empty, SDC identifies all the neighbors of the investigated router (i.e., through which it could forward Interest packets issued by the consumer) and moves to inspect their PIT tables, as just explained. To avoid redundant analysis, the router just inspected is not considered in the next rounds.

For sake of clarity, an example is commented through Figure 2. The consumer C_1 issues an Interest packet for the content *ndn://traffic/1* and moves before receiving the corresponding Data packet. R_1 sends the handover initiation message to SDC. Then, SDC retrieves the entries related to the *ndn://traffic/1* content from PIT table of R_1. The iterative process realizes that C_1 is the only consumer requesting the considered content through R_1. Therefore, SDC deletes the PIT entry in R_1 and moves inspecting PIT tables of its neighbors (that are R_2 and R_3). In the considered scenario, R_2 does not have PIT entry associated with the content name of interest. Therefore, the protocol does not produce any change to its table. On the contrary, SDC modifies the PIT entry of R_3 as well and moves inspecting its neighbors, R_5 and R_2. Here, SDC updates the PIT entry of R_5 and the protocols stops running because a PIT entry with multiple faces is found. In the end, Data packets related to the *ndn://traffic/1* content will not be sent to the location left by the consumer C_1 because

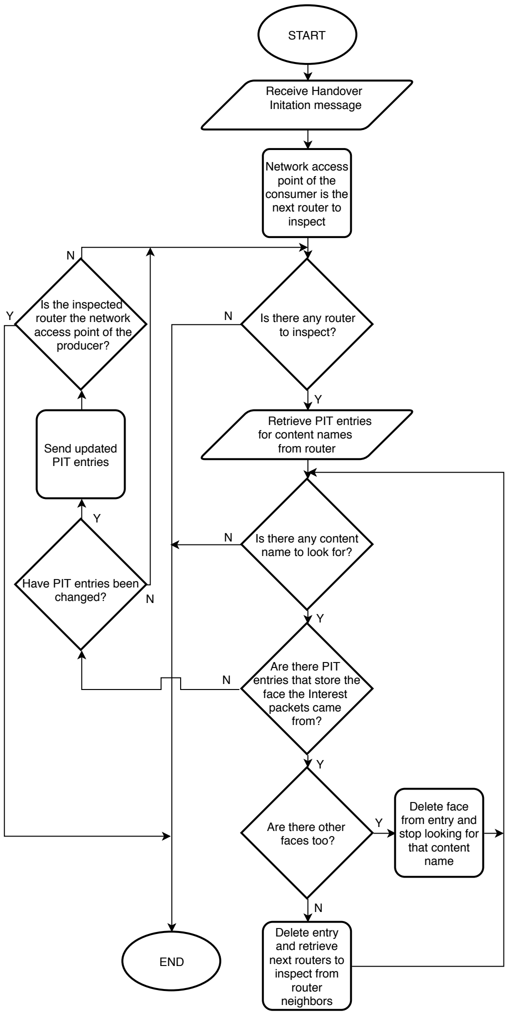


Fig. 4. Neighbor inspection protocol

of the absence of corresponding PIT entries. Nevertheless, Data packets will still be delivered to C_2, whose path with the producer has not been modified.

B. Implementation of the attachment process

The consumer triggers the attachment process when it attaches to a new network access point. Specifically, it sends a *handover completed message*, storing its identifier and the topic-name of its interest. The new network access point forwards such a message to the SDC controller, which in turns updates the network topology details in its possession. Then, the mobile consumer retrieves the identifier of the latest content generated by the producer during the handover. With this information, the consumer restores the synchronization with the producer, which (optionally) allows it to request missed contents and start retrieving new updates. To reach this goal, two protocols are conceived, that are *SDN-based re-sync* and *NDN-based re-sync*.

1) SDN-based re-sync protocol: The synchronization is fully managed by the SDC, as depicted in Figure 5. Once the SDC receives the handover completed message, it contacts the producer and retrieves the identifier of the latest generated content. Then, it sends such information back to the consumer. To this end, new messages are exchanged through the control plane: the SDC issues the *last content name request*, the producer answers with the *last content name response*, and the SDC forwards the retrieved detail to the consumer through the *last content name* message. The mobile consumer is now able to retrieve the latest content generated by the producer, the set of missed ones (if needed), and future updates. Indeed, with reference to the sliding-window mechanism, the consumer issues a window of Interest packets, whose content names are properly set based on its preferences.

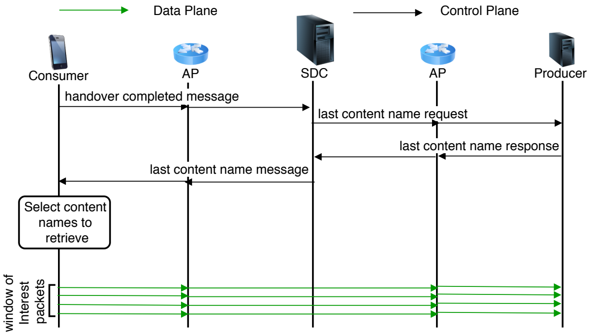


Fig. 5. SDN-based re-sync protocol

2) NDN-based re-sync protocol: The synchronization is managed by the consumer, through NDN communication primitives. Unlike the previous case, this strategy only uses the conventional NDN data plane (see Figure 6). Specifically, the consumer issues an Interest packet with a special content name set to *ndn://[topic-name]/LAST/[timestamp]*. It is important to note that the field *[timestamp]* in the name is appended to avoid retrieving cached (and, hence, not updated) responses. The request is forwarded to the producer according to the NDN forwarding mechanism. The producer generates and sends back to the consumer a Data packet containing the identifier of the latest generated content. Also in this case, the mobile consumer is now able to retrieve the latest content generated by the producer, the set of missed ones (if needed), and the future updates. Therefore, a window of Interest packets is released accordingly.

IV. PERFORMANCE ASSESSMENT

All the protocols described in Section III require the exchange of control information among network elements through control and data planes. Let Information Unit (IU) be the atomic detail carried by a generic control message. Considering that examples of IUs are content names, topic-names, PIT entries, and FIB entries, the number of IUs carried by each message is summarized in Table I.

This section presents analytical models able to evaluate the upper bound of the average number of IUs exchanged among all the network elements in a unit of time, simply referred to as *control overhead*. The terms "upper bound" mean that the

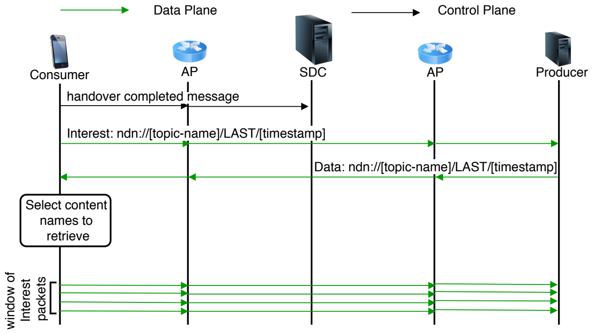


Fig. 6. NDN-based re-sync protocol

models suppose to implement each algorithm in any router of the path established, before the detachment process, between consumer and producer. To simplify the analysis, it is assumed that all the considered IUs have the same weight in the conducted study.

The control overhead also depends on the length of the path connecting any network element pair. The proposed study considers a scale-free network topology with M routers. According to [22], the number of neighbors of each NDN router, technically named *node degree*, follows a power law distribution with the exponent parameter γ set to 3. Let k_{min}, k_{max}, and $\bar{k}$ be the minimum, the maximum, and the average node degree, respectively. For simplicity, $k_{min} = 1$. In line with [22], $k_{max} = k_{min} N^{\frac{1}{\gamma - 1}}$ and the average node degree is given by $\bar{k} = \left(\sum_{k=k_{min}}^{k_{max}} k^{1-\gamma}\right) / \left(\sum_{k=k_{min}}^{k_{max}} k^{-\gamma}\right)$. The average shortest path length, i.e., $\bar{d}$, is equal to $\bar{d} = log(N)/log(log(N))$ [22], [23].

The communication path established between the consumer and the producer before the detachment process embraces the following NDN routers: $\mathcal{D} = \{r_0, r_1, r_2, ...r_{d_{old}}\}$. It embraces a number of links equal to $|\mathcal{D}| = d_{old}$. The new path established after the attachment process, instead, has a number of links equal to d_{new}. The number of faces stored within the FIB entry associated with the topic-name of interest for the $i-$th router is equal to ϵ_i.

In what follows, let $d_{c,sdc}$, $d_{p,sdc}$, $d_{c,p}$, and $d_{r_{ij},sdc}$ be the shortest path established between the access point which the consumer is attached to and the SDC, the access point which the producer is attached to and the SDC, the network attachment points of both consumer and producer, and the j-th neighbor of the i-th router of $\mathcal{D}$ and the SDC, respectively. As expected, all of these paths have the same average value, that is: $E[d_{c,sdc}] = E[d_{p,sdc}] = E[d_{c,p}], = E[d_{r_{ij},sdc}] = \bar{d}$.

In line with [6], the cell residence time Δt is defined as the amount of time in which the consumer remains connected to a given network attachment point. Its average value, that is $E[\Delta t]$, can be calculated by considering the average radius of coverage area of network access points, r, and the average consumer speed, v. Thus, it holds that $E[\Delta t] = \pi r/2v$, as already shown in [6].

The control overhead due to a specific protocol is calculated as the ratio between the total number of exchanged IUs and the residence time, as described below. The former contribution embraces the IUs exchanged hop-by-hop on all the involved links.

A. Control overhead due to the FIB inspection protocol

The number of IUs exchanged during the implementation of the FIB inspection protocol includes those carried by the handover initiation message (i.e, H_I), the set of messages exchanged between SDC and network access point (i.e. F_{req}, F_{res}, and F_{up}), and the set of messages exchanged with the ϵ_i neighbors of each $i-$th router belonging to $\mathcal{D}$, excepting $r_{d_{old}}$. Therefore, by also considering the details reported in Table I, the average control overhead due to the FIB inspection protocol, $\bar{O}_{FI}$, is equal to[1]:

$$\begin{aligned}\bar{O}_{FI} &= \frac{1}{E[\Delta t]}\Bigg(E\left[H_I d_{c,sdc}\right] + \\ &+ E\left[(F_{req} + F_{res} + F_{up}) d_{c,sdc}\right] + \\ &+ E\left[\sum_{\substack{i=0 \\ i \ni' r_i \in \mathcal{D}}}^{d_{old}-1} \sum_{j=1}^{\epsilon_i} (F_{req} + F_{res} + F_{up}) d_{r_{ij},sdc}\right]\Bigg) = \\ &= \frac{2v}{\pi r}\bar{d}\left(4W + 1 + (3W+1)\bar{d}\bar{\epsilon}\right) \ \frac{IUs}{s}. \end{aligned} \tag{1}$$

B. Control overhead due to the neighbor inspection protocol

The number of IUs exchanged during the implementation of the neighbor inspection protocol includes those carried by the handover initiation message (i.e., H_I), the messages exchanged between the SDC and the network access point of the consumer (i.e., N_{req}, N_{res}, and N_{up}), and the messages exchanged with the $k_i - 1$ neighbors of each i-th router belonging to $\mathcal{D}$, excepting $r_{d_{old}}$. Therefore, by also considering the details reported in Table I, the average control overhead due to the neighbor inspection protocol, $\bar{O}_{NI}$, is equal to:

$$\begin{aligned}\bar{O}_{NI} &= \frac{1}{E[\Delta t]}\Bigg(E\left[H_I d_{c,sdc}\right] + \\ &+ E\left[(N_{req} + N_{res} + N_{up}) d_{c,sdc}\right] + \\ &+ E\left[\sum_{\substack{i=0 \\ i \ni' r_i \in \mathcal{D}}}^{d_{old}-1} \sum_{j=1}^{k_i-1} (N_{req} + N_{res} + N_{up}) d_{r_{ij},sdc}\right]\Bigg) = \\ &= \frac{2v}{\pi r}\bar{d}(4W + 3W\bar{d}(\bar{k}-1)) \ \frac{IUs}{s}. \end{aligned} \tag{2}$$

C. Control overhead due to SDN-based re-sync protocol

With respect to the attachment process, the SDN-based re-sync protocol envisages the delivery of IUs through the handover completed message (i.e, H_C) and the set of messages managed by the SDC on the control plane to retrieve the last content name (i.e. $L_{SDN,req}$, $L_{SDN,res}$, and $L_{SDN,mes}$).

---

[1] It can be proven that $E\left[\sum_{i=1}^{d} \epsilon_i | d\right] = E\left[\sum_{i=1}^{d} \epsilon_i\right] = \sum_{i=1}^{d} E[\epsilon_i] = d\bar{\epsilon}$. Moreover, given the probability distribution of the shortest path length, i.e., $p(d)$, it holds that $E\left[d\bar{\epsilon}\right] = \sum_d d\bar{\epsilon}p(d) = \bar{\epsilon}\bar{d}$.

Therefore, the average control overhead due to SDN-based re-sync protocol, $\bar{O}_{SDN}$, is equal to:

$$\begin{aligned}\bar{O}_{SDN} &= \frac{1}{E[\Delta t]} E\big[H_C d_{c,sdc} + \\ &+ (L_{SDN,req} + L_{SDN,res}) d_{p,sdc} + \\ &+ L_{SDN,msg} d_{c,sdc}\big] = \\ &= 4\bar{d}\frac{2v}{\pi r}\ \frac{IUs}{s}. \end{aligned} \tag{3}$$

D. Control overhead due to NDN-based re-sync protocol

Regarding the NDN-based re-sync protocol, the number of exchanged IUs includes contributions from the handover completed message (i.e., H_C) and the Interest and Data packets forwarded across the routers belonging to the new path (i.e., $L_{NDN,req}$ and $L_{NDN,res}$). Accordingly, the average control overhead due to the NDN-based re-sync protocol, $\bar{O}_{NDN}$, is equal to:

$$\begin{aligned}\bar{O}_{NDN} &= \frac{1}{E[\Delta t]} E\big[H_C d_{c,sdc} + \\ &+ (L_{NDN,req} + L_{NDN,res}) d_{c,p}\big] = \\ &= 3\bar{d}\frac{2v}{\pi r}\ \frac{IUs}{s}. \end{aligned} \tag{4}$$

E. Cross comparison in different network scenarios

To practically evaluate the control overhead in different scenarios, a network with a variable number of nodes, M ranges from 10 to 10000, is considered. The window size W is set to 1 and 10. The average number of faces stored within the FIB entry, $\bar{\epsilon}$, is set to $\bar{\epsilon} = \bar{k} - 1$. Moreover, in order to consider an urban scenario the average radius of the coverage area of network access points is set to $r = 150$ m and the average consumer speed is set to $v = 3$ km/h and $v = 50$ km/h. Figures 7 and 8 show the upper bound of average control overhead evaluated in different network conditions for each combination of protocols addressing detaching and attaching processes.

Obtained results generally demonstrate that the total average control overhead increases with the number of routers in the network, the window size W, and the consumer speed. As well known, the average shortest path increases with the network size. Therefore, the higher the average shortest path, the higher the number of routers and neighbors to inspect. Regarding the sliding-window mechanism, the window size W influences the control overhead as well. In this case, the higher the window size, the higher the number of IUs exchanged during the implementation of protocols for the detachment procedure. The control overhead also grows with the average consumer speed v. This is because when a mobile consumer changes a higher number of network access points over time, it inevitably triggers many handover procedures. Accordingly, while the consumer speed increases, the amount of time protocols for both detachment and attachment processes increases too.

Regarding the combination of solutions used for both detachment and attachment mechanisms, it is evident how the adoption of neighbor inspection and NDN-based re-sync protocols achieves the best performance. This result is due to two main reasons. First, the neighbor inspection protocol requires the exchange of only PIT entries (FIB inspection protocol, instead, also involves FIB entries). Second, the NDN-based re-sync protocol envisages the interaction among only two network entities. On the contrary, the adoption of FIB inspection and SDN-based re-sync protocols always provides the highest control overhead.

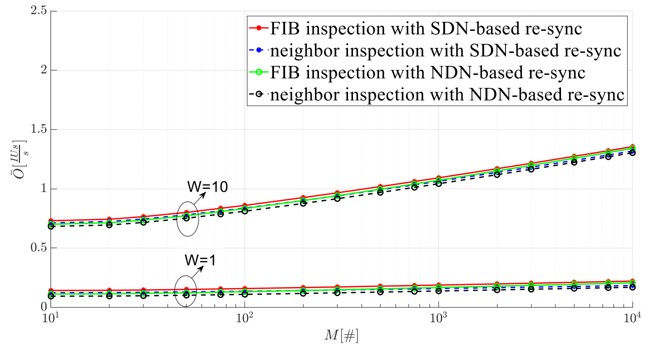


Fig. 7. Control overhead evaluated when the average consumer speed is set to $v = 3$, varying the number of routers M in the topology.

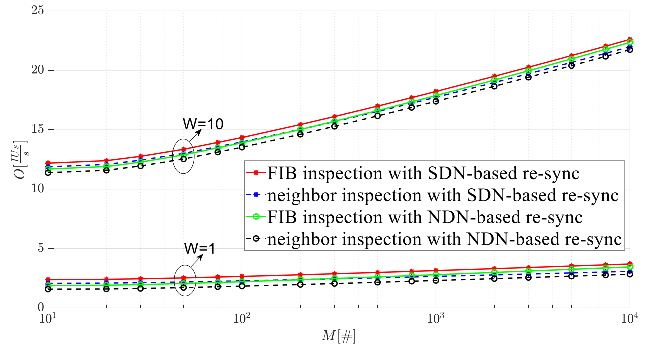


Fig. 8. Control overhead evaluated when the average consumer speed is set to $v = 50$, varying the number of routers M in the topology.

V. CONCLUSION

This work focuses on the implementation of publish-subscribe functionalities in Information-Centric Network, and formulates novel methodologies based on the Software-Defined Network paradigm, willing to solve the flaws due to consumer mobility. The handover procedure is divided into detachment and attachment processes. To address the detachment process, FIB inspection and neighbor inspection protocols have been conceived. Regarding the attachment process, SDN-based and NDN-based re-sync protocols have been designed. Moreover, to evaluate the average number of control information each single protocol exchanges per unit of time, preliminary analytical models have been developed as well. The proposed performance evaluation demonstrated that the overall average control overhead increases with the network size, the publish-subscribe window size, and the

TABLE I

NUMBER OF IUS CARRIED BY EACH MESSAGE OF THE PROPOSED PROTOCOLS

Protocol	**Message**	**Type of IU**	**Symbol and # of carried IUs**
FIB inspection	handover initiation message	content names	$H_I = W$
	FIB inspection request	content names	$F_{req} = W$
	FIB inspection response	PIT entries + FIB entry	$F_{res} = W + 1$
	FIB inspection update	PIT entries	$F_{up} = W$
Neighbor Inspection	handover initiation message	content names	$H_I = W$
	neighbor inspection request	content names	$N_{req} = W$
	neighbor inspection response	PIT entries	$N_{res} = W$
	neighbor inspection update	PIT entries	$N_{up} = W$
SDN-based re-sync	handover completed message	[topic-name]	$H_C = 1$
	last content name request	content name	$L_{SDN,req} = 1$
	last content name response	content name	$L_{SDN,res} = 1$
	last content name message	content name	$L_{SDN,mes} = 1$
NDN-based re-sync	handover completed message	[topic-name]	$H_C = 1$
	Interest packet for *ndn://[topic-name]/LAST/[timestamp]*	special content name	$L_{NDN,req} = 1$
	Data packet for *ndn://[topic-name]/LAST/[timestamp]*	special content name	$L_{NDN,res} = 1$

consumer speed. Moreover, among all the investigated solutions, the adoption of neighbor inspection and NDN-based re-sync protocols emerged as the solution reaching the best performance. Future research activities will focus on more complex scenarios embracing producer mobility and a variable number of users. Indeed, analytical models will be properly extended and validated through computer simulations.

ACKNOWLEDGMENTS

This work was partially funded by Italian MIUR PON projects Pico&Pro (ARS01_01061), AGREED (ARS01_00254), FURTHER (ARS01_01283), RAFAEL (ARS01_00305) and by Apulia Region (Italy) Research projects E-SHELF (OSW3NO1) and INTENTO (36A49H6).

REFERENCES

[1] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos *et al.*, “A survey of information-centric networking research.” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[2] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe, “A survey of mobility in information-centric networks: Challenges and research directions,” in *proc. of ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, 2012, pp. 1–6.

[3] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, “A survey of mobile information-centric networking: Research issues and challenges,” *IEEE Communications Surveys & Tutorials*, 2018.

[4] N. B. Melazzi, S. Salsano, A. Detti, G. Tropea, L. Chiariglione, A. Difino, A.-C. G. Anadiotis, A. S. Mousas, I. S. Venieris, and C. Z. Patrikakis, “Publish/subscribe over information centric networks: A standardized approach in convergence,” in *Future Network & Mobile Summit (FutureNetw)*. IEEE, 2012, pp. 1–8.

[5] A. Detti, D. Tassetto, N. B. Melazzi, and F. Fedi, “Exploiting content centric networking to develop topic-based, publish–subscribe MANET systems,” *Ad hoc networks*, vol. 24, pp. 115–133, 2015.

[6] A. V. Ventrella, G. Piro, and L. A. Grieco, “Publish-subscribe in mobile information centric networks: Modeling and performance evaluation,” *Computer Networks*, vol. 127, pp. 317–339, 2017.

[7] M. Amadeo, C. Campolo, and A. Molinaro, “Internet of things via named data networking: The support of push traffic,” in *proc. of IEEE International Conference and Workshop on the Network of the Future (NOF)*, December 2014, pp. 1–5.

[8] C. Tsilopoulos and G. Xylomenos, “Supporting diverse traffic types in information centric networks,” in *proc. of ACM SIGCOMM workshop on Information-centric networking*, August 2011, pp. 13–18.

[9] J. Ge, S. Wang, Y. Wu, H. Tang, and E. Yuepeng, “Performance improvement for source mobility in named data networking based on global–local FIB updates,” *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 670–680, 2016.

[10] L. Liu, Z. Ye, and A. Ito, “CAMS: coordinator assisted mobility support for seamless and bandwidth-efficient handover in ICN,” in *proc. of IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–7.

[11] F. Ren, Y. Qin, H. Zhou, and Y. Xu, “Mobility management scheme based on software defined controller for content-centric networking,” in *proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 193–198.

[12] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang, “Named data networking of things,” in *proc. of IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2016, pp. 117–128.

[13] J. François, T. Cholez, and T. Engel, “CCN traffic optimization for IoT,” in *proc. of IEEE Fourth International Conference on the Network of the Future (NOF)*, October 2013, pp. 1–5.

[14] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, “Copss: An efficient content oriented publish/subscribe system,” in *proc. of IEEE Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, November 2011, pp. 99–110.

[15] W. Drira and F. Filali, “A Pub/Sub extension to NDN for efficient data collection and dissemination in V2X networks,” in *proc. of IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, October 2014, pp. 1–7.

[16] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[17] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using Openflow: A survey,” *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 493–512, 2014.

[18] G. Siracusano, S. Salsano, P. L. Ventre, A. Detti, O. Rashed, and N. Blefari-Melazzi, “A framework for experimenting ICN over SDN solutions using physical and virtual testbeds,” *Computer Networks*, vol. 134, pp. 245–259, 2018.

[19] S. Rowshanrad, M. R. Parsaei, and M. Keshtgari, “Implementing NDN using SDN: A Review on Methods and Applications,” *IIUM Engineering Journal*, vol. 17, no. 2, pp. 11–20, 2016.

[20] S. Shailendra, B. Panigrahi, H. K. Rath, and A. Simha, “A novel overlay architecture for information centric networking,” in *proc. of IEEE National Conference on Communications*, 2015, pp. 1–6.

[21] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, “OSPFN: An OSPF based routing protocol for named data networking,” Technical Report NDN-0003, Tech. Rep., 2012.

[22] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[23] B. Bollobás and O. Riordan, “The diameter of a scale-free random graph,” *Combinatorica*, vol. 24, no. 1, pp. 5–34, 2004.