Computer Communications 103 (2017) 94–103

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

An autonomous and efficient controller-based routing scheme for networking Named-Data mobility

João Vitor Torres[a,*], Igor Drummond Alvarenga[a], Raouf Boutaba[b], Otto Carlos M.B. Duarte[a]

[a] Universidade Federal do Rio de Janeiro - GTA/COPPE/UFRJ, Brazil
[b] University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 27 April 2016
Revised 2 January 2017
Accepted 8 February 2017
Available online 9 February 2017

Keywords:
Named-Data
Content-Centric
Information-Centric
Networking
Software-Defined

ABSTRACT

The number of routes to store in Named-Data Networking (NDN) requires a huge amount of control messages to exchange because of the Named-Data and the non-aggregated prefixes. Multiple copies of content in different locations and content mobility worsen the scalability challenge. We propose a Controller-based Routing Scheme, named CRoS that runs on top of NDN and, thus, preserves all NDN features using the same interest and data packets. We define specific names and procedures for routers and controller efficient communication over NDN. CRoS adds router actions and avoids control message overhead by coding signaling information on content names. Our scheme enables data mobility and avoids the replications of routing information from controller to routers because they request the routes on-demand. Our proposal also requires low router memory size because it stores only the routes for simultaneously consumed prefixes. Furthermore, the scheme automates router provisioning and efficiently installs a new route, on all routers, in a path with a single Route Request to controller. We provide a protocol proposal description using the Specification and Description Language and we validate the protocol, proving that CRoS behavior is free of dead or live locks. The simulation results show that the efficiency of the proposed scheme is robust when the consumer-interest rate increases with additional throughput of more links. The scheme efficiency is close to optimum when routers operate with enough memory.

1. Introduction

Named-Data Networking (NDN) [1] focuses the content delivery instead of host-to-host communication. NDN proposes a new network layer that forwards two types of packets: the interest and the data packets. The interest packet expresses consumers will for content and leaves information on each hop to reach the consumer back. The network replies each interest packet with a data packet containing the desired content. The NDN ensures efficient communication, load balance, energy efficiency, and flow control through popular content storage and data packet replies from any content cache copy [1–3]. More importantly, interest and data packets one-to-one correspondence avoids link congestion due to Distributed Deny of Service (DDoS) attacks. Furthermore, unlike IP Multicast [4], NDN flow control is receiver-oriented and adapts to the link capacity of each individual consumer.

Named-Data identify content directly and, then, NDN must learn how to route interest packets based on the announcement of Named-Data prefixes and the diffusion of their associated data location. However, NDN routing schemes based on Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) inherit IP characteristics of prefix dissemination and routing. These schemes suffer with the amount of Named-Data prefixes that is intrinsically higher than for IP prefixes. In addition, multiple copies of content in different locations, content mobility, and multihoming introduce non-aggregated prefixes that increase the number of routes. In these scenarios, the routing schemes should store more routes and exchange more control messages to announce all the addressable content, which results in high control overhead and possible risk of Forwarding Information Base (FIB) explosion [5].

In this paper, we propose, formally specify, and validate the Controller-based Routing Scheme (CRoS) that runs on top of the NDN and, then, routers forward only NDN interest/data packets. Consequently, it preserves NDN features such as congestion control, network failure detection, and path diversity. Like OpenFlow-based solutions for Information Centric Network (ICN), CRoS consolidates the control plane on the controller, which is responsible for the Named-Data location storage and routing, but employs only NDN packets for router-controller communication. Therefore, CRoS avoids IP restrictions on host mobility and multihoming.

* Corresponding author.
E-mail addresses: jvitor@gta.ufrj.br (J.V. Torres), alvarenga@gta.ufrj.br (I.D. Alvarenga), rboutaba@uwaterloo.ca (R. Boutaba), otto@gta.ufrj.br (O.C.M.B. Duarte).

http://dx.doi.org/10.1016/j.comcom.2017.02.001
0140-3664/

We formally specify our protocol proposal in Specification and Description Language to avoid ambiguity [6]. We validate our protocol using Petri Net to prove its feasibility and correctness [7,8].

Our scheme adds router actions and avoids control message overhead by coding signaling information on content names. Additionally, the router that requests a route directly instructs the new route to routers in the path to content requested, avoiding new requests to controller. Furthermore, CRoS avoids the constant replication of routing information from controller plane to routers data plane. The CRoS router updates the forwarding plane by requesting new routes to controller upon no-response time-expiration of interests and, thus, the scheme reduces the overhead of communication between routers and controller from the large number of available prefixes to the fraction of consumer momentously requested prefixes. Moreover, CRoS reduces the router FIB memory requirement by storing only active consumed prefixes instead of all published prefixes and by replacing the oldest added routing rules with new ones.

The rest of this paper is structured as follows. In Section 2, we describe the main related work. We formally specify the proposed routing scheme in Section 3. In Section 4 we validate the CRoS NDN protocol using Petri Nets. Then, in Section 5, we discuss data delivery efficiency and convergence delay. In Section 6 we discuss the main features of our proposal. Finally, we conclude and present future work in Section 7.

2. Related work

Bari et al. analyze, compare, and contrast the naming and routing mechanisms proposed by the most prominent Information Centric Network (ICN) research projects [9]. Algren et al. compare and discuss design choices and features of proposed ICN architectures, focusing on the following main components: Named-Data objects, naming and security, Application Programming Interface, routing and transport functions, and caching [10]. Xylomenos et al. identify the core functionalities of ICN architectures, describe the key ICN proposals in a tutorial manner, highlighting the similarities and differences among them with respect to those core functionalities, identify the key weaknesses of ICN proposals and outline the main unresolved research challenges in this area of networking research [11]. All the three preceding papers pointed out scalability as challenge because of the vast size of the content naming space.

A number of schemes propose a publish-subscribe architecture to address content network [12–14]. Nevertheless, we consider that this approach is vulnerable to denial of service attacks because it does not preserve the packet flow balance. Carzaniga et al. compare NDN on-demand content retrieval to subscription approaches. They argue that both on-demand and subscription approaches should be supported and available in order to reduce states at routers. They propose a hybrid solution that selectively reduces the packet flow state at routers [14].

Jacobson et al. proposed the Content-Centric Networking (CCN) and introduced the interest-data packet pair model [15,16]. This proposal resulted in a project that aims at building a new network architecture, the Named-Data Networking (NDN) [1]. The NDN model allows fast detection of network failures and forwards packets to alternative paths according to a strategy layer [17,18]. However, current NDN routing schemes construct Named-Data forwarding rules based on conventional routing protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). Therefore, an NDN router announces name prefixes and it floods non-aggregated updates to all network nodes, imposing serious scalability limitations on the supported number of distinct prefixes and their mobility.

The centralized controller and the global view feature of Software Defined Network (SDN) technology, has been used to consolidate routing information [19–21]. Fernandes et al. observe controller-based solutions alleviate general packet forwarding nodes from control message processing and fit well for next generation networks [22]. Shi et al. propose a data synchronization scheme for NDN that can replicate the controller information [23] and provide redundancy. Salsamo et al. propose the OpenFlow-based architecture for the SDN technology applied to ICN [24], however the OpenFlow approach brings the known IP restrictions, for example, host mobility and multihoming [25].

We propose the Controller-based Routing Scheme (CRoS) that follows the Software Defined Networks (SDN) technology and preserves the same interest and data packets defined by Named-Data Networking (NDN). Thus, our proposal does not require additional packets. Moreover, the Software-Defined network approach overcomes the unnecessary control message flooding and reduces the router FIB memory. Furthermore, the on-demand Route-Request avoids the replications of routing information from controller to routers upon topology change or content mobility.

3. Proposed protocol description and formal specification

We formally present our protocol in Specification and Description Language to avoid ambiguity. Our CRoS proposal assumes two types of network elements: one controller and routers. The routers forward packets to destination, cache contents, and register the Named-Data location on behalf of producers. Moreover, routers request to the controller paths for unknown content names. The controller maintains a global view of the network, avoiding control-message flooding. Our proposal codifies the signaling information on specific data names, similarly to Jacobson et al. strategy [16]. The proposed scheme autonomously finds a path from every router to the controller. This important feature preserves the original NDN stack and automates new routers provisioning.

All routers and the controller own a unique identification (ID), and we define five specific data-name prefixes reserved for the routing scheme: i) interest `/hello` prefix followed by the router ID, which advertises the router presence to its neighbors; ii) interest `/router` prefix followed by the router ID, which addresses a specific router; iii) interest `/controller` prefix, which addresses the controller; iv) interest `/controllerx` prefix followed by the controller ID, which addresses the controller; and, finally, v) interest with `/registerNamedData` prefix, which requests the registration of new Named-Data.

Routers initiate without any forwarding rule in FIB, except the forwarding rules or procedures that the routers themselves process such as: `/hello`, `/hello/routerID`, `/controller`, and `/registerNamedData`. FIB entry `/hello` points to the router internal application that processes neighbor keep-alive messages. The FIB entries `/hello/routerID` and `/controller` point to all neighbor interfaces. The `/registerNamedData` FIB entry points to the router internal application that processes Named-Data registration requests from users.

The protocol operation is divided in two phases: Bootstrap phase, which monitors the nodes and assures the knowledge of the global network topology, and Named-Data Routing phase, which guarantees the localization and access to the requested content. This separation enables content mobility features because it splits data names from data locations.

3.1. Bootstrap phase

In Bootstrap phase, the controller obtains the global view of the network and, consequently, it can install routes on routers. Routers find the controller in order to register themselves and the controller, with the information received from all routers, constructs the global topology. Then, the controller calculates all routes. The

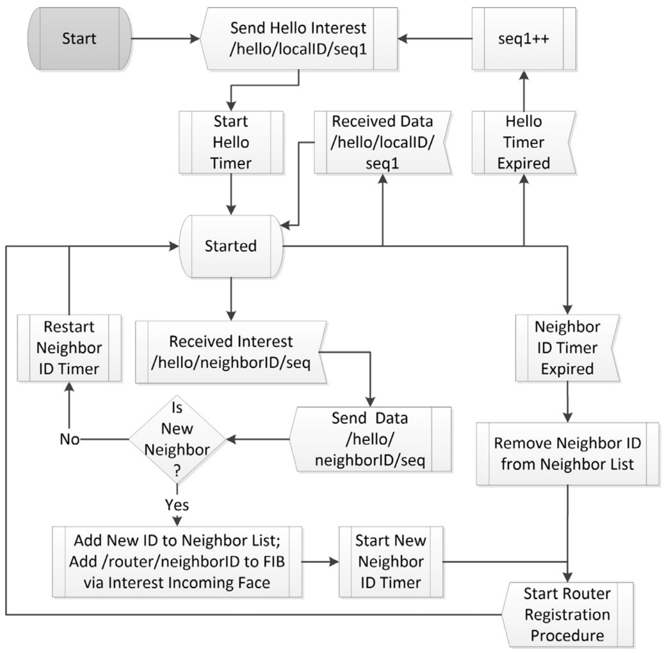


Fig. 1. Formal specification in SDL of Hello procedure.

Bootstrap phase is composed of three procedures: Hello, Controller Discovery, and Router Registration. *Hello Procedure*: All routers send a Hello interest packet to inform their directly connected neighbors about their presence. Routers keep running periodically the Hello procedure to monitor connectivity changes with their neighbors, and forward this information to the controller. Fig. 1 presents the formal specification in Specification and Description Language (SDL). Following the described behavior, each router keeps locally a restricted view of the network topology. Hello and Controller Discovery procedures start simultaneously.

Controller Discovery: Routers flood all interfaces asynchronously with interest packets to find the controller, except interfaces that received the same interest packet recently. Fig. 2 shows the SDL formal specification of the Controller Discovery procedure. The controller generates a data packet in response to the discovery interest. It is worth to notice that, as all routers subscribe to this data packet, NDN pending interest storing and NDN data caching reduces subsequent interest flooding, and the interest response time.

Router Registration: Every router register itself sending information about his presence and its neighbors. Fig. 3 presents the SDL formal specification of Router Registration procedure. Therefore, the controller constructs a global network topology model and calculates all routes between node pairs.

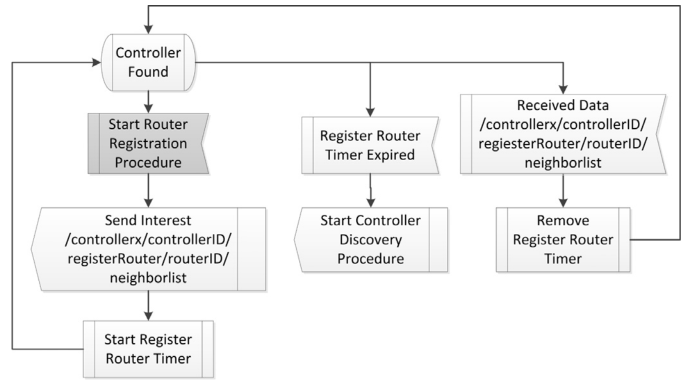


Fig. 3. Formal specification in SDL of Router Registration procedure.

3.2. Named-Data Routing phase

Named-Data are registered in the controller in order to become available to consumers.

Named-Data Registration A producer registers new content by sending a specific interest packet as shown the SDL specification of Fig. 4.

This interest packet is not forwarded to a next set of routers, instead, a newly formed interest packet is sent directly by the first receiving router to the controller. This packet signals new content availability to the controller, which reacts by storing the information for the related content prefix in the Named-Data location table, and sending content registration confirmation to the producer. *Route Request and Route Installation*: A Named Data request results from an interest packet from a node. Fig. 5 shows the SDL behavior diagram for the Route Request procedure. As a Router receives a content request, it forwards the request directly to the controller, or it replies with a NACK in case it is not the first hop. NACK messages reduce the delay to remove invalid

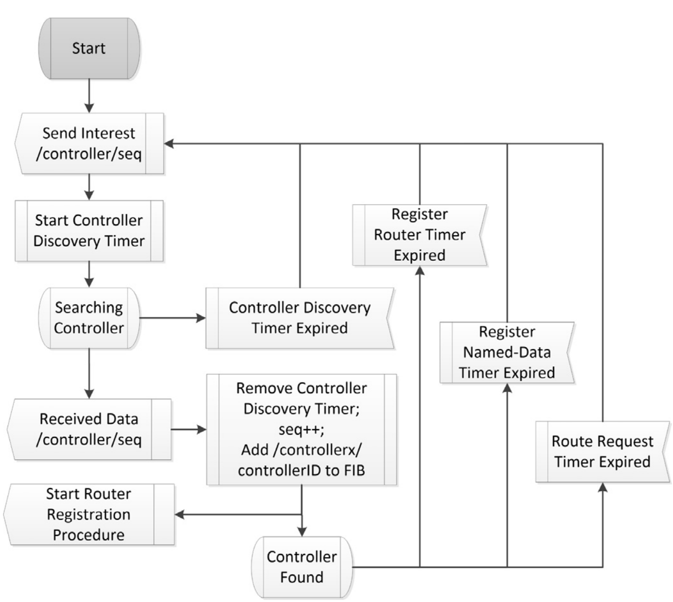


Fig. 2. Formal specification in SDL of the Controller Discovery procedure.

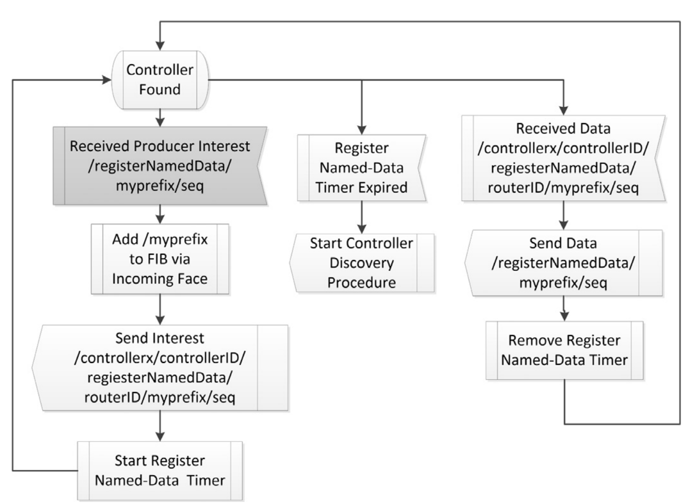


Fig. 4. Formal specification in SDL of Named-Data Registration procedure.

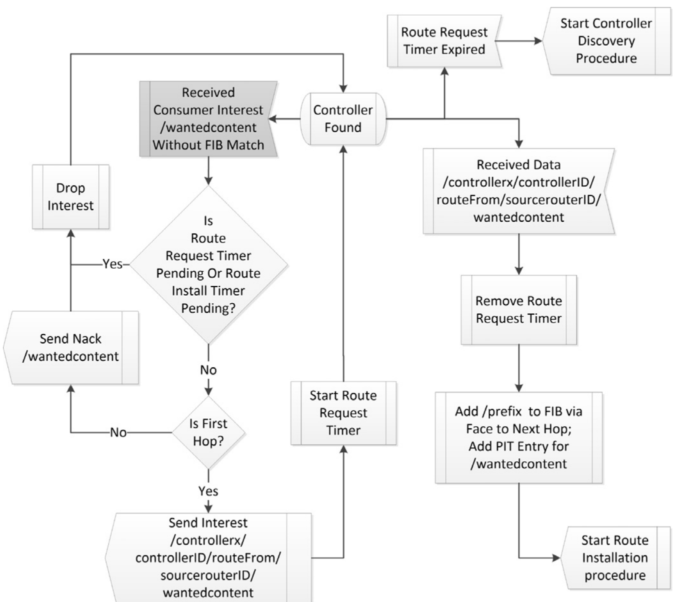


Fig. 5. Formal specification in SDL of Route Request procedure.

routes from FIB before the expiration of interest lifetime in PIT. As the controller receives content and Router Registration requests, it computes the best routes for content delivery based on the content requesting node location. Then a data packet is constructed in response to the routing request containing the full-calculated route and the content prefix. CRoS router aggregates Route Request for the same prefix and NDN caches route content, which reduce the load on the controller. This feature helps to mitigate possible attempts of denial of service attack, such as repeated requests for non registered contents. A route-install interest-packet is created by the content requesting router. Fig. 6 shows the SDL behavior diagram for the Route Installation procedure. This interest packet is responsible for carrying routing information to be installed in all routers in the contained paths, avoiding active Route Requests from these routers, as well as requesting the desired content.

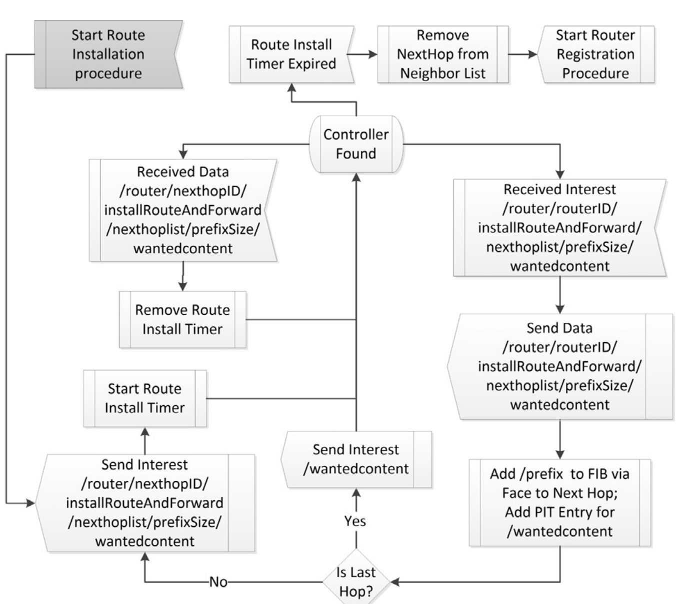


Fig. 6. Formal specification in SDL of Route Installation procedure.

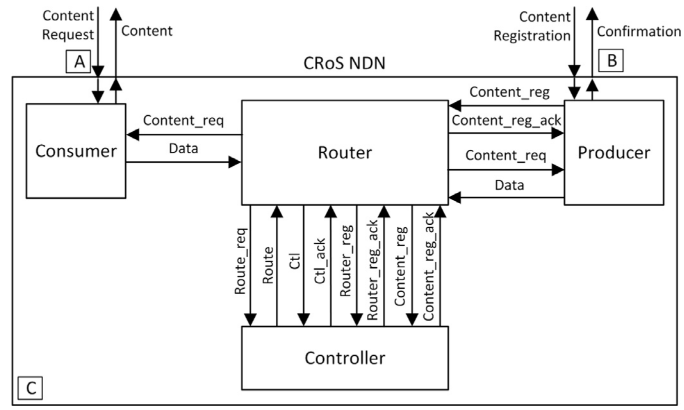


Fig. 7. CRoS NDN service diagram. User interaction with the service represented in A and external content provider interaction represented in B. All internal service interactions between service entities are represented in C.

4. CRoS-NDN protocol validation using Petri Net

The interacting entity roles involved in CRosS-NDN service delivery are: `Consumer`, `Producer`, `Controller`, and `Router`. Fig. 7 presents the service diagram of CRoS NDN, outlining the basic communication primitives between entities. In order to validate the proposed protocol, Petri Nets representing the proposed service and external interaction are constructed, as depicted in Fig. 8. The behavior associated with Hello background procedure is not depicted, but it is implied during transition TR_5 when a network change is detected and when the router first starts in state SR_1. Compared to the SDL specification, the Petri Networks focus on detailed entity interaction, but use shorter paths for sequential actions with no external interaction, and simplifies interaction with storage and software facilities. Message timeouts are handled by interest packet retransmission and were omitted for clarity. A timeout of a Route-Installation interest alerts a topology change in consumer-producer path and dispatch the Router Registration procedure faster than the periodic Hello interest.

Validation of the global Petri Network was conducted using TINA (Time Petri Net Analyzer), a software toolbox [26] for editing and analyzing Petri Nets developed and maintained by VerTICS, research groups of LAAS/CNRS. Preliminary reachability analysis indicate that CRoS NDN Petri Net is bounded, live and reversible, thus, without the presence of dead or live locks. Further structural analysis indicates the presence of transition invariants (T-invariants) and place invariants (P-invariants) that proves feasibility of the intended service using CRoS NDN proposed protocol.

A T-invariant indicates a possible loop in the net, a sequence of transitions whose net effect is null and which leads back to the marking it starts in, thus, denoting a feasible and stable path between markup states in a Petri Net. Analysis of CRoS NDN protocol equivalent Petri Network identified five T-invariants:

1. $TCtl_3$ $TPro_1$ $TPro_3$ $TPro_4$ TR_{15} TR_{16};
2. $TCtl_1$ $TCtl_2$ TR_1 TR_2 TR_3 TR_4 TR_5;
3. $TCsm_1$ $TCsm_2$ TR_6 TR_7 TR_8;
4. $TCsm_1$ $TCsm_2$ $TPro_2$ TR_{10} TR_{11} TR_{12} TR_6 TR_8 TR_9;
5. $TCsm_1$ $TCsm_2$ $TCtl_4$ $TPro_2$ TR_{10} TR_{12} TR_{13} TR_{14} TR_6 TR_8 TR_9.

The first transition-invariant denotes a stable path for content registration procedure, as well as non-conflicting content expiration handling. The control transition `Tctl3` represents content registration, while producer transitions denote a full cycle of producer operation: publish content, serve content and content expiration. `TR15` and `TR16` represent content registration router message forwarding. The second T-invariant denotes a stable path

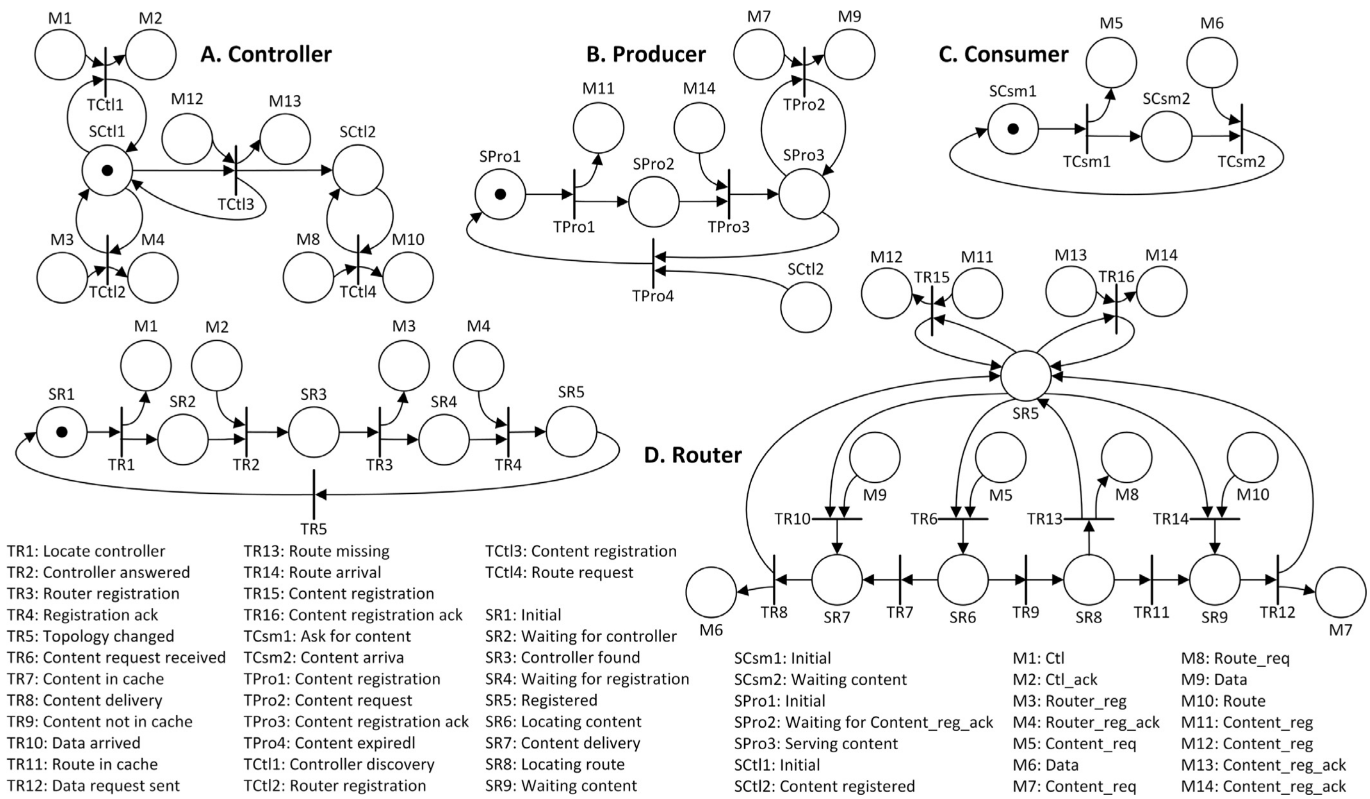


Fig. 8. Controller, Producer, Consumer, and Router entity Petri Nets. A global Petri Net is the junction, at identical named states, of presented SDL equivalent router and controller Petri Nets, with consumer and producer behavioral model Petri Nets. The initial markup used consists of one token in places SR_1, $SCtl_1$, $SCsm_1$ and $SPro_1$.

for Controller Discovery and Router Registration procedures, denoted by transitions Tctl1 and Tctl2 respectively.Transitions TR1 through TR5 indicate correct router advancement during Bootstrap phase. The last three T-invariants denote three possible stable paths for content solicitation and delivery: cached content, known route content, and unknown route content. Transitions TCsm1, TCsm2, TR6 and TR8 account for content request and delivery. The fourth T-invariant represents the case of cached content, where transition TR7 indicates cached content. The fifth T-invariant represents the case of content with a known route, denoted by the presence of TR11, this is expected if the content request packet is received from another router, thus, containing routing information. The last T-invariant accounts for content with an unknown route, denoted by the presence of TR13. Successful route retrieval triggers a Route Installation procedure, in which M7 contains routing information.

A P-invariant indicates that the number of tokens in a set of reachable markings satisfies some linear invariant. As a special case, when the sum of tokens and weight of composing places in a P-Invariant is 1, it denotes a set of places that cannot be marked concurrently, thus, proving mutual exclusion properties. Analysis of CRoS NDN protocol equivalent Petri Network identified six meaningful cases of such P-invariants:

1. $M_{11}\ M_{12}\ SCtl_2\ SPro_1$;
2. $M_1\ M_2\ M_3\ M_4\ SR_1\ SR_3\ SR_5\ SR_6\ SR_7\ SR_8\ SR_9$;
3. $M_{11}\ M_{12}\ M_{13}\ M_{14}\ SPro_1\ SPro_3$;
4. $M_{10}\ M_5\ M_6\ M_7\ M_8\ M_9\ SCsm_1\ SR_6\ SR_7\ SR_8\ SR_9$;
5. $M_3\ M_4\ SR_1\ SR_2\ SR_3\ SR_5\ SR_6\ SR_7\ SR_8\ SR_9$;
6. $M_1\ M2\ SR_1\ SR_3\ SR_4\ SR_5\ SR_6\ SR_7\ SR_8\ SR_9$.

The first place-invariant denotes that all content is available only after registration and before expiration. The second P-invariant denotes that Controller Discovery and Router Registration always occur in an orderly manner. The last four P-invariants demonstrate that there is no unnecessary message duplication for all specified CRoS protocol message pairs. Therefore, the protocol validation proves the correctness of the protocol proposal.

5. CRoS simulations and proof of properties

We have implemented our proposed scheme in the ndnSIM [27] simulator in order to demonstrate its operation, analyze its behavior, prove its properties, and evaluate its performance. Fig. 9 shows the block diagram of the CRoS router and controller implementation. CRoS controller and routers share the same structure, although their different functions in the proposed scheme. CRoS router implements a specific forwarding strategy and auxiliary applications to execute the scheme procedures that manipulate FIB and PIT employing internal calls triggered by specific data names. CRoS controller extends the NDN router with additional applications to implement the scheme procedures. CRoS router applications consume the controller-produced data on the respective controller applications. NDN simulator (ndnSIM) is the closest to reality tool and it offers a customizable forwarding strategy. Interest and data packets flow from node to node, and from/to node to/from application through faces. The strategy layer exposes customizable decisions on packet forwarding events. CRoS forwarding strategy redirects interests with unknown prefixes to the Route Request procedure.

The simulation considers that multiple collocated nodes ran the CRoS controller function as a single entity and that these nodes share a database that stores both the Named-Data location and the routers adjacency for a single domain. This assumption does not invalidate our results because it relies on data-center well-connected infrastructure to host the nodes and, therefore, it eliminates processing power and storage bottlenecks of a single node.

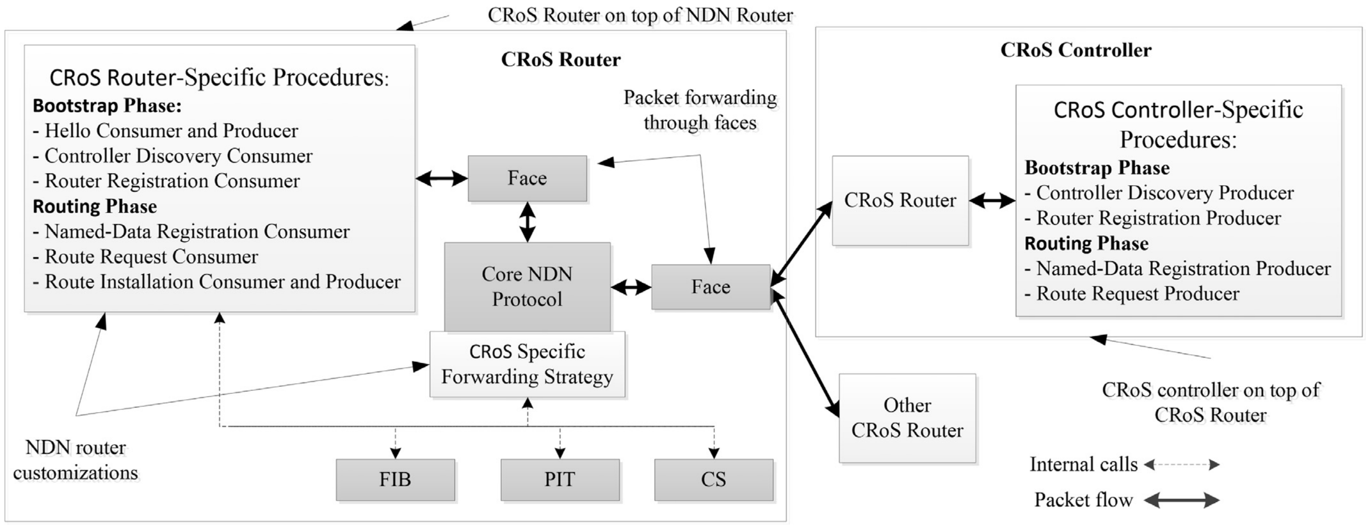


Fig. 9. CRoS-NDN Router implementation. CRoS defines a specific forwarding strategy that interacts with specific procedures to manipulate FIB and PIT entries based on specific data names. CRoS controller employs additional procedures to execute the control plane on top of CRoS Router.

Simulation results have 95% confidence interval and max and mean errors are indicated for each experiment.

In the first set of simulations, we want to show basic behaviors of the proposal. We define the data delivery efficiency, which is the amount of consumer-received data packets divided by the number of interest packets sent on each link, consumer and routing interests required to obtain the desired content. Our efficiency definition produces a measure of the CRoS signaling overhead, the consumer to content distance, and the overall cost of the data delivery. Hence, in NDN networks, the longer is the distance in node hops from the consumer to the producer; the lower is the data delivery efficiency. To show this property we use a "specific three incrementing path topology" from consumer to producer, shown in Fig. 10. The simulations evaluate the proposed scheme operation after link failures and the consequent consumer-to-producer and router-to-controller path recovery for consumer rates of 10, 100, and 1000 interests per second. Fig. 11a shows the expected decrease of the data delivery efficiency for the longer path between the producer and consumer. Fig. 11 also confirms the self-discovery of controller and self-configuration of consumer-to-producer path of our proposal. Fig. 11b presents the controller-received interest rate for Controller Discovery (top graph) and for Router Registration (bottom graph) procedures. Our protocol shows a stable overhead of control messages after a peak of control messages at the initialization and after a node failure in order to process the Controller Discovery procedure and Router Registration procedure.

B-C and F-C links fail at 1000 and 2000 s respectively, `Failure 1` and `Failure 2`, and each failure adds one hop to the consumer to producer distance.

The optimum value of the data delivery efficiency is $1/d$, where d is the consumer to producer distance. Then, 0.20 (1/5) for the five hops before link failures. The experiment demonstrates the higher is the rate of consumer interests (10, 100, and 1000 per second), the closer is the data delivery efficiency to the optimum value. Furthermore, the results confirm the data delivery efficiency (0.20, 0.17, and 0.14 for 1000 consumer interests per second) is equal to $1/d$ (5, 6, and 7 hops) for the highest considered consumer interest rate and the efficiency decreases with lower consumer rates closer to Hello rate of 0.1 interests/s .

For the same experiment, Fig. 12a presents the data delivery efficiency improvement due to the size of the FIB memory, which decreases from the number of published prefixes to the number of simultaneous consumed prefixes. CRoS router achieves FIB memory reduction because it replaces the oldest added FIB entries by new ones. The simulation demonstrates that the data delivery efficiency does not decrease with the number of published prefixes for FIB sizes higher than the number of simultaneous consumed prefixes. Moreover, the result shows that CRoS correctly operates under insufficient FIB memory for simultaneous consumed prefixes, but the data delivery efficiency decreases proportionally to the rate of Route Requests to controller, Fig. 12c. It is worth to note that when the consumer-controller path increases, then the round trip delay for Route Requests also increases and, therefore, the rate of Route Requests decreases due to interest aggregation.

Fig. 12b presents the efficient separation of data and control planes. The simulation shows that when B-C link fails at 1000 s,

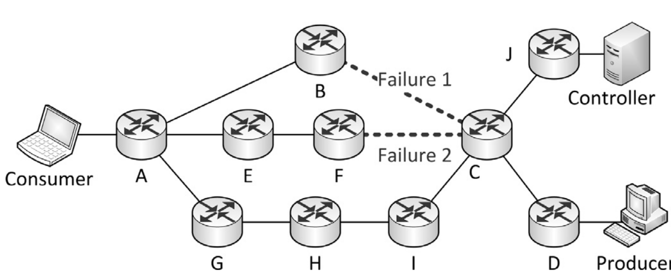
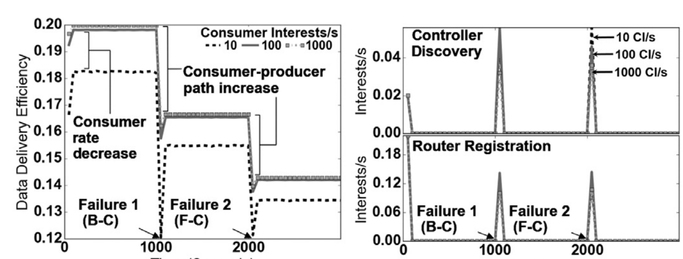
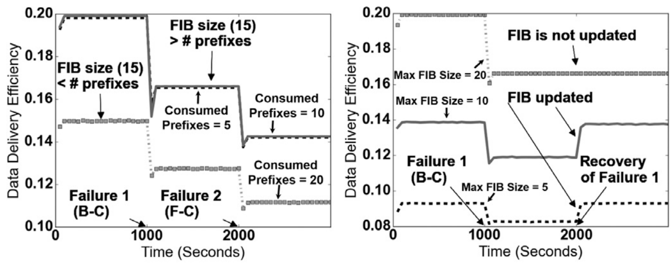


Fig. 10. The "specific incrementing path topology" used in the first set of simulations. The path distance from consumer to producer increases after failures 1 and 2: A-B-C-D, A-E-F-C-D, and A-G-H-I-C-D.



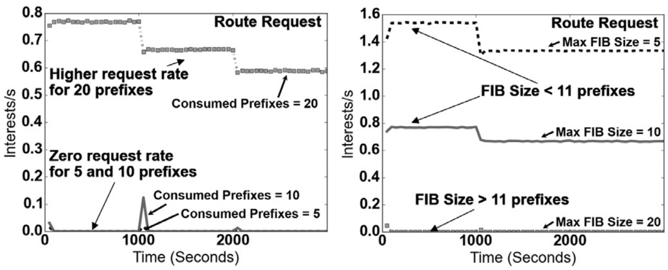
(a) Data delivery efficiency. Max/mean error: 0.0064/0.0002. (b) Controller received interests. Max/mean error: 0.0154/0.0004.

Fig. 11. *a*) The data delivery efficiency, b_{top}) the rate of interests received by the controller for the Controller Discovery procedure, and b_{bottom}) the rate of interests received by the controller for the Router Registration procedure.



(a) FIB size 15. Max/mean error: 0.0060/0.0002.

(b) Consumed prefixes 11. Max/mean error: 0.0030/0.0002.



(c) Controller received interests in a). Max/mean error: 0.084/0.002.

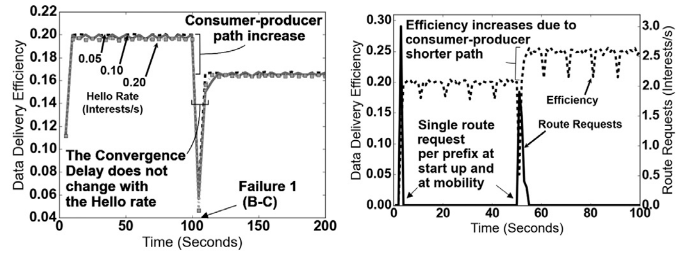
(d) Controller received interests in b). Max/mean error: 0.051/0.004.

Fig. 12. CRoS behavior for Fig. 10 topology, Link B-C failure at 1000 s, and recovery at 2000 s. a) Data delivery efficiency behavior for simultaneous consumed prefixes values of 5, 10, and 20 prefixes, with fixed FIB size equal to 15. b) Data delivery efficiency behavior in a link failure/recovery event for 5, 10, and 20 FIB sizes with fixed simultaneous consumed prefixes equal to 11. c) and d) The rate of interests received by the controller for the Route Request procedure of simulations a) and b), respectively.

the consumer to producer path increases one hop and the data delivery efficiency decreases proportionally. Afterwards, B-C link recovers at 2000 s, and then data delivery efficiency does not recover to the original value for FIB size higher than the number of simultaneous consumed prefixes. The simulation demonstrates that routers do not update their forwarding rules upon topology changes that do not break working paths. The approach avoids the proactive update of routers forwarding rules with the controller network view. Moreover, it is worth to note in Fig. 12d that the Route Request rate does not change after link recovery because the consumer-controller path is not updated.

We evaluate the convergence delay of the proposed routing scheme. The Hello interest rate defines the detection latency of a link up/down change. Thus, the higher is the Hello interest rate, the lower is the latency of link change detection. On the other hand, the interest/data balance of Route Installation procedure accelerates the detection of connectivity failure between nodes in consumer to producer path and, then, this balance removes the delay dependency on the Hello procedure.

Fig. 13a presents the data delivery efficiency and the convergence delay after a link failure for the Hello rate of 0.05, 0.10, and 0.20 interests/s. The maximum delay is 7.7, 5.1, and 5.9 s, respectively. The simulation shows the convergence delay does not change with the Hello rate due to route-install detection of connectivity failure. The data delivery efficiency does not show significant change with Hello rate in this setting. Moreover, B-C link fails at 100 s, the failure adds one hop on consumer to producer distance, and reduces 0.03 on the data delivery efficiency, from a maximum of 0.20 before `Failure 1` to 0.17 afterwards. Fig. 13b presents the Route Installation with a single Route-Request to controller per prefix, thus reducing the control message overhead. Producer node starts connected to router D and publishes 3 prefixes, them the producer moves to router F at 50 s. The simulation demonstrates that i) at start up, the controller receives 3 Route Re-



(a) Hello rate. Max/mean error: 0.038/0.002.

(b) Content mobility. Max/mean error: 0.015/0.006 (DDE), 0.77/0.02 (Route Requests).

Fig. 13. The Hello interest rate and the content mobility simulations for Fig. 10 network. a) The convergence delay in link failure event does not change with the Hello interest rate (0.05, 0.10, and 0.20 interests/s) due to route-install detection of connectivity failure. b) Route Installation with a single Route-Request to controller per prefix and the data delivery efficiency for a mobile producer with 3 prefixes; The producer starts connected to router D and moves to router F at 50 s.

quests, 1 per prefix, and ii) after producer mobility, producer registers the new data location and the controller receives 3 additional Route Requests. CRoS router reactively removes the failed routes pointing to the producer old location on PIT entry time-expiration and sends a new Route Request to controller upon a new consumer request for content. The number of Route Requests received by the controller does not change with the number of routers in consumer-producer path. Thus, for a single Route Request to controller, the controller replies with the data carrying the end-to-end path. Then, each router informs the new route to the next router on the path.

In a second set of experiments, we extend the simulation to other topologies to confirm a number of features and the robust behavior of our proposal. We use four ISP-like topologies based on the largest connected component of Rocketfuel's VSNL, Ebone, Tiscali, and Telstra topologies [28,29], which corresponds to 5, 163, 191, and 279 nodes, respectively. In addition, we evaluate CRoS with two fat-tree topologies with 25 and 122 nodes. We choose the topologies forming pairs with similar number of nodes, similar number of links, and similar mean distances to compare the effect of these parameters. We also simulate for the 10 nodes "specific incrementing path topology" of Fig. 10.

In order to evaluate the data delivery efficiency and the data delivery delay for different topologies with different network diameter, number of nodes, and number of links, we place the consumer, the producer, and the controller nodes at random positions in each simulation round. The data delivery delay is the computation of the latency from consumer request to data reception, and accounts for link related delay and CRoS timers related delay, CPU processing related delay is considered zero. The timers related delay is tied to interest lifetime expiration. The longer is the interest lifetime, the longer is the delay for PIT entry expiration and subsequent CRoS procedures execution. Fig. 14 shows the efficiency, bottom left y axis, and the mean delay, bottom right y axis, for all the seven topologies. The top graph shows the mean distance for every combination of node pairs in each network, left y axis, and the number of links of each topology, right y axis. The results confirm that the data delivery efficiency strongly depends on consumer-producer distance, being close to the inverse of the mean distance. Therefore, our results demonstrate that for a fixed consumer interest rate, the higher is the number of links, the higher is the number of Hello interests, and the higher is the control message overhead, but the data delivery efficiency shows low reduction. As worst case delay in CRoS occurs once at initial convergence, an then only after link failure, it's impact on overall performance is diluted over time for stable networks. Thus, the scheme is robust; the efficiency does not decrease when the consumer inter-

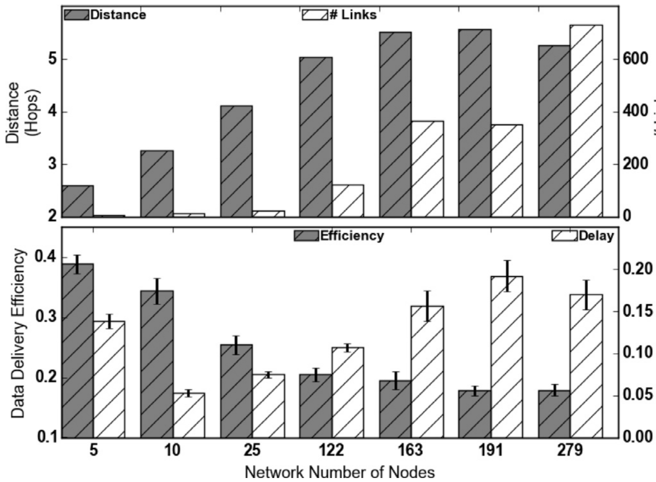
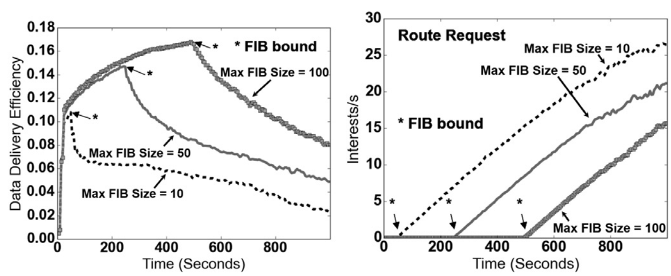
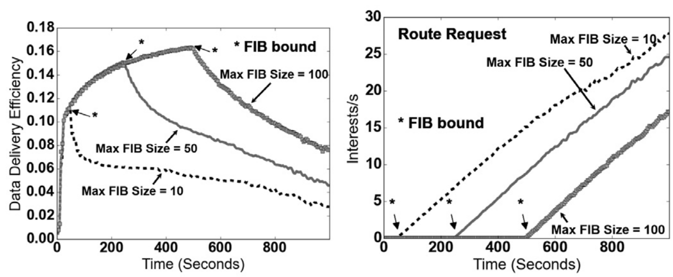


Fig. 14. Comparative simulation for distinct network topologies. The efficiency halves and the delay doubles when the consumer-producer distance doubles. The efficiency is stable when the number of links doubles.



(a) Telstra Data deliv. efficiency. Max/mean error: 0.04/0.01.
(b) Telstra Cont. receiv. interests. Max/mean error: 1.6/0.3.

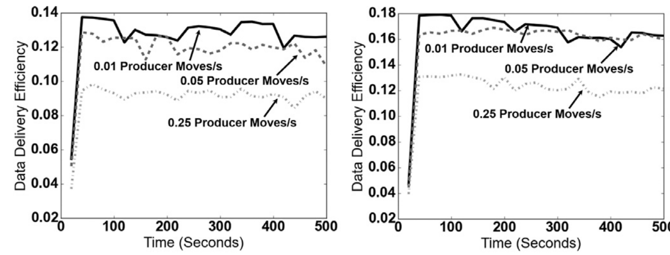


(c) Ebone Data delivery efficiency. Max/mean error: 0.03/0.01.
(d) Ebone Cont. receiv. interests. Max/mean error: 1.4/0.3.

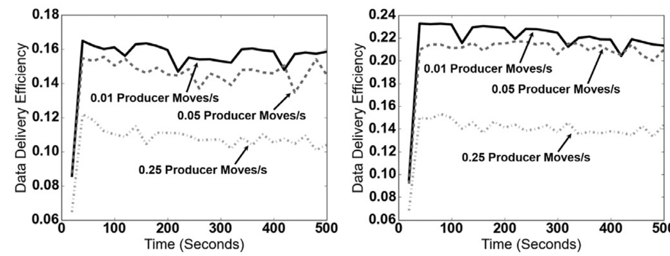
Fig. 15. a) and c) Data delivery efficiency behavior as function of the maximum FIB size. b) and d) controller received interest rate behavior as function to maximum FIB size.

est rate grows with additional throughput of more links. Therefore, the scheme scales well for a controller with enough resources. Furthermore, we envision that consumers can identify the distance to content, cache copies, and cooperatively register the copy location in the controller. Thus, the cooperation for specific contents potentially reduces the distance and the delay for new consumers and, this increases the data delivery efficiency for these contents with no cache in routers on the path to producer. This motivates an incrementally deployable approach for content producers irrespective of cache capacity in routers.

In the next simulation, Fig. 15, we want to show the implications of unbalanced FIB memory capacity, link capacity, and amount of simultaneously consumed prefixes. We observe that the efficiency decreases due to recurrent Route Requests when the number of simultaneously consumed prefixes exceeds the FIB



(a) Telstra topology, 3 consumers, and 20 interest/s per consumer. Max/mean error: 0.015/0.007.
(b) Telstra topology, 30 consum., and 2 interest/s per consumer. Max/mean error: 0.012/0.005.



(c) Ebone topology, 3 consumers, and 20 interest/s per consumer. Max/mean error: 0.018/0.010.
(d) Ebone topology, 30 consum., and 2 interest/s per consumer. Max/mean error: 0.015/0.007.

Fig. 16. Data delivery efficiency behavior of Telstra and Ebone topologies for a growing rate of producer mobility.

memory capacity. The early replacement of FIB entries causes the recurrent Route Requests in this scenario. Furthermore, the rate of Route Requests increases linearly with the rate of interests for prefixes without FIB entries up to link congestion and, afterwards, the efficiency decreases additionally due to interest retransmission caused by packet drop. It is worth to note that FIB entry removal erases all associated PIT entries and, therefore, the efficiency also decreases due to data packet drop when the FIB size is too small to store prefixes for time enough to receive producer data.

The simulations employ Telstra and Ebone topologies with 279 and 163 nodes, respectively, a growing rate of simultaneously consumed prefixes, and an increasing rate of consumer interests. A new consumer starts every 5 s and each consumer sends 1 interests/s for a distinct prefix. It is worth to note that simulated consumers do not employ flow control to adapt the interest rate to response failures and, therefore, the growing rate of interests exacerbates the efficiency decrease.

In the last simulation, Fig. 16, we show the data delivery efficiency robustness with producer mobility and with the number of consumers. The simulation employs 3 (30) consumers to request data with rate of 20 (2) interests per second in Fig. 16a and c (16 b and d), and a single producer to reply data packets. The producer moves with rates 0.01, 0.05, and 0.25 movements per second. We chose the parameters in order to exhibit a reference efficiency behavior due to 10 times variation window in both the ratio of consumer interests and the ratio of producer mobility.

6. Discussion

Our proposal preserves basic features of NDN, such as interest forwarding based on content names, and therefore preserves aggregation, caching, and mobility, being a fully NDN compliant routing solution. Our controller-based routing scheme separates data identity and location, and, thus, facilitates data mobility. Unlike current NDN routing schemes, CRoS reaches the closest registered content copy without neither incurring FIB size explosion nor supposing prefix aggregation. CRoS routers forward interest based on content names and the controller evaluates routes based on content location in network topology. Furthermore, CRoS automatically discovers/configures routers and controller and, thus, avoids manual provisioning. In addition, this automation introduces low control overhead because it restricts the interest flooding to specific name prefixes employed for routers and controller auto discovery.

CRoS avoids frequent proactive FIB-updates of routers, and, then reduces the router-controller control-message overhead. It restricts the control messages to the number of unknown prefixes of the requested data. Not all topology changes or content mobility require path updates. Actually, only the faults, identified by interest/data unbalance, that break the path from consumer to content require path updates. Whenever a path fails, the PIT entry expires, the router removes the respective FIB entry, and then the router requests a new route to the controller and updates its local forwarding information.

The data delivery efficiency of our proposal is robust in relation to producer mobility and to the number of consumers. Whenever the producer moves, it starts the Register Named-Data procedure in order to inform its new location, the Route Request procedure also starts, and consumer interests to old location expire. The higher is the rate of producer mobility; the lower is the efficiency due to the overhead of additional interest packets required for the above mentioned procedures. However, the efficiency improves with the number of consumers requesting the same content due to cache and interest aggregation.

The results obtained from an preliminary performance evaluation, comparing our proposal with other similar routing schemes [30,31], seam promissory. Nevertheless, we have to improve our simulator to work with a greater number of prefix, to achieve a valuable result.

Finally, we argue that our scheme can be integrated with the depot approach described by Zhang et al. [32], in which the depot server intermediates consumer and producer communication and keeps an updated route to producer avoiding to request new routes when producer moves. Besides, our proposed scheme can be extended with hierarchical distributed hash tables [33,34] to further improve scalability.

7. Conclusion

We proposed, formally specified, and validated the Controller-based Routing Scheme (CRoS) for Named-Data Networking (NDN). The proposed scheme natively splits content identity from content localization and, therefore, facilitates content mobility, a known challenge in traditional IP networks.

We formally specified our proposal using the Specification and Description Language (SDL) to avoid ambiguity usually caused when using a natural language. We validate our protocol using Petri Net to prove its feasibility and correctness, as well as the absence of dead or live locks. Moreover, the protocol validation guarantees that the scheme ensures a valid working path from consumer to producer, even if it does not necessarily ensure the shortest path.

We implemented CRoS using the NDN simulator (ndnSIM) to show its behavior in its two protocol phases. We conducted experiments, considering targeted scenarios, to show convergence, resilience and to demonstrate its main features.

CRoS automates the configuration of routers and controller. The autonomous discovery/configuration operate correctly at start up, after topology changes, and on data mobility. Additionally, CRoS reduces the routers-controller communication overhead by i) coding routing information on content names, ii) reactively updating the controller upon routers local information change, iii) avoiding the replications of routing information from controller to routers, iv) installing a new end-to-end route on all routers in consumer-producer path with a single Route Request to controller, and v) restricting the interest flooding.

For future work, we will extend the simulations including scenarios with other network topologies and workloads (e.g., data centers), compare with related routing approaches, and explore the cooperation of caching and routing using the controller. We will test it using CCNx distribution in Future Internet Testbed with Security (FITS).

Acknowledgment

PETROBRAS, CNPq, CAPES, and FAPERJ supported this work.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73.

[2] G. Carofiglio, V. Gehlen, D. Perino, Experimental evaluation of memory management in content-centric networking, in: IEEE International Conference on Communications (ICC), 2011, pp. 1–6.

[3] P.H.V. GuimarÃ£es, L.H.G. Ferraz, J.V. Torres, D.M.F. Mattos, A.F.M. Piedrahita, M.E.A. Lopez, I.D. Alvarenga, C.S.C. Rodrigues, O.C.M.B. Duarte, Experimenting content-centric networks in the future internet testbed environment, in: IEEE International Conference on Communications Workshops (ICC), 2013, pp. 1383–1387.

[4] L.H.M.K. Costa, S. Fdida, O.C.M.B. Duarte, Incremental service deployment using the hop-by-hop multicast routing protocol, IEEE/ACM Trans. Networking 14 (3) (2006) 543–556.

[5] D. Perino, M. Varvello, A reality check for content centric networking, in: Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, 2011, pp. 44–49. USA

[6] J.V. Torres, I.D. Alvarenga, O.C.M.B. Duarte, Proposing, specifying, and validating a controller-based routing protocol for a clean-slate named-data networking, in: Intern. Conf. on Network of the Future, 2016.

[7] K. Jensen, L.M. Kristensen, Colored petri nets: a graphical language for formal modeling and validation of concurrent systems, Commun. ACM 58 (6) (2015) 61–70.

[8] H. Macia, M.C. Ruiz, J.A. Mateo, J.L. Calleja, Petri nets-based model for the analysis of NORIA protocol, Concurrency and Comput. 27 (17) (2015) 4704–4715.

[9] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, A survey of naming and routing in information-centric networks, IEEE Commun. Mag. 50 (12) (2012) 44–53.

[10] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (7) (2012) 26–36.

[11] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, A survey of information-centric networking research, IEEE Commun. Surv. Tutorials 16 (2) (2014) 1024–1049.

[12] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 181–192.

[13] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, P. Nikander, LIPSIN: line speed publish/subscribe Inter-networking, SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 195–206.

[14] A. Carzaniga, M. Papalini, A.L. Wolf, Content-based publish/subscribe networking and information-centric networking, in: ACM SIGCOMM Workshop on Information-Centric Networking, 2011, pp. 56–61.

[15] V. Jacobson, D.K. Smetters, J.D. Thornton, M. Plass, N. Briggs, R. Braynard, Networking named content, Commun. ACM 55 (1) (2012) 117–124.

[16] V. Jacobson, D.K. Smetters, N.H. Briggs, M.F. Plass, P. Stewart, J.D. Thornton, R.L. Braynard, VoCCN: voice-over content-centric networks, in: Workshop on Re-architecting the Internet, in: ReArch '09, 2009, pp. 1–6.

[17] C. Yi, A. Afanasyev, L. Wang, B. Zhang, L. Zhang, Adaptive forwarding in named data networking, SIGCOMM Comput. Commun. Rev. 42 (3) (2012) 62–67.

[18] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, A case for stateful forwarding plane, Comput. Commun. 36 (7) (2013) 779–791.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S., J. Turner, Openflow: enabling innovation in campus networks, ACM SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.

[20] D.M.F. Mattos, N.C. Fernandes, V. da Costa, L.P. Cardoso, M.E.M. Campista, L.H.M.K. Costa, O.C.M.B. Duarte, OMNI: openflow management infrastructure, in: International Conference on the Network of the Future, 2011, pp. 52–56.

[21] I.M. Moraes, D.M.F. Mattos, L.H.G. Ferraz, M.E.M. Campista, M.G. Rubinstein, L.H.M.K. Costa, M.D. de Amorim, P.B. Velloso, O.C.M.B. Duarte, G. Pujolle, FITS: a flexible virtual network testbed architecture, Comput. Netw. 63 (2014) 221–237.

[22] N.C. Fernandes, M.D.D. Moreira, I.M. Moraes, L.H.G. Ferraz, R.S. Couto, H.E.T. Carvalho, M.E.M. Campista, L.H.M.K.C. Costa, O.C.M.B. Duarte, Virtual networks: isolation, performance, and trends, Ann. Telecommun. 66 (5–6) (2011) 339–355.

[23] W. Shi, A. Afanasyev, RepoSync: combined action-based and data-based synchronization model in named data network, in: Workshop of IEEE Conference on Computer Communications, 2015, pp. 275–280.
[24] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, L. Veltri, Information centric networking over SDN and openflow: architectural aspects and experiments on the OFELIA testbed, Comput. Netw. 57 (16) (2013).
[25] R. Atkinson, S. Bhatti, Identifier-Locator Network Protocol (ILNP) Architectural Description, RFC, RFC Editor, 2012.
[26] B. Berthomieu, P.-O. Ribet, F. Vernadat, The tool TINA – construction of abstract state spaces for petri nets and time petri nets, Int. J. Prod. Res. 42 (14) (2004) 2741–2756.
[27] A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM: NDN Simulator for NS-3, Technical Report, University of California, Los Angeles, 2012.
[28] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, L. Zhang, Interest flooding attack and countermeasures in named data networking, in: IFIP Networking Conference, 2013, 2013, pp. 1–9.
[29] N. Spring, R. Mahajan, D. Wetherall, T. Anderson, Measuring ISP topologies with rocketfuel, IEEE/ACM Trans. Netw. 12 (1) (2004) 2–16.
[30] J.V. Torres, I.D. Alvarenga, R. Boutaba, O.C.M.B. Duarte, Evaluating CRoS-NDN: A Comparative Performance Analysis of the Controller-based Routing Scheme for Named-Data Networking, Technical Report, Electrical Engineering Program (COPPE/UFRJ), 2016.
[31] J.V. Torres, An Autonomous and Efficient Controller-based Routing Scheme for Networking Named-data Mobility, Universidade Federal do Rio de Janeiro (COPPE/PEE/UFRJ), 2016 Ph.D. thesis.
[32] Y. Zhang, A. Afanasyev, L. Zhang, A survey of mobility support in named data networking, Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM'2016), 2016.
[33] C. Dannewitz, M. D'Ambrosio, V. Vercellone, Hierarchical DHT-based name resolution for information-centric networks, Comput. Commun. 36 (7) (2013) 736–749.
[34] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, S. Shenker, ROFL: routing on flat labels, SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 363–374.