Proposing, Specifying, and Validating a Controller-based Routing Protocol for a Clean-Slate Named-Data Networking

João Vitor Torres*, Igor Drummond Alvarenga*, Aloysio de Castro Pinto Pedroza* and Otto Carlos M. B. Duarte*
*Universidade Federal do Rio de Janeiro – GTA/COPPE/UFRJ
Email: {jvitor,alvarenga,aloysio,otto}@gta.ufrj.br

***Abstract*—Named-Data Networking (NDN) is the most prominent proposal for a clean-slate proposal of Future Internet. Nevertheless, NDN routing schemes present scalability concerns due to the required number of stored routes and of control messages. In this work, we present a controller-based routing protocol using a formal method to unambiguously specify, and validate to prove its correctness. Our proposal codes signaling information on content names, avoiding control message overhead, and reduces router memory requirements, storing only the routes for simultaneously consumed prefixes. Additionally, the protocol installs a new route on all routers in a path with a single route request to the controller, avoiding replication of routing information and automating router provisioning. As a result, 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. Furthermore, the protocol validation guarantees that the scheme ensures a valid working path from consumer to producer, even if it does not assure the shortest path.**

I. Introduction

Named-Data Networking (NDN) focuses the content delivery instead of host-to-host communication and this concept drastically changes networking principles [1]. This content-based network paradigm proposes a new network layer. In this network layer, unique network-visible names identify content. Unlike current TCP/IP host-to-host protocol stack, the proposed network protocol forwards two types of packets: interest and data packets. The interest packet requests content and leaves information on each hop to allow provider-to-consumer path trace back. Hence, for each interest packet, the network replies with a data packet carrying the desired content. The NDN ensures efficient communication, load balance, energy efficiency, and flow control. Additionally, interest and data packets one-to-one correspondence avoids link congestion due to Distributed Deny of Service (DDoS) attacks [1], [2].

NDN routing schemes must announce named-data prefixes and diffuse their associated data location. Current NDN routing scheme proposals, based on Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP), inherit IP characteristics due to their focus on prefix dissemination and routing [3]. Therefore, these routing schemes are not appropriate to content-based network because the number of named-data prefixes is much higher than for IP prefixes. In addition, content mobility, multihoming, and multiple copies of content in different locations worsen this scenario, introducing non-aggregated prefixes that increase the number of routes to be announced. These scenarios imply the storage of more routes and the exchange of more control messages to announce all the addressable content. Actually, the resulting scenario presents a high control-message overhead and a possible risk of Forwarding Information Base (FIB) explosion.

In this paper, we present, specify, and validate the Controller-based Routing Scheme (CRoS) that follows the Software Defined Networks (SDN) principles. The proposed routing scheme preserves the same interest and data packets defined by Named-Data Networking (NDN), as well as the same NDN processing structure detailed in [4]. Consequently, it preserves NDN features such as congestion control, network failure detection, and path diversity. 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 eliminate the IP identification/localization semantic overload that restricts host mobility and multihoming. 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.

Petri nets are an established methodology to formally validate communication and cooperation protocols, as well as distributed systems, against a proposed service offered by a set of primitives [5], [6]. We validate our protocol using Petri net to prove its feasibility and correctness. The validation procedure results prove the correctness of the protocol proposal.

The rest of this paper is structured as follows. We present and formally specify using SDL the proposed routing scheme and its properties considering a single administrative domain in Section II. In section III we validate the CRoS NDN protocol using Petri Nets. Finally, we conclude and present future work in Section IV.

II. Formal Specification of the Proposed Protocol

Our CRoS proposal codifies the signaling information on specific data names, similarly to Jacobson *et al.* strategy [7].

978–1–5090–4671–3/16/$31.00

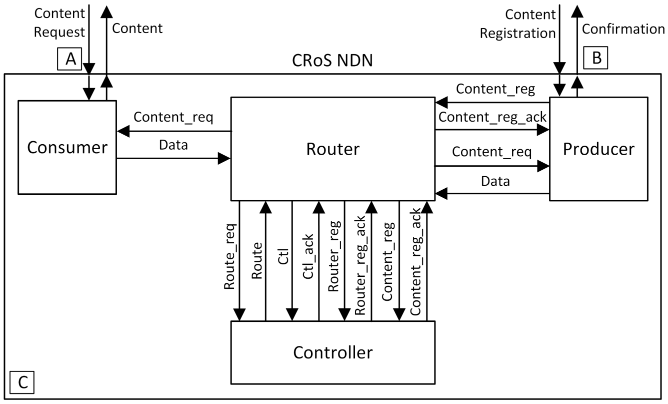

Fig. 1. CRoS NDN service diagram. User interaction with the service represented in A and external content provider interaction represented in B. All internal service interactions between service entities are represented in C.

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. We assume two network elements: one controller and routers. The routers forward packets to destination, cache content, 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 floods.

All routers and controllers own a unique identification (ID), and we define five specific data-name prefixes reserved for the routing scheme. An interest with the data name `/hello` prefix followed by the router ID advertises the router presence to its neighbors; an interest with the data name `/router` prefix followed by the router ID addresses a specific router; an interest with the data name `/controller` prefix addresses any controller; an interest with the data name `/controllerx` prefix followed by the controller ID addresses the controller; and, finally, an interest with the data name `/registerNamedData` prefix requests the registration of new named-data. The interacting entity roles involved in NDN service delivery through CRoS proposal are `Consumer`, `Producer`, `Controller`, and `Router`. Figure 1 presents the service diagram of CRoS NDN, outlining the basic communication primitives between entities.

Routers start 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. 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.

Protocol operation is divided in two phases: the bootstrap phase, which monitors the nodes and assures the knowledge of the global network topology and the named-data routing phase, which guarantees the localization and access to the requested content. This separation enables the independence of data names and data locations and, then, it enables content mobility.

A. Bootstrap Phase

The Bootstrap phase allows the controller to obtain the global view of the network and, consequently, it can install the routes, forwarding rules, on routers. Routers find the controller to register themselves, the controller acquires information to construct the global topology, and the controller calculates all routes. The Bootstrap phase is composed by 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 the Hello procedure to monitor connectivity changes with their neighbors, and forward this information to the controller. Figure 2 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: Before controller location is known, routers flood all interfaces asynchronously with interest packets in search for the controller, except interfaces that received the same interest packet recently. Figure 3 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 interest response time.

Router Registration: Figure 4 presents the SDL formal specification. After each router has informed the controller of itself and neighbors, the controller constructs a global network topology model and calculates all routes between node pairs. The route informed upon a route request can be the shortest path, but the controller may also select a slightly costlier

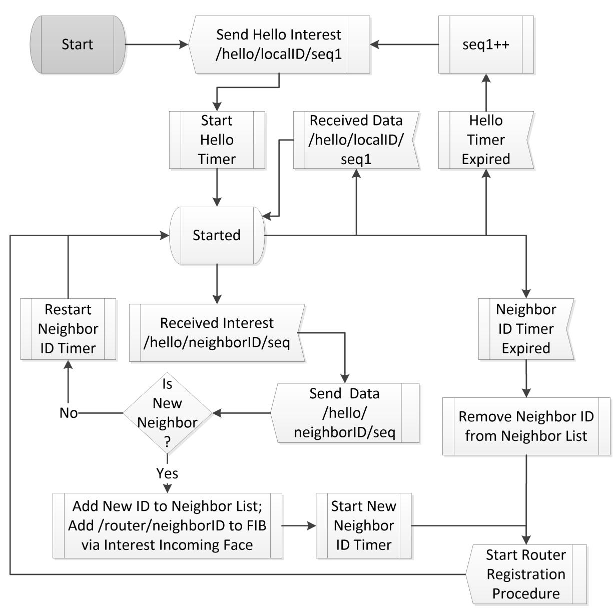

Fig. 2. Formal specification in SDL of Hello procedure.

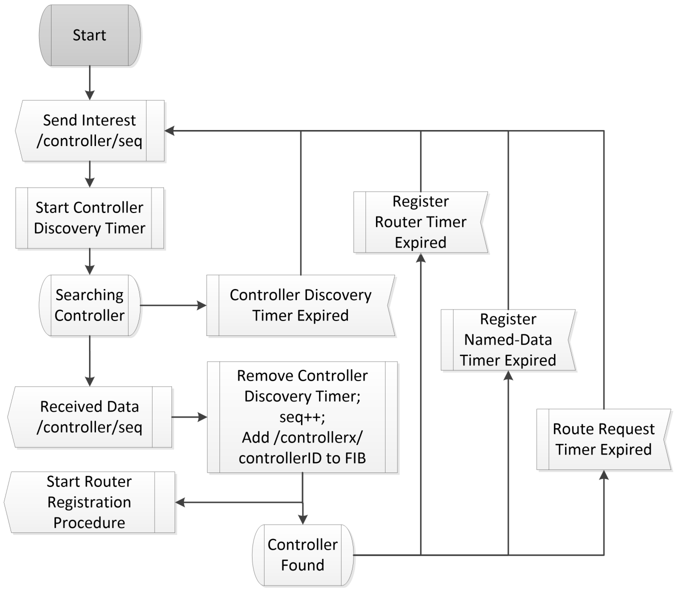

Fig. 3. Formal specification in SDL behavior of Controller Discovery procedure.

route for a node pair, in order to achieve higher efficiency from the network perspective by aiming to distribute bulk traffic and preserve overall performance [8]. This routing strategy decision affects network wide behavior, but needs to be implemented only on the controller, and can be changed on demand.

B. Named-Data Routing Phase

Named-data must be registered in order to become available to consumers. Therefore, producers must inform the availability of named-data to the controller.

Named-Data Registration A producer registers new content by sending a specific interest packet. Figure 5 presents the SDL behavior diagram. 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 registered controller of this router. 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. Even during mobility scenarios, this procedure restricts the consolidation of named-data location, to the controller, and thus, reduces overheads related to control message exchanging.

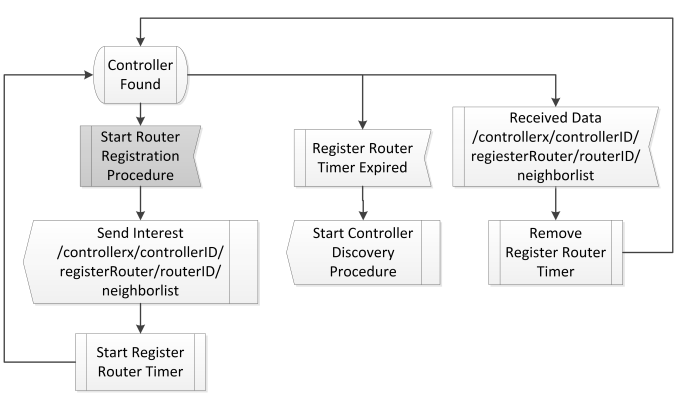

Fig. 4. Formal specification in SDL of Router Registration procedure.

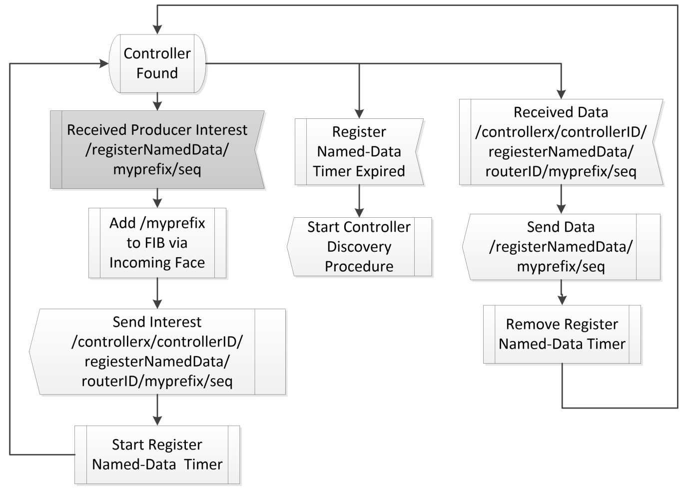

Fig. 5. SDL behavior diagram for routers execution of Named-Data Registration procedure.

Route Request and Route Installation: A named data request results from an interest packet from a node. Figure 6 shows the SDL behavior diagram for the Route Request procedure. 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.

A route-install interest packet is created by the content requesting router. Figure 7 shows the SDL behavior diagram for the Route Installation procedure. It is worth to notice that 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. This route installation triggers

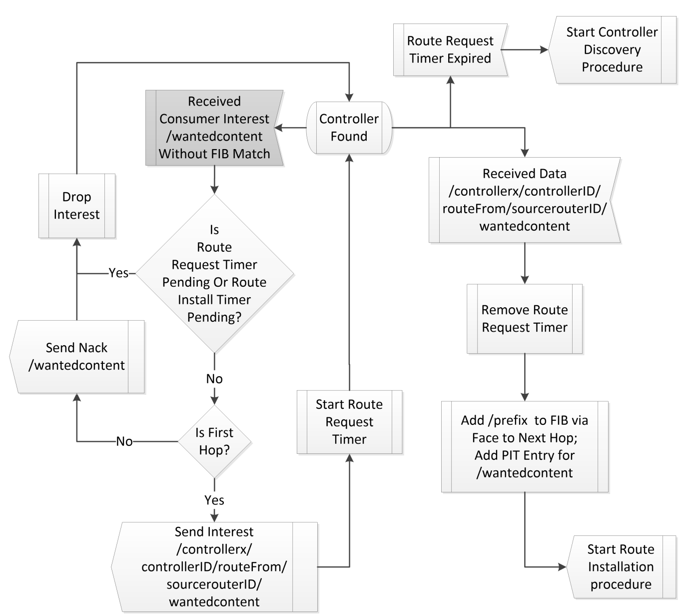

Fig. 6. SDL behavior diagram for routers execution of Route Request procedure.

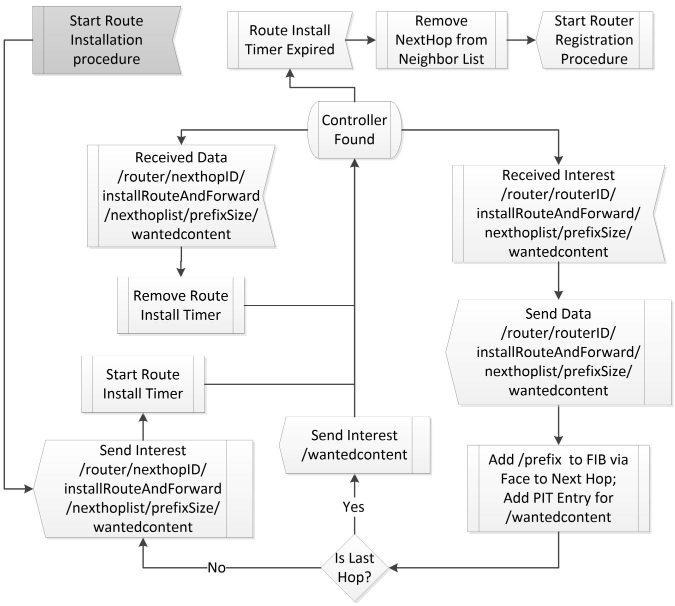

Fig. 7. SDL behavior diagram for routers execution of Route Installation procedure.

control plane updates in the aforementioned paths and results in reduction of convergence delay. Furthermore, this procedure contributes to data mobility scenarios as it updates invalid routes as needed.

III. CRoS-NDN Protocol Validation Using Petri Net

In order to validate the proposed protocol, Petri nets representing the proposed service and external interaction are constructed, as depicted in Figure 8. The behavior associated with Hello background procedure is not depicted, but it is implied during transition `TR5` when a network change is detected and when the router first starts in state `SR1`. 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) software toolbox [9]. TINA is a toolbox for the editing and analysis of Petri Nets developed and maintained by OLC, then VerTICS, research groups of LAAS/CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes/Centre National de la Recherche Scientifique). 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) TCtl3 TPro1 TPro3 TPro4 TR15 TR16;
2) TCtl1 TCtl2 TR1 TR2 TR3 TR4 TR5;
3) TCsm1 TCsm2 TR6 TR7 TR8;
4) TCsm1 TCsm2 TPro2 TR10 TR11 TR12 TR6 TR8 TR9;
5) TCsm1 TCsm2 TCtl4 TPro2 TR10 TR12 TR13 TR14 TR6 TR8 TR9.

The first T-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 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) M11 M12 SCtl2 SPro1;
2) M1 M2 M3 M4 SR1 SR3 SR5 SR6 SR7 SR8 SR9;
3) M11 M12 M13 M14 SPro1 SPro3;
4) M10 M5 M6 M7 M8 M9 SCsm1 SR6 SR7 SR8 SR9;
5) M3 M4 SR1 SR2 SR3 SR5 SR6 SR7 SR8 SR9;
6) M1 M2 SR1 SR3 SR4 SR5 SR6 SR7 SR8 SR9.

The first P-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.

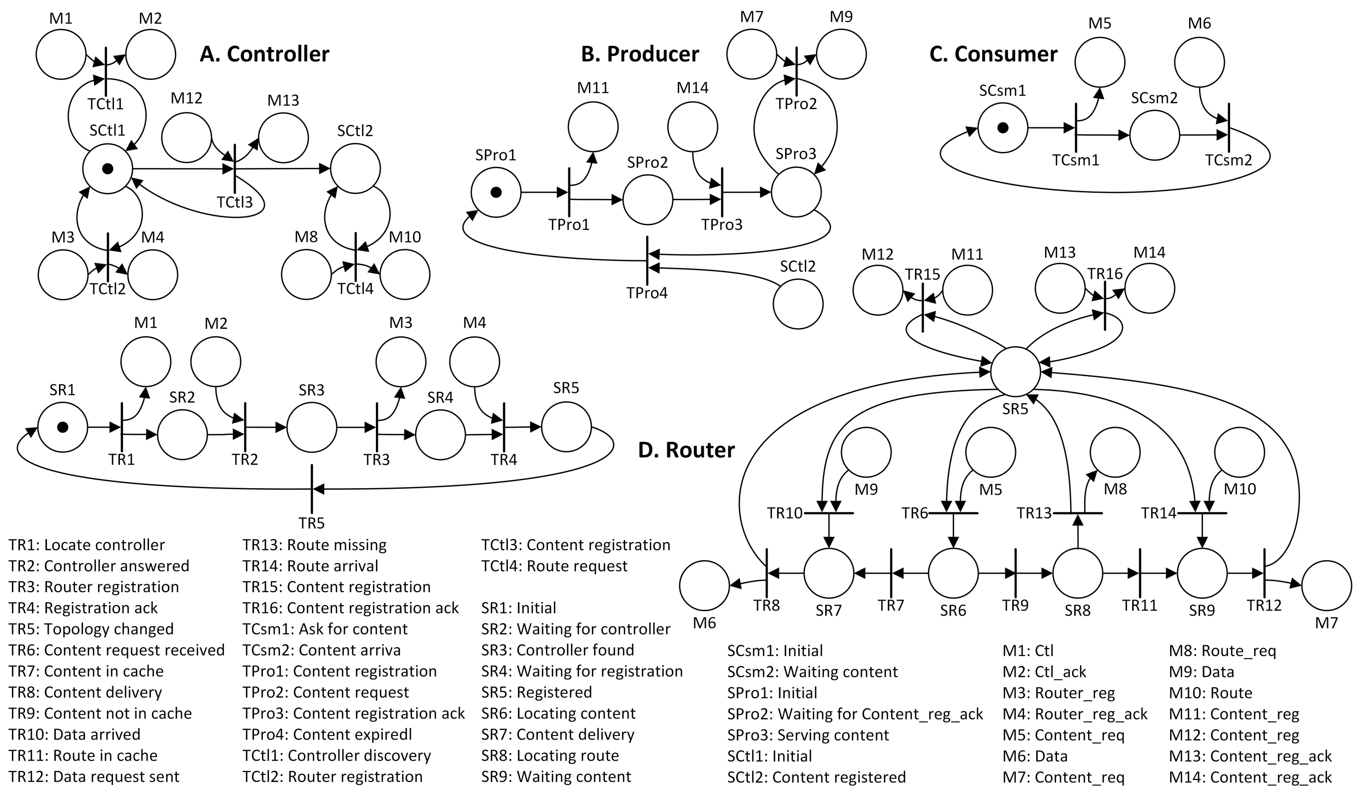

Fig. 8. Entity Petri nets. A global Petri net is obtained by the junction, at same named states, of presented SDL equivalent router and controller Petri nets, with consumer and producer behavioral model Petri nets. The initial markup used consists of one token in places `SR1`, `SCtl1`, `SCsm1` and `SPro1`.

IV. Conclusion

In this paper, we proposed, specified, and validate the controller-based routing protocol (CRoS) for Named-Data Networking. Our proposal presents several advantages in comparison with current IP-based routing approaches for Named-Data Networking because it reduces the routers-controller communication overhead by restricting the interest packet flooding. Moreover, CRoS codes routing information on content names and reactively updates the controller upon routers local information change. These features avoid replications of routing information from controller to routers and installs a new end-to-end route on all routers in consumer-producer path. One important feature provided by CROS-NDN protocol is that it does not require manual provisioning of network routers, automating router and controller configuration.

We formally specify the proposed protocol using Specification and Description Language (SDL) in order to provide a clear and unambiguous behavior of the proposal. We validate our protocol using Petri net to prove its feasibility and correctness, as well as the absence of dead or live locks. The protocol validation guarantees that the scheme ensures a valid working path from consumer to producer, even if it does not assure the shortest path.

For future work, we will evaluate and compare the performance by simulations with the current IP-based approaches.

Acknowledgment

Work supported by PETROBRAS, CNPq, CAPES, and FAPERJ.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[2] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental Evaluation of Memory Management in Content-Centric Networking," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–6.

[3] D. Perino and M. Varvello, "A Reality Check for Content Centric Networking," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN'11. New York, NY, USA: ACM, 2011, pp. 44–49.

[4] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," *Comput. Commun.*, vol. 36, no. 7, pp. 779–791, Apr. 2013.

[5] K. Jensen and L. M. Kristensen, "Colored petri nets: A graphical language for formal modeling and validation of concurrent systems," *Commun. ACM*, vol. 58, no. 6, pp. 61–70, May 2015. [Online]. Available: http://doi.acm.org/10.1145/2663340

[6] H. Macia, M. C. Ruiz, J. A. Mateo, and J. L. Calleja, "Petri nets-based model for the analysis of noria protocol," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 4704–4715, 2015, cpe.3452. [Online]. Available: http://dx.doi.org/10.1002/cpe.3452

[7] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-over Content-centric Networks," in *Proceedings of the 2009 Workshop on Re-architecting the Internet*, ser. ReArch '09. New York, NY, USA: ACM, 2009, pp. 1–6.

[8] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul, "SPAIN: COTS data-center Ethernet for multipathing over arbitrary topologies," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, ser. NSDI'10, 2010, pp. 18–18.

[9] B. Berthomieu, P.-O. Ribet, and F. Vernadat, "The tool tina – Construction of abstract state spaces for petri nets and time petri nets," *International Journal of Production Research*, vol. 42, no. 14, pp. 2741–2756, 2004. [Online]. Available: http://dx.doi.org/10.1080/00207540412331312688