Chapter 2
Information-Centric Networks (ICN)

Muhammad Azfar Yaqub, Syed Hassan Ahmed, Safdar Hussain Bouk and Dongkyun Kim

Abstract During the past decades, serious efforts have been made to propose various architectures for the future Internet. Each of those architectures has one thing in common, i.e., to focus on content delivery rather than on host-centric approaches. However, only few of them gained popularity due to their possible applications being investigated. In this chapter, we describe the overview of various future Internet architectures such as data-oriented networking architecture, content-centric networking, named-data networking, publish/subscribe, and network of information. The main objective of this chapter is to allow our readers to become familiar with the transformation of these architectures.

Keywords ICN · DONA · CCN · NDN · Pub/sub · Net-Inf

2.1 Information-Centric Network (ICN)

2.1.1 Brief History

The core idea behind information-centric networking (ICN) architectures is that who is communicating is less significant than what data are required. This paradigm shift has occurred due to end-users' use of today's Internet, which is more content-centric than location-centric, e.g., file sharing, social networking, or retrieval of aggregated data. The ICN concept was initially proposed in TRIAD [1], which proposed name-based information communication. Since then, researchers have proposed multiple architectures (Fig. 2.1). In 2006, the data-oriented network architecture (DONA) project [2] at UC Berkeley proposed an ICN architecture, which improved the security and architecture of TRIAD. The Publish Subscribe Internet Technology (PURSUIT) [3] project, a continuation of the Publish Subscribe Internet Routing Paradigm (PSIRP) [4] project, both funded by the EU Framework 7 Program (FP7), have proposed a publish/subscribe protocol stack that replaces the IP protocol stack. In another approach, the Network of Information (NetInf) project [5] was initially proposed by the European FP7 4WARD [6]

S.H. Ahmed et al., *Content-Centric Networks*, SpringerBriefs in Electrical and Computer Engineering, DOI 10.1007/978-981-10-0066-9_2

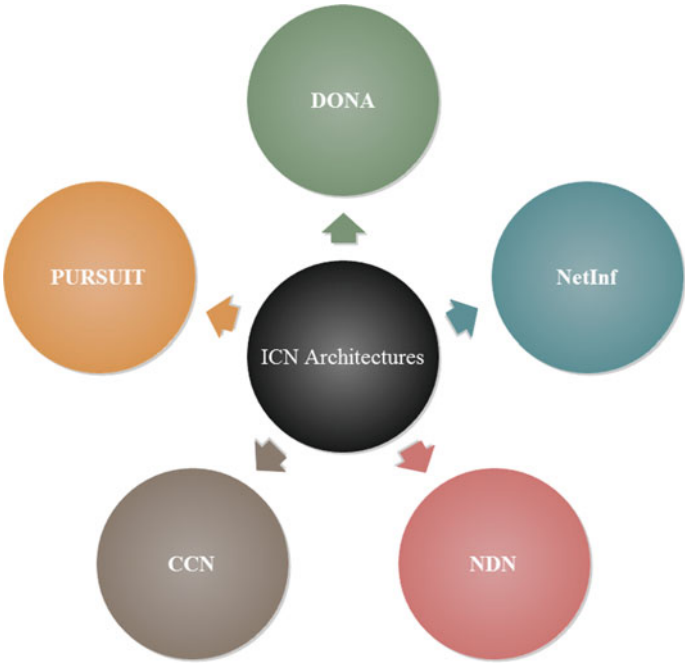

Fig. 2.1 ICN architectures

project, and further development has been made by the Scalable and Adaptive Internet Solutions (SAIL) [7] project. Similarly, Van Jacobson, a Research Fellow at PARC, proposed the Content Centric Networking (CCN) project [8] in 2007. Currently work is being performed to enhance the CCN architecture called "named-data networks" (NDN) [9].

All of these approaches differ in terms of implementation, but they have the same goal, i.e., to improve the performance and end-user experience of the Internet by providing access to content and services by name rather than by original location. This is achieved by changing the concept of link protection to content protection and by exploiting in-network storage of content. In addition, traditional networks also benefit from ICN technologies, i.e., content delivery networks, which aim to distribute content efficiently and swiftly.

2.1.2 ICN Core Architectures

In this section, we give an introduction and overview of four ICN approaches in networking, namely, CCN/NDN, DONA, NetInf, and PURSUIT. A higher level description of these architectures is illustrated to provide a general understanding of the readers

2.1.2.1 CCN/NDN

Introduction

The content-centric network (CCN) architecture was originally proposed by Van Jacobson as a project initiated by Palo Alto Research Center (PARC). Before publishing [8] the architecture, Jacobson first introduced it in his Google Tech Talk [10]. The main idea behind CCN is that the request (Interest) packets containing the desired content/data name are broadcast by a consumer node, and routing protocols are employed to distribute information about the location of content based on the name using the longest prefix matching. Routing aggregation is leveraged through a hierarchical naming scheme. The content provider, or any other network node with a copy of the requested content, routes the required content, along with additional authentication and data-integrity information, along the interests reverse route. Furthermore, caching on each path node is enabled depending on the caching policy of the node. An overview of the communication is illustrated in Fig. 2.4. In an intermittent-connectivity scenario, this can speed up content retrieval because the content is replicated in the network [8]. Further details—such as naming, security, caching, name resolution and routing, transport, and mobility of the CCN architecture—are discussed in detail in the next chapter.

Named-data networking (NDN) is an enhanced version of the CCN architecture. Similar to CCN, NDN also follows the interest/data packet combination to obtain any particular data. There are, however, some architecture differences incorporated into NDN, which reduces the interest/data search time as well as the interest looping issue. Figures 2.2 and 2.3 illustrate the difference in both CCN and NDN basic operations (Fig. 2.4).

Naming

NDN adopts a hierarchical naming scheme, e.g., information may have the name/work/class101/presentation.pdf, where the sign "/" shows the hierarchy of the name component. The relationships and context of the data elements are easily represented in this hierarchical structure. In a typical CCN, each node consists of three data structures: a pending interest table (PIT), a content store (CS), and a forwarding information base (FIB). PIT contains a list of pending and satisfied interests. The entries include a content name, the interest-incoming interface, a NONCE value to identify the individual interest packet, and timers for PIT-entry management. CS provides a cache to store the content available at the node and content received from other nodes based on the caching policy of the node. FIB helps in routing the incoming interest to the next hop toward the content provider; it maintains name prefixes and outgoing interfaces for interest packets. In addition, a forwarding algorithm is used to provide a forwarding strategy, which uses these data structures.

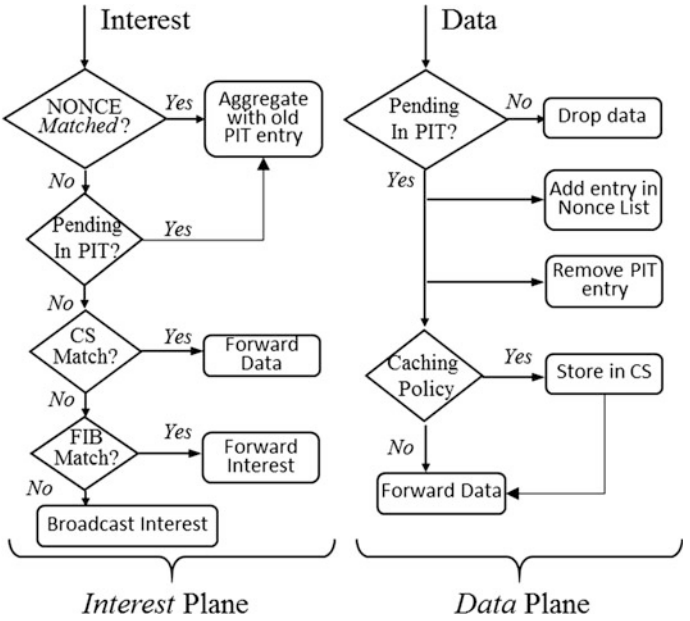

Fig. 2.2 CCN

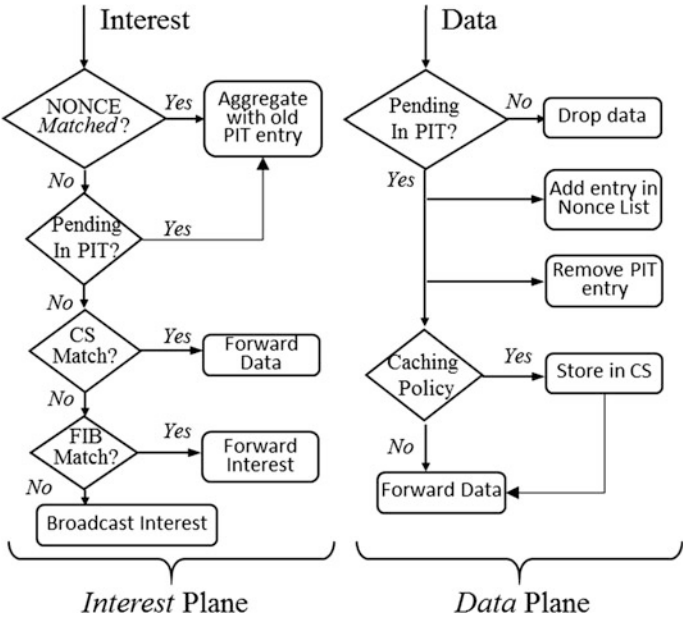

Fig. 2.3 NDN

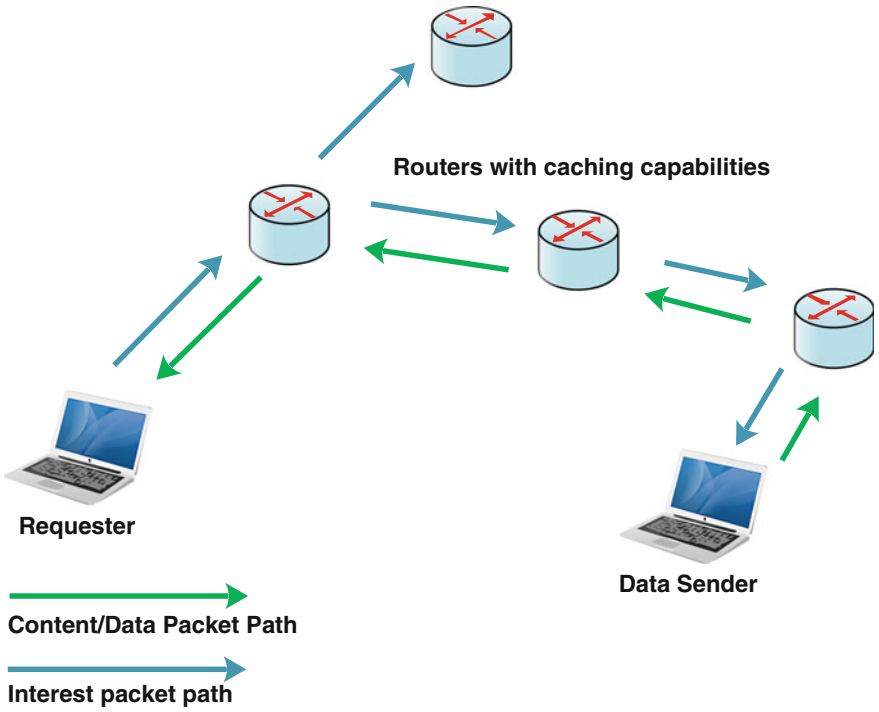

Fig. 2.4 CCN/NDN overview

Security

In NDN, the content publisher provides security by cryptographically signing each data packet [8]. Hierarchical namespace is used to achieve better routing scalability. To achieve data integrity, every content is signed with the publisher's secret key, but the trust in the signing key must be established through some external means. The naming in CCN/NDN typically does not contain the publisher's key (PK). Although this helps with the human readability of the names, self-certification is not possible. Multiple methods are used to verify the key, such as information through a friend, direct information, information through a trusted third party, or information through a global PKI.

Routing and Name resolution

In CCN/NDN, name-based routing is used to forward packets between the source and destination. The client/requester broadcasts interest packets for the required content in the network. This interest is forwarded to the name-prefix of the destination using longest-prefix matching at the FIB of each intermediate node.

Each incoming interest's information is stored in the PIT; furthermore, multiple requests of the same content are aggregated together. When the requested content copy is found at a node, a data packet is sent back with the requested data on the reverse path toward the client. In addition, each node along the data path can cache a copy of the data.

Caching

Caching is one of the major advantages of the ICN architectures [11]. Caching the content in the CS of a node is analogous to the buffer memory in IP routers; however, the IP routers cannot reuse the data packet after forwarding. However, in NDN the storage of packets is possible at each NDN node, thus allowing the node to satisfy any future request for the particular data. In addition, as the content name does not contain any information of the user, thus making the users more secure.

Transport

The CCN/NDN architecture does not provide any transport-layer functionality. The transport-layer functionalities are provided by the application or some supporting libraries as well as the forwarding algorithm used in NDN architecture. The hierarchical namespace allows the information required for transport to be included in the content name, thus eliminating the need for transport-layer information such as sequence and port numbers. The application itself monitors the state of each outstanding interest in the PIT. After a certain timeout, retransmission is initiated. To limit congestion in the network, each interest packet has a limited lifetime; furthermore, caching the data packets at each node mitigates any congestion losses in the network because retransmitted interest packet will be satisfied by the node with the particular data packet in its cache.

2.1.2.2 DONA

Introduction

DONA's architecture involves a redesigning of the current Internet naming, i.e., DNS names are replaced with flat, self-certifying names, and DNS name resolution is replaced with any cast name resolution process. Furthermore, these changes are incorporated above the IP layer, thus leveraging the lower layers of path discovery mechanisms. The architecture provides improved data retrieval as well as improved service by providing persistence, authentication [12, 13], and availability.

In DONA, the source/content provider is responsible for publishing the content in the network. To serve data, the nodes must authorize with the resolution infrastructure. A route-by-name paradigm is used for name resolution. Now, instead

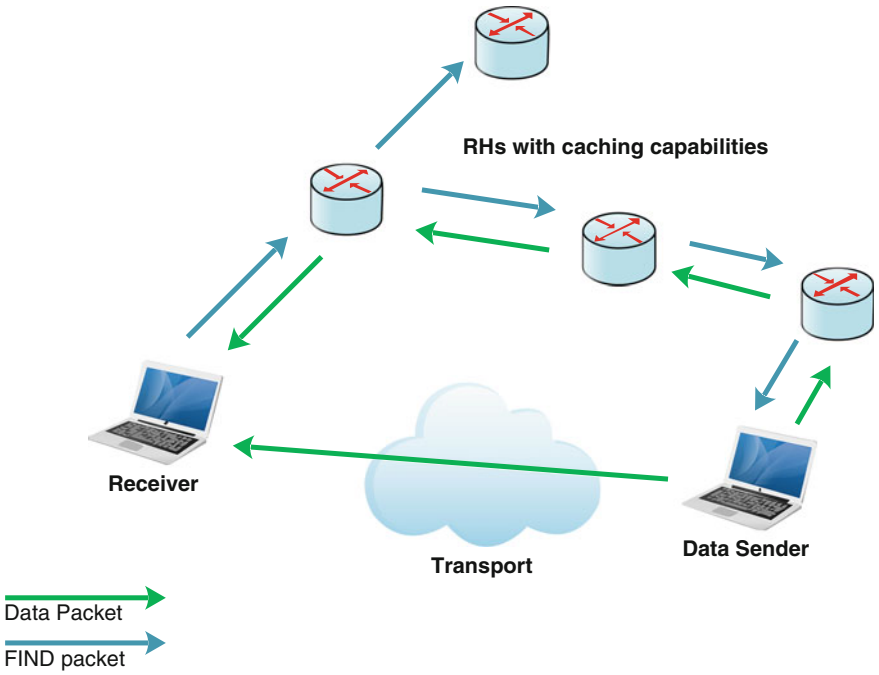

Fig. 2.5 DONA overview

of using DNS servers, DONA relies on the network entities called "resolution handlers" (RHs). The request (FIND) packets are forwarded through multiple RHs toward the node with a copy of the content as illustrated in Fig. 2.5. The content/data can be acquired through two methods: (1) it is sent back through the same path the interest packet came in on with caching enabled on each encountered RH or (2) it can be sent back directly toward the consumer. The source also has the option to register their principals with the RH so that the request packets can be sent to them directly. However, the registrations must be renewed periodically. RH routes requests using a hierarchical approach to find the closest content provider. The any cast name resolution used in DONA provides support for the network middle boxes (e.g., firewalls, net address translators, proxies, etc.) by providing a separate mechanism for path discovery.

Naming

In DONA, a flat namespace is used, and the names are organized by using principals. A public private key pair is used to associate each principal, which is in the form P:L where P and L are the globally unique principal fields containing the cryptographic hash of the publisher's public key and the object label, respectively.

Each datum received contains metadata with a minimum of three fields, i.e., data, public key, and signature. To ensure data integrity, the requesting client relies on the principals' signatures. In addition, because P is unique for each publisher, republishing the same content by a different publisher will result in multiple copies of the same content. These multiple copies can either be removed by using various methods, i.e., wildcard queries or principal delegation or be used to satisfy multiple content requests [14].

Security

The self-certifying namespace used in DONA attributes to name-data integrity. This removes the necessity for PKIs. This is achieved by adding the cryptographic hash function of the content in the object label "L." In the event of dynamic data, the signature of the contents hash is added in the metadata of the content; furthermore, the public key corresponding to the hash in the ID's authenticator field is used to sign the metadata. This allows the object label to be securely bound with the data and also provides information to handle the data yet to arrive/exist. In contrast, using self-certifying names provides a trade-off between the human readability of the content names and name-data integrity. However, self-certifying names eliminates the need for a PKI by allowing the comparison of the receiving data identity with the one sent in the data request, consecutively making the security process simple and reliable (offline data verification).

DONA architecture relies on the IP-level mechanisms and contractual limits set by the providers to restrain the unwanted packets from overwhelming the server, the client, or the RH.

Routing and Name Resolution

As discussed previously, a name-based routing paradigm is used for name resolution in DONA. This is achieved through the network entities called "resolution handlers" (RHs) instead of DNA servers. There is at least one RH available in each domain. Each node in the domain has information on the RH through local configuration information. The client and RH use two primitives, FIND (P:L) and REGISTER (P:L), to achieve name resolution. The nodes willing to serve data use the REGISTER (P:L) packet to register the datum with the RH. Each RH maintains a registration table, which is used to map the incoming requests to the destination of the content, copy, or to the next hop RH toward the content copy. The RHs are organized in a hierarchical manner.

In the event that the RH receives a new REGISTER (P:L) packet from a child node, it is stored in the registration table, and the RH also forwards the packet to its parent and peers. The peer receiving the REGISTER will forward the packet on the basis of the local policy. The REGISTER packet is not forwarded onward if a

record already exists or if the new REGISTER comes from a copy further away from the previous copy.

The client issues a FIND (P:L) packet to locate the content named "P:L." If there is an entry in the registration table when the request is received at the RH, the request is forwarded to the next hop RH; otherwise the RH forwards the request to its parent RH. In case of more than one entry being in the registration table, the closest one is selected. Once a content copy is found, the data are either routed back to the client by way of the reverse request path with caching enabled on the RHs or forwarded directly to the client as shown in Fig. 2.5. In DONA, name matching is accomplished using longest-prefix matching [15].

Caching

In DONA, RHs can be enabled with a universal-caching mechanism. To enable caching, the RH populates its cache by replacing the source IP address and port number of the FIND packet with its own and then forwarding it to the next-hop RH, thus, thus ensuring the traversal of the response packet through this RH. The RH stores the data in its cache before forwarding it to the requesting node. All the data items in the cache are labeled with a TTL or some other validation metadata, which ensures the time period of the data.

When a FIND arrives and there is a cache match, the RH will initiate the appropriate transport response, which will lead to the standard application-level exchange of data. In case the transport or application-level protocol information in the FIND is ambiguous to the RH, then it does not provide any caching for that particular request.

Transport

The DONA architecture relies on the existing transport protocols, i.e., TCP, to provide the forwarding mechanisms and other transport functionalities such as flow control, congestion control, and reliability.

2.1.2.3 NetInf

Introduction

Like the DONA architecture, NetInf also uses flat namespace [16]; hence, a public key infrastructure (PKI) is not required. NetInf's content model is based on the widely used multipurpose internet mail extensions (MIME) standard. Furthermore, search primitives, which provide links between the search item and the object name, are also a part of the architecture. Two objects-retrieval approaches are offered in the architecture, namely, name resolution and name-based routing. Depending on

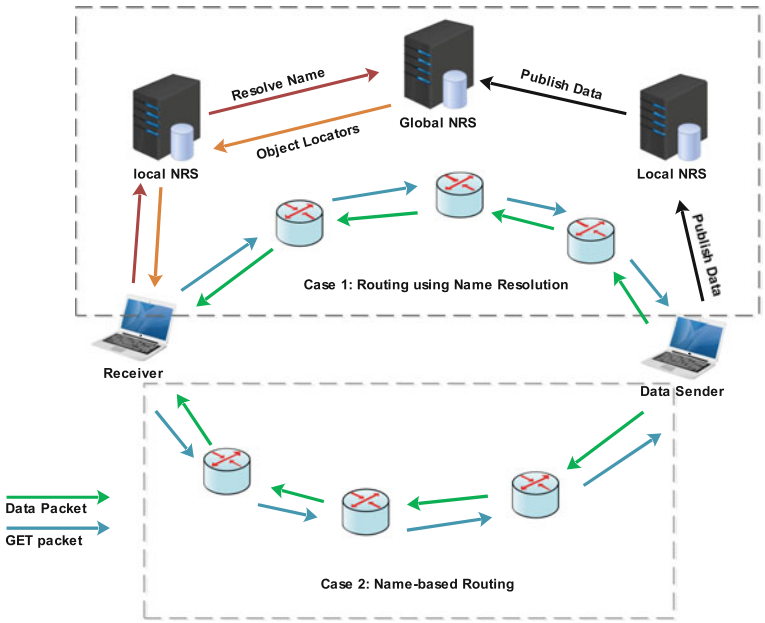

Fig. 2.6 NetInf overview

the model used, the source/NetInf node can either register with the name resolution service (NRS) to publish the content (termed "named-data objects" [NDOs]) or use a routing protocol to announce the routing information.

As illustrated in Fig. 2.6, in case no. 1, the client first forwards the request to the NRS, which gives the available locaters of the particular NDO name; subsequently, the client retrieves a copy of the data from the best available sources. Alternatively, using name-based routing, the client can directly send out an NDO GET request, which is forwarded to the source. The data are sent to the client as soon as a copy of the NDO is reached. These can either be used separately in the network or merged into a hybrid scheme, in which case switching between the two schemes is performed on hop-by-hop basis. This hybrid scheme allows NetInf to adapt and scale itself to the different requirements in the network such as network mobility [17], delay-tolerant networking (DTN), and global connectivity. Furthermore, NetInf architecture can be deployed as an extra layer on top of the existing network infrastructure, thus simplifying the migration of applications to the new infrastructure.

Naming

NetInf uses a flat namespace with a structure similar to the DONA namespace. NetInf aims to accommodate different ICN architectures by deploying the naming

such that the naming aims to differentiate three aspects: (1) the common naming format used by each node; (2) security related information to maintain the integrity of the data; and (3) name-object binding validation mechanisms. The NetInf naming format supports different hashing schemes. The owners public key hash digests is also contained in the name to support the data that is yet to arrive. Furthermore, different naming representation is supported, i.e., Uniform Resource Identifier (URI) and binary representation.

Security

NetInf uses the self-certifying namespace, which provides object security service for static as well as dynamic objects. The aforementioned naming format and the object model enable data-integrity validation by the nodes. Like DONA, NetInf validation of the named data can be achieved without the PKI infrastructure. In addition, object security is provided through public key cryptography, the pseudonym of the owner, and identification.

Name resolution

To incorporate the different ICN architectures, NetInf supports a variety of name-resolution services. NetInf merges name resolution and name-based routing to retrieve the data. A new interdomain interface is defined for name resolution and routing, which allows different schemes to be applied in multiple parts of the network. Today's URLs are supported by NetInf name resolution; hence, NetInf can be integrated smoothly with the current infrastructure. Multiple types of NRS are supported in NetInf such as both local and global NRS. Using local NRS, the operators can reduce and control the traffic flow thus potentially decreasing the load on caches and servers.

A number of name-resolution mechanisms have been developed such as multilevel distributed hash table (MDHT) [18], hierarchical skipnet (HSkip) system [19] and late-locator construction (LLC) [20]. MDHT and HSkip systems provide a global and hierarchical NRS that is topologically embedded in the core network to improve stability, scalability, copy-locator selection, and efficient data dissemination. LLC focuses on high-dynamic network topology handling, which includes movable networks. This NRS approach allows a smooth transition from the Internet while using the current infrastructure. For example, traditional URLs can be resolved from object names and retrieved using the existing HTTP protocol.

Caching

Caching plays an important role in efficient content distribution in NetInf. NetInf supports three caching options: on-path caching, off-path caching, and peer caching.

The NetInf router has a built-in caching feature to enable on-path cache, which caches objects while routing objects in response to the GET request. Off-path cache is placed in the network to reduce the traffic and latency. This cache is not directly in the request/data path. It is typically connected to an NRS in the network. The cache broadcasts the cached objects to the NRS and, based on the popularity, the NRS informs which data to cache. In this way, the off-path cache can avoid the steps to obtain the information from GET requests of the requested objects. In peer-caching, the NetInf nodes can function as an on/off-path cache. The peers can broadcast the cached data in the network. The NRS can route the GET request, thus reducing interdomain traffic and latency and additionally minimizing the load on the data servers.

Transport

In NetInf, different forwarding mechanisms are used to retrieve a data object, locator, or redirection hints using request/response messages. This communication is managed by convergence layers (CLs), which provide a concrete abstraction of the NetInf protocol from the lower layers. Thus, by ensuring smooth NetInf implementation across technologies. CL ensures that communication between the NetInf nodes is achieved in a hop-by-hop manner. An example of the CL system is given in [21]. The CL implements a specific transport protocol, which manages the resources required for sharing and reliability of corresponding network paths.

2.1.2.4 PURSUIT

Introduction/Model

The Publish-Subscribe Internet Technology (PURSUIT) project was previously known as the Publish-Subscribe Internet Routing Paradigm (PSIRP) [22]. In PURSUIT, sources publish the contents into the network as shown in Fig. 2.7. The receivers can subscribe to the published contents through the rendezvous systems. A rendezvous system helps in locating the scope and publications in the network. Each piece of the published content belongs to a specific named scope. The subscription requests contain the scope identifier (SI) and the rendezvous identifier (RI), which together identify/name the particular desired content. Using these identifiers in a matching procedure results in a forwarding identifier (FI), which is used by the source to forward the data. A bloom filter [23] is specified in the FI, which is used by the intermediate routers to select the interfaces to forward contents as shown in Fig. 2.7. This relieves the router from maintaining the forwarding states. However, a bloom filter yields some false-positive results, thus leading to forwarding on interfaces where there are no receivers.

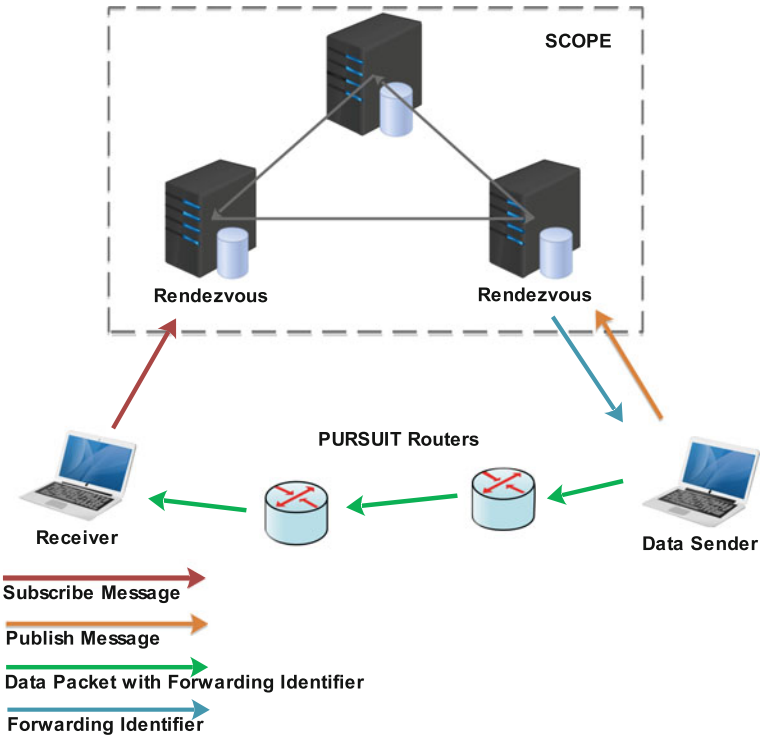

Fig. 2.7 PURSUIT overview

Naming

PURSUIT uses a flat namespace with two types of names, namely, RI and SI. These identifiers together establish the name of the content. RIs help in mapping the content between publishers and subscribers. In addition, the forwarding identifier (FI) is used by routers to identify the path from the publisher to the subscribers.

Security

Security is an integral aspect of PURSUIT design employed to avoid inherent security glitches in the network. PURSUIT uses self-certifying names, which alleviate the need for a PKI; therefore, nodes can easily check the name-data integrity based on the received data's name. The other security aims are to avoid unwanted traffic on both the rendezvous and forwarding layers; furthermore, policy should be enforced such that only valid subscribers can obtain the data. PURSUIT makes use of elliptic-curve cryptography (ECC) [24] for signature verification and packet-level authentication (PLA) [25] to provide network layer confidentiality, authenticity, and accountability of the data.

Routing and Name Resolution

The routing management is responsible for selecting the best interdomain-route forwarding of the publications. PURSUIT uses the resolution model called a "rendezvous point." The name resolution of the data is performed at this point. However, the data return path to the subscriber does not have to include the rendezvous point. Forwarding is performed using the source routing approach using bloom filters called "zfilters" carried in the FI. The bloom filter describes the route from the source to the destination because it contains all the names of the routing links. This information is attached to the data as the FI. At each node, the router checks whether or not the link identifier is present in the packet using a simple AND operation. Thus, in PURSUIT, increasing packet length as well as the network resources are reduced.

Cache

Caching in PURSUIT is mainly provided as a dedicated solution to a problem for which caching might offer some benefit. Moreover, multiple caches of an object can be maintained based on the scope of the rendezvous point for the identifier associated with the object.

Transport

PURSUIT's basic forwarding process is based on the bloom filters as mentioned previously. Each object has a unique algorithmically derived name from the original name, which helps to handle the flow control. Alternatively, subscribers can add flow-control feedback in an algorithmically derived name to which the source can subscribe.

In addition to the other initiatives taken by the research community for future Internet architectures, the CCN got much attention recently. Therefore, we discuss CCN in detail in the following chapters and will also provide research challenges for this subject.

References

1. Cheriton D, Gritter (2000) Triad: a new next-generation internet architecture
2. Koponen T, Chawla M, Chun B-G, Ermolinskiy A, Kim KH, Shenker S, Stoica I (2007) A data-oriented (and beyond) network architecture. SIGCOMM Comput Commun Rev 37 (4):181–192
3. FP7 PURSUIT project (Online). Available: http://www.fp7-pursuit.eu/PursuitWeb/
4. FP7 PSIRP project (Online). Available: http://www.psirp.org/

5. Dannewitz C, Kutscher D, Ohlman B, Farrell S, Ahlgren B, Karl H (2013) Network of information (netinf)–an information-centric networking architecture. Comput Commun 36 (7):721–735
6. FP7 4WARD project (Online). Available: http://www.4ward-project.eu/
7. FP7 SAIL project. [Online]. Available: http://www.sail-project.eu/
8. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pp 1–12, New York, NY, USA, ACM
9. Zhang L, Afanasyev A, Burke J, Jacobson V, Claffy K, Crowley P, Papadopoulos C, Wang L, Zhang B (2014) Named data networking. SIGCOMM Comput Commun Rev 44(3):66–73
10. Jacobson V (2006) A new way to look at networking, Google Tech Talk, Aug 2006
11. Fayazbakhsh SK, Lin Y, Tootoonchian A, Ghodsi A, Koponen T, Maggs B, Ng K, Sekar V, Shenker S (2013). Less pain, most of the gain: incrementally deployable ICN. SIGCOMM Comput Commun Rev 43(4)
12. Mazi`eres D, Kaminsky M, Kaashoek M.F, Witchel E (1999) Separating key management from file system security. In: Proceedings of SOSP '99, pp 124–139, Charleston, SC, USA, Dec 1999
13. Moskowitz R, Nikander P (2006) Host Identity protocol architecture. RFC 4423, IETF, May 2006
14. Xylomenos G, Ververidis CN, Siris VA, Fotiou N, Tsilopoulos C, Vasilakos X, Katsaros KV, Polyzos GC (2014) A survey of information-centric networking research. Commun Surv Tutorials IEEE 16(2):1024–1049 (Second Quarter 2014)
15. Ghodsi A et al (2011) Naming in content-oriented architectures, In: Proceedings of ACM SIGCOMM workshop information-centric networking, Toronto, Canada, Aug 2011
16. Dannewitz C et al (2010) Secure naming for a network of information. In: Proceedings of 13th IEEE global internet symposium '10, San Diego, CA, Mar 2010
17. Eriksson A, Ohlman B (2007) Dynamic internetworking based on late locator construction. In: 10th IEEE global internet symposium, 2007
18. D'Ambrosio M, Dannewitz C, Karl H, Vercellone V (2011) MDHT: a hierarchical name resolution service for information-centric networks. In: Proceedings of ACM SIGCOMM workshop on information-centric networking, ACM, New York, NY, USA, 2011, pp. 7–12
19. Dannewitz C, D'Ambrosio M, Karl H, Vercellone V (2013) Hierarchical DHT-based name resolution for information-centric networks, Elsevier computer communications, special issue on information-centric networking, 2013
20. Eriksson A, Ohlman B (2007) Dynamic internetworking based on late locator construction. In: 10th IEEE global internet symposium, 2007
21. Kutscher D, Ahlgren B, D'Ambrosio M, Davies E, Eriksson AE, Farrell S, Grönvall B, Imbrenda C, Kauffmann B, Kunzmann G, Lindgren A, Marsh I, Muscariello L, Ohlman B, Persson K-A, Pöyhönen P, Shehada M, Staehle D, Strandberg O, Tuononen J, Vercellone V (2012) (D.3.2) Content delivery and operations, deliverable, SAIL 7th FP EU-funded project, May 2012
22. Ain M et al (2009) D2.3–Architecture definition, component descriptions, and requirements, deliverable, PSIRP 7th FP EU-funded project, Feb 2009
23. Bloom BH (1970) Space/time Trade-offs in hash coding with allowable errors. ACM Commun 13(7):422–426
24. Miller VS (1985) Use of elliptic curves in cryptography. In: Proceedings of CRYPTO '85: the advances in cryptology. Aug 1985
25. Lagutin D (2008) Redesigning internet-the packet level authentication architecture, Licentiate's thesis, Helsinki University of Technology, Finland