http://dx.doi.org/10.1016/j.cosrev.2016.01.001

Named Data Networking: A Survey

Divya Saxena, IIT Roorkee, India
Vaskar Raychoudhury, IIT Roorkee, India
Neeraj Suri, Technische Universität, Germany
Christian Becker, Universität Mannheim, Germany
Jiannong Cao, Hong Kong Polytechnic University, Hong Kong

Abstract: **Internet was developed as a packet data network where users and data sources (server) with specific IP addresses interacted over a pre-established communication channel. This model of client-server data communication has evolved into a peer-to-peer mode of data sharing in recent times. Applications like, YouTube, Bit Torrent, social networks have revolutionized the idea of user generated contents. Modern users care only for specific data items irrespective of their sources. So, the idea of using IP addresses to identify servers hosting a particular content is losing its importance. Moreover, want of IP addresses is a challenging issue haunting the Internet community since long. The need of the time is a content-centric networking platform where data hosts are of less importance, and Named Data Networking (NDN) has been proposed to that end. NDN allows users to float a data request without any knowledge about the hosting entity. NDN can handle user mobility, security issues more efficiently than the current Internet. Although NDN has been proposed in 2010, so far, there is no survey paper studying its architecture and various schemes proposed for its different characteristic features, like, naming, adaptive forwarding and routing, caching, security, mobility, etc. In this paper, we introduce a novel taxonomy to study NDN features in depth. We have also covered several NDN applications. We conclude our survey by identifying a set of open challenges which should be addressed by researchers in due course.**

Keywords: NDN; Named Data Networking; IP; Information Centric Networking; ICN; NDN Routing; Adaptive Forwarding; NDN Mobility; NDN Caching; NDN Security; Privacy and Trust; NDN Applications.

1. INTRODUCTION

Internet was designed more than thirty years ago as a point-to-point conversation between two end hosts which allowed the users to fetch data from well-known servers. After TCP/IP protocol stack was introduced [1], packet switching allowed users to transfer text, audio, and video packets over the Internet.

Though, the Internet has shown great resilience over the years, more recently, changes in the nature of applications, user requirements, and usage patterns have significantly strained it. Recently evolving content-centric applications, like, social networking, e-commerce, YouTube [2], Netflix [3], Amazon [4], iTunes [5], etc., allow users to share texts, images, audios, and videos and have become the source of half of the world's Internet traffic. Recent surge on production and consumption of user generated contents (UGCs) are failing the Internet because it was not designed to support the newly evolving content distribution model [6][7][8]. Today, most of the application data delivery model is concerned about *what* data is needed irrespective of their locations. Moreover, support for mobility and security is not in-built in Internet, but offered as multiple patches or add-on features which may fail at times.

The aforementioned reasons urged researchers to find an efficient alternative architecture to the Internet, which will inherently support content-centric communication. Among several funded projects for designing content-based future Internet paradigms, *Named Data Networking* (NDN) came up as the promising candidate [6][7][8][9] which directly deals with application generated variable-length, location-independent names to search and pull contents for a requesting user, irrespective of their hosting entity.

The basic design principles of NDN are based on the Internet. NDN can directly use major IP services like, Domain Name Service (DNS) and inter-domain routing policies. IP routing protocols like, BGP and OSPF can be adapted to NDN with little modifications. However, NDN offers certain enhanced features as explained below. It uses data packets with content names [6] instead of source and destination addresses. The use of *unique content names* for communication allows routers to keep track of packets' states, which supports numerous functions unlike the IP routers. The data packets are *self-contained* and *independent* from where they are retrieved and where they can be forwarded. These features allow *in-network caching* of contents for fulfilling future requests and inherently support consumer mobility. In NDN, all data packets are signed by its producer and verified by the consumer, unlike IP. NDN routers support *multi-path forwarding*, i.e., they can forward a user request to multiple interfaces at the same time. Moreover, the use of content name for communication removes the need of application-specific middleware too.

NDN and Internet share the same layered hourglass architecture with functional differences between corresponding layers [6][10], as shown in Fig. 1 (a) and (b). The OSI communication model has only Internet Protocol (IP) in the Network layer. However, it is difficult to add new functionalities to the IP and to modify the existing ones. As a future Internet paradigm, NDNs network layer [6][7] must support *scalability* (support to large Internet topology and high amount of name prefixes), *security* (integrity, origin authentication and relevance of routing information), *resiliency* (to detect and recover from packet delivery performance), and *efficiency* (support multi-path forwarding and in-network caching for efficient data dissemination).

As shown in Fig. 1(b), *security* and *strategy* are two new layers added to the NDN protocol stack. Security layer provides security to each and every piece of content, unlike securing the entire communication channel in Internet. Strategy

layer is used for the stateful NDN forwarding plane, which makes forwarding decision for each incoming content request. NDN does not maintain a separate transport layer. All the functions of Internet's transport layer are embedded into the NDN forwarding plane.

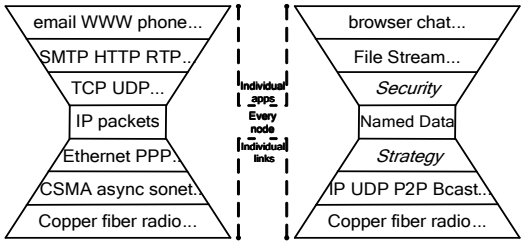

Fig. 1: Internet and NDN Hourglass Architectures [6]

NDN has evolved from the Information Centric Networking (ICN) research area as shown in Fig. 2, which has also inspired many other Internet architectures [11]. Recently, researchers have analyzed the key features and issues of NDN as the future Internet architecture (FIA) [12][13][14][15][16][17][18]. They have discussed about design principles of NDN compared to other FIAs, such as FIND [19], NEBULA [20], XIA [21], GENI [22], 4WARD [23], FIRE [24], AKARI [25], JGN2Plus [26], etc.

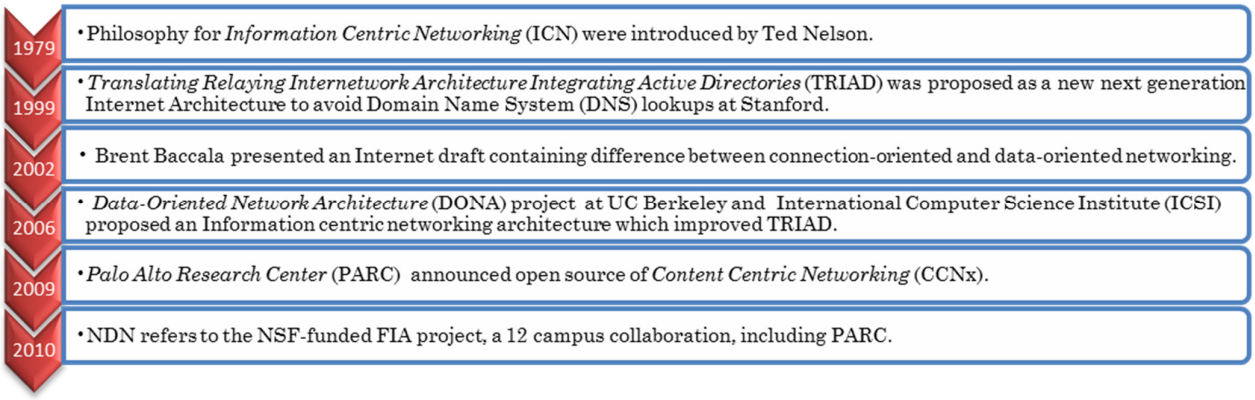

Fig. 2: NDN Timeline [9]

NDN, although novel and promising, has many open research challenges, like scalability problem in routing, want of support of wire-rate forwarding, and absence of a proper naming guideline for application development and network data delivery. A lot of work has been done on NDN within a very brief time span, but no survey exists till date to study about NDN architecture and its features in a contiguous and coherent manner. In this paper, we shall analyze NDN characteristic features and summarize several proposed techniques on various functionalities of NDN. We shall further compare/contrast features and functionalities of NDN and IP in a purely objective manner for the benefit of future researchers. In summary, in this paper, we make the following key contributions.

- We propose a feature tree-based taxonomy that organizes NDN key features and their relationships into a framework to help understand and classify the existing work.
- We provide a brief overview of NDN and its characteristic features and compare them with core functionalities of the Internet.
- We provide a detailed review of existing works on different aspects of NDN as per our proposed taxonomy, covering architecture and organizational structure, routing, data dissemination and retrieval strategies, and applications.
- We also discuss several open problems and identify directions for future research.

The remainder of this paper is organized as follows. In Section 2, we present the tree taxonomy of NDN in terms of the architecture, system services, and applications. In Section 3, we briefly explain the core features and functions of NDN. In Section 4 and 5, we provide a detailed survey and analysis of several NDN characteristic features based on our proposed taxonomy. In Section 6, we provide a functional comparison of NDN with the Internet/IP. Finally, in Section 7, we conclude the paper after discussing the challenges, open problems, and future directions of NDN research.

2. NDN CLASSIFICATION/TAXONOMY

Analyzing the architectural design, key functional characteristics, and important auxiliary support, we have proposed taxonomy of NDN as shown in Fig. 3. We have broadly classified NDN features into *system architecture*, *system services*,

and *NDN applications*. As NDN is a novel networking paradigm, we have also covered a short survey of several practical applications that have been or can be developed using NDN. This will help to convince future researchers that NDN have the potential to shoulder the responsibilities of Internet for all practical purposes in coming years. The time span of the papers covered in this survey is from Oct. 2010 to Sep. 2015.

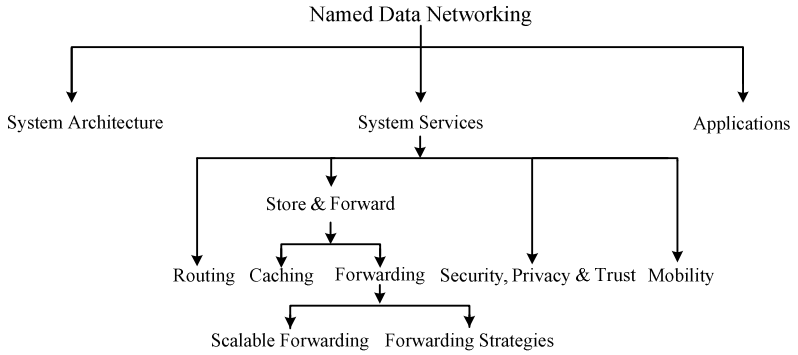

Fig. 3: Classification/Taxonomy of NDN

- ***System Architecture.*** This discusses the organization of different components and their interaction within the system along with its several working principles.
- ***System Services.*** It concerns about the main functional characteristics of NDN, which includes *routing*, *store-&-forward*, *security, privacy and trust*, and *mobility*. Since, NDN supports in-network caching of user requested content while delivering the same. Store-&-forward can further be sub-divided into *caching* and *forwarding*. Security is the best in-built feature of the NDN architecture which provides secure communication through named data. While, mobility deals with the handling of all issues related to the consumer and provider mobility.
- ***Applications.*** It shows the practical benefits of the well-designed NDN architecture in real life through different traditional as well as novel applications. Based on the nature of the underlying architecture design, NDN supports these features for the development of robust network and applications. Naming in NDN is purely application-specific. So, we have discussed various naming schemes in existing NDN literature along with the several applications.

3. NDN SYSTEM ARCHITECTURE AND WORKING PRINCIPLES

NDN fetches content by names where naming is implemented as a part to facilitate content search and/or retrieval. Naming schemes are application-specific and are independent of the network [6][10]. The names that are used to get a global data must be globally unique. A *content name* (CN) (used by an academic institute) and its hierarchical structure are shown in Fig. 4. In order to access specific course content, a user will type related keywords, based on which search application will generate a request with the following *CN/edu/yale/oyc/computer/comp-201/lecture-1*.

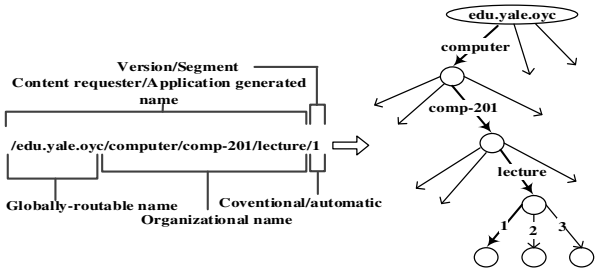

Fig. 4: Human Readable *CN* and its Hierarchical Representation [10]

NDN communication is initiated by the consumer in the form of an Interest packet (*I_pkt*) (see Fig. 5(a)). When an *I_pkt* reaches to content publisher or a node having valid requested content, a Data packet (*D_pkt*) (see Fig. 5(b)) is issued for that *I_pkt*. *CNs* are embedded in both the *I_pkt* and *D_pkt*. *D_pkt* retraces the path of the *I_pkt* in reverse (symmetric communication) to reach the user. In this paper, the node which returns (a cached copy of) the requested content is termed as '*provider*', and the node which actually produced (created or generated) the content, is termed as '*producer*'.

(a) (b)

Fig. 5: Packets in NDN Architecture [6]

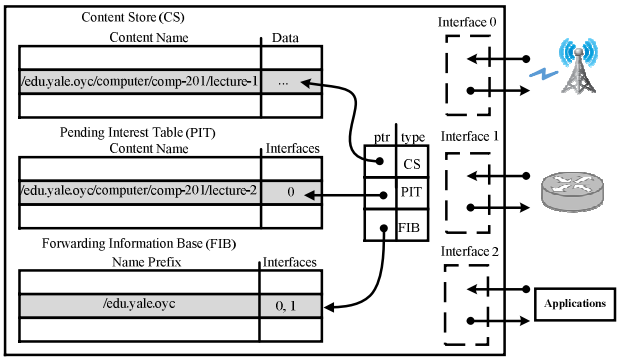

Fig. 6: NDN Node and its Forwarding Structures [10]

Each NDN router maintains the following three data structures [10] (Fig. 6).

- ***Content Store (CS).*** In NDN, each *D_pkt* is idempotent, self-identifying (*CN* identifies the data a user wants) and self-authenticating (contains producer signature) [10]. Therefore, each *D_pkt* can be used by several consumers, e.g., multiple users may read the same newspaper. For increasing the sharing probability, saving bandwidth, and reducing content retrieval time, NDN router caches a copy of *D_pkt* passing through them (based on the local caching policy) in the CS until they get replaced by the new content (because of finite cache size). Searching CS entries take place through exact matching (character-by-character match) of names.
- ***Pending Interest Table (PIT).*** PIT maintains an entry for each incoming *I_pkt* until its corresponding *D_pkt* arrives or the entry lifetime expires, whichever is earlier. These PIT entries are used to forward *D_pkt* downstream to the consumer. Searching PIT entries take place through exact matching of names.
- ***Forwarding Information Base (FIB).*** FIB maintains next-hop(s) and other information for each reachable destination name prefix. The FIB is populated by the routing protocol and used for forwarding the *I_pkt* upstream.

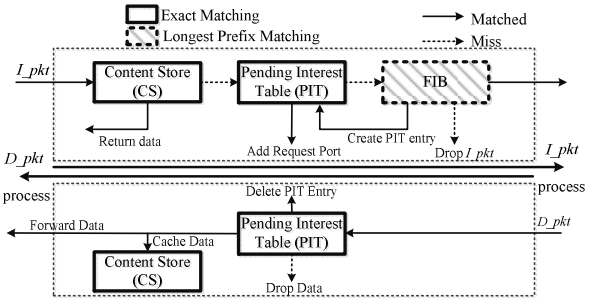

Fig.7: Forwarding Process at NDN Node [6]

The forwarding process of an *I_pkt* and the corresponding *D_pkt* at an NDN router is illustrated in Fig. 7. Whenever an *I_pkt* from a user *U* arrives at an NDN router, the associated *CN* is searched in the CS to find a match. If a matching content is found, the NDN router forwards the contents to *U* through a *D_pkt*, signed by the producer's key [10]. Otherwise, the *CN* is looked up in the PIT. If a matching PIT entry is found, *I_pkt's* incoming interface will be added to the list of interface(s) which is known as *Interest aggregation*. So, when the corresponding *D_pkt* is available, all interested users will receive a copy of that *D_pkt*. If no PIT entry is found for an incoming *I_pkt*, the *I_pkt* is passed to the router's FIB which performs a longest prefix match (LPM), e.g., for a *CN '/p/q/r/s'*, FIB will find out possible LPMs like, *'/p'*, *'/p/q'*, *'/p/q/r'* and *'/p/q/r/s'* as shown in Fig. 8. When a matching FIB entry is found for these LPMs, *I_pkt* is forwarded to the corresponding next-hop(s) and a new PIT entry will be created with its incoming interface for the same. Otherwise, either the *I_pkt* is flooded to all outgoing interfaces or deleted as the router's forwarding policy decides.

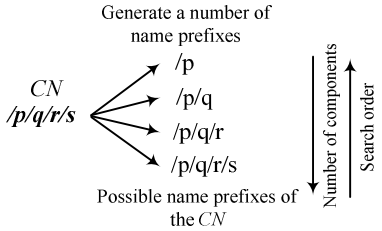

Fig. 8: Name Lookup Process in the NDN's FIB [27]

Whenever a *D_pkt* returns to an NDN router, all PIT entries are searched to find a matching *CN*. If a corresponding PIT entry is available, the *D_pkt* is forwarded to all the interfaces mentioned in the list of incoming interface(s). After that, the PIT entry is deleted and the content is stored in the CS based on the local caching policy, for serving future requests for the same content. If no matching entry exists (may be because its lifetime is over), the *D_pkt* is dropped.

4. NDN SYSTEM SERVICES

In this Section, we shall introduce an elaborate survey of the major routing, caching, forwarding, security, privacy and trust, and mobility techniques proposed for NDN in recent years.

4.1 Routing

In NDN, routing is used for setting the topology and policies and handling their long-term changes, as well as for updating the forwarding table. NDN routing protocol coordinates with NDN forwarding plane for interface ranking and probing [28][29]. In NDN, the only difference between routing and forwarding is, while routing decides about the availability of routes, forwarding makes decisions about the preference and usage of routes based on their performance/status. The routing algorithm suitable for Internet, i.e., link-state and distance-vector can be used for NDN with slight modifications [6]. Both IP and NDN use the FIB to store routing related information. IP searches the destination address in the FIB to find the next-hop and to deliver packets to the destination address, not necessarily via the best path. NDN searches the name prefix in the FIB to find the next-hop(s) and fetches data, not necessarily the nearest copy. Current Internet routing protocols can be adapted for usage in NDN by changing the message types (*I_pkt* / *D_pkt*) and adding multi-path forwarding. As, NDN deals with *CN*s, NDNs routing table may consume more memory space in compared to the IP routing table. The main performance metrics to evaluate the performance of NDN routing approaches [30] are *CPU utilization* (total number of CPU resources like, CPU time, used for routing the named contents), *PIT count* (total number of PIT entries existing at an NDN router), *memory consumption* (total memory consumed during named content routing), *network utilization* (total data transferred over the network), *Interest re-transmission rate* (total number of re-issued *I_pkt(s)* as a result of either packet loss or lifetime expiry), and *time-to-completion* (total time required to satisfy a content request), etc. There are mainly three major routing protocols [31][32][33] in NDN and all are designed to perform only intra-domain routing as shown in Fig. 9.

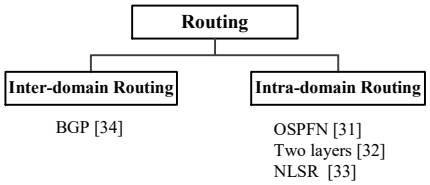

Fig. 9: Classification of Different Routing Schemes

Dibenedetto, et al. [34] have argued that inter-domain routing protocol like, Border Gateway Protocol (BGP) must be considered in the future Internet architecture. Inter-domain routing protocols make routing decisions based on paths and network policies while intra-domain routing protocols like, Link State Routing (LSR) and Open Shortest Path First (OSPF) calculate the shortest path among the routers. Network operators set up and implement policies for route selection. The authors have also discussed about the different parameters like, routing, cache access, and FIB usage for inter-domain routing policies. Moreover, the authors have also focused towards NDN economic incentives for cache sharing, routing rebate, CS, and PIT policies.

4.1.1 OSPF-based Routing for NDN

For supporting the name based routing in NDN, IP based routing protocol OSPF is extended to distribute name prefixes and to calculate routes for them. A name-based dynamic routing protocol OSPF for Named-data (OSPFN) [31], is designed, implemented and tested on the NDN testbed [35]. The OSPFN routing protocol uses three main modules which run in parallel at a node: *Content Centric Networking Daemon* (CCND), *OSPF Daemon* (OSPFD), and *OSPFN*. CCND handles the forwarding of NDN packets. OSPFD floods the Opaque Link State Advertisements (OLSA) to the entire network. OSPFN builds the name OLSA and provides them to its local OSPFD.

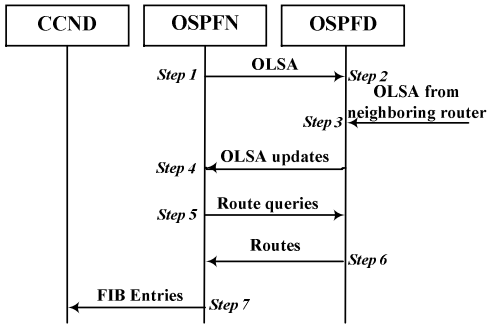

Fig.10: Sequence of Messages Exchange among OSPFN, OSPFD, and CCND [31]

OSPFN routing works in seven main steps as stated and explained here. *Step 1*, when a node boots up, it prepares name OLSA for each name prefix it wants to announce in the network. *Step 2*, OSPFN provides OLSA to its local OSPFD for flooding in the entire network. *Step 3*, when a node's OSPFD receives an OLSA, it provides updated OLSA to its local OSPFN as shown in Fig. 10. *Step 4*, If OLSA is new and updated, OSPFN extracts the name prefix along with the router-ID from name OLSA and maintains it into a name prefix table. *Step 5*, OSPFN sends query to OSPFD for finding the next-hop(s) to reach the origin router(s) of each name prefix. *Step 6*, after receiving a query message from OSPFN, OSPFD checks its routing table for next-hop list, and path cost and send the routes back to OSPFN. *Step 7*, OSPFN prepares FIB entries (name prefix, next-hop(s), and path cost) and updates to CCND.

OSPFN also supports multi-path routing only by ranking the list of next-hops in the CCND's FIB. Still, OSPFN uses IP addresses as Router-IDs and relies on Generic Routing Encapsulation (GRE) tunnels to cross legacy networks and returns only single best next-hop for each name prefix. Management of these IP addresses and tunnels increases overhead and limits support for NDN multi-path forwarding.

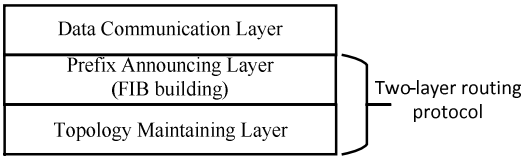

Fig. 11: Layered Routing Hierarchy [32]

4.1.2 Two-layer Routing for NDN

Dai, et al. [32] have presented a two-layer multi-path routing protocol for NDN by combining the *Topology Maintaining* (TM) layer and the *Prefix Announcing* (PA) layer together as shown in Fig. 11. TM layer maintains topology of the entire network and calculates the shortest path. TM layer provides the shortest path to upper layer as a service. The upper layer PA actively publishes the content by sending announcements to all other nodes using single-source shortest-path tree available at the router. Also, router maintains their FIB using incoming announcements. In passive service mode, an Interest is flooded till the *I_pkt* reached to any corresponding content provider (or, producer).

Actively publishing any content may increase the FIB entries and passive service of the contents can increase the traffic in the network. These factors can degrade the system performance and scalability of NDN routing algorithms. To deal with these problems, the authors have also proposed popularity-based active publishing. Now, the main task of the upper layer PA is to actively publish popular contents decided on the basis of their access frequency and passively publish unpopular contents on the basis of their requirements. Moreover, another solution for suppressing the FIB size is popularity threshold, which is shown in Fig.12 (a). Whenever an announcement *A* arrives at a router, it compares announcement popularity *AP* with router's local popularity threshold *T* (calculated on the basis of individual router's memory size and other factors). If *AP* >*T*, then announcement name prefix is maintained in the FIB, otherwise, not.

Name aggregation can also be applied for suppressing the FIB size as shown in Fig. 12 (b). E.g., '*/in/co/google/NDN*', '*/in/co/google/IP*', and '*/in/co/google/CCN*' share a common prefix '*/in/co/google*'. So, these name prefixes can be suppressed into '*/in/co/google*'.

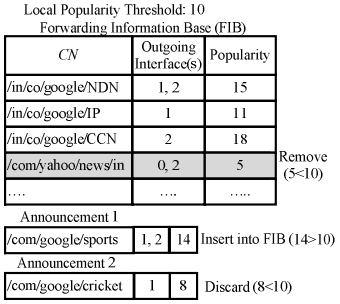

Fig. 12 (a): Popularity-Based FIB Size Suppression [32]

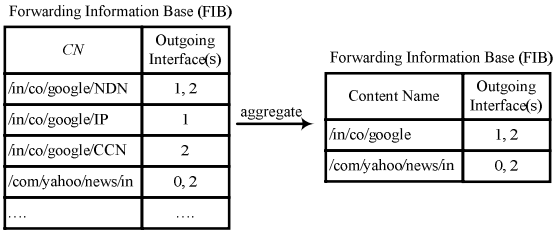

Fig. 12 (b): FIB Entries Aggregation [32]

4.1.3 Link State Routing in NDN

Named-data Link State Routing protocol (NLSR) [33], is a link-state routing protocol which runs on the top of NDN where each *D_pkt* used for routing updates are signed by the originating router for supporting authentication. There are four key factors of NLSR routing protocol. First is, to name the routers, links, process, data, and keys. Second is, distribution of keys and trust for them. Third is, to pull the routing updates dissemination instead of pushing updates as in OSPF. Fourth is, to rank the interfaces for supporting multi-path forwarding.

NLSR propagates the LSA for building the network topology and distributing the name prefixes. Each node maintains the Link state Database (LSDB) for storing the latest LSA. There are two types of LSA: *adjacency LSA*, which advertises all active links connecting an NDN router to its neighbors and *prefix LSA*, which advertises registered name prefixes. Through this adjacency LSA, every NLSR node finds a network topology. Whenever NLSR finds any failure or recovery in the connected links or neighbors' processes, it sends recent or new LSA to all routers of the network. It also propagates new LSA when any name prefix is added or deleted. NLSR uses NDN *I_pkt*s and *D_pkt*s, therefore, there is need to name the routers and the routing updates where a NLSR process can be named as /<network>/<site>/<router>/NLSR. Routers also exchange the hashes of the LSDB to detect contradictions and perform recoveries from them, which is known as *hop-by-hop synchronization*. This approach reduces the flooding overhead in a stable network because there is a need to exchange only one hash instead of many LSAs. The NLSR routing protocol uses the CCNx synchronization protocol, or Sync for disseminating the updated (or recent) LSA among the neighboring routers where it remains attached with the CCNx repository (or, Repo). Repo is an application which supports communication with CCND and maintains collections of named data, named as *slices*. These slices (slice contains LSAs) are kept in the Sync with other identical slices in neighbor nodes.

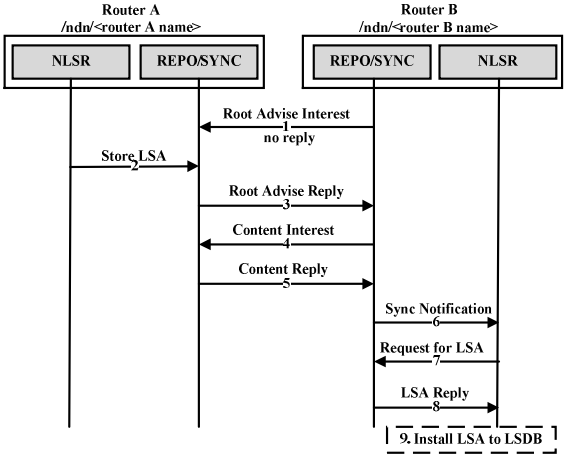

Fig. 13: LSA Dissemination from Router A to Router B using CCNx Sync/Repo [33]

The main steps for LSA dissemination among two routers *A* and *B*, as shown in Fig. 13, are as follows. *Step 1*, when Sync of router *B* wants to synchronize its slice, then, it periodically sends a special *I_pkt*, Root Advise Interest with the hash value of its local slice to all neighbor nodes. *Step 2*, whenever the router *A's* NLSR creates a new LSA, router *A* stores in its sync-slice. *Step 3*, if the router *A's* slice-hash value does not match to router *B*, then router *A* sends Root Advise Reply to router *B* with a new hash value of its local slice. *Step 4*, router *B's* sync compares the hashes and request router *A* for latest hash. Router *B's* Sync identifies the data required for synchronization and sends content Interest. *Step 5*, router *A* provides content reply to router *B*. *Step 6*, router *B's* sync sends *CN* to its local NLSR. *Step 7 and 8*, NLSR at the router *B* retrieves the data from its repo. *Step 9*, NLSR at the router *B* sends recent LSA to its LSDB.

For detecting link failures or NLSR remote process failure, NLSR at the router sends info Interest to its neighbors periodically and waits for the response. If no response arrives during this time, that link is considered down. For recovery of this link, NLSR at the router keeps sending these *I_pkt*s at higher interval of time. When NLSR gets a response, it sets the link status to active, updates adjacency LSA, disseminates LSA and performs routing table computation. Experiments are

performed to evaluate the NLSR processing time, messaging overhead and convergence time, which shows that NLSR supports *update dissemination*, *built-in update authentication*, and *multi-path forwarding* efficiently. For real-time evaluation, NLSR is implemented on the NDN testbed having 26 nodes (various institutes or organizations), and 62 links [35].

TABLE I DIFFERENCES BETWEEN IP-BASED LSR, NLSR AND TWO-LAYER INTRA-DOMAIN ROUTING

Properties	IP-based LSR	Two-layer	NLSR
***Sync* Support**	No	No	Yes
PUSH/PULL Updates	PUSH	PULL	PULL
Support of Multi-path Forwarding	Limited	Limited	Full
Forwarding Strategies	No	No	Yes
Authenticity of Update Packets	No	No	Yes
Stability and Scalability	Poor due to fast routing convergence	Best due to slow routing convergence	Best due to slow routing convergence

The main differences between *IP-based LSR*, *NLSR* and *two-layer* intra-domain routing protocols are shown in Table I. The two main differences between IP-based routing protocol LSR and NDN-based routing protocol, NLSR are stated below. First, NLSR takes benefit of NDN authentication by using NDN *I_pkt*s / *D_pkt*s for advertising routing updates. Second, NLSR supports NDN adaptive forwarding strategies like, rank each name prefix for better support of multi-path forwarding.

4.2 Caching

Content caching [6] at intermediate nodes, also called *in-network storage*, is fundamentally important to support the basic concept of content-centric, peer-to-peer data delivery model of NDN at low cost. Caching in NDN has several benefits. Caching contents produced by other nodes help to truly dissociate contents from their producers. Also, it reduces the overhead at the producer side and avoids a single point of failure by making available multiple copies of the same content in the network. It provides high benefits to dynamic contents in case of multicast or retransmission due to packet loss [6][10]. Also, it significantly reduces the network load and data dissemination latency [7][10].

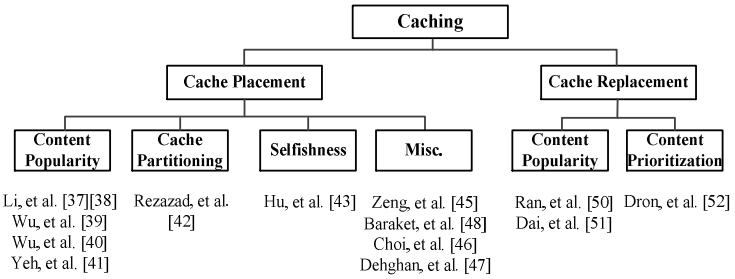

Fig. 14: Classification of Different Caching Strategies

Cache *performance measurement* is usually carried out using the following four metrics: *hit ratio*, *content retrieval delay* (the total time elapsed between the instant a content request is generated and the instant it is received by the consumer), *average number of hops traversed* (to locate and to fetch a requested content is also measured to gauze how well contents are distributed across the network). Another useful cache performance metric can be *dissemination speed* which measures the time required to disseminate contents down to the network edge.

Cache *decision policy* determines whether to cache the *D_pkt* in the intermediate routers or not. For an efficient caching algorithm, two main important questions are *where to cache the content?* and *which content to replace first?* Therefore, we have broadly classified caching schemes into *cache placement* (decision to place the content at the network or not) and *cache replacement* (decision to cache the content at the router) (see Fig. 14).

4.2.1 Cache Placement

Leave Copy Everywhere (LCE) is the cache placement policy in-built in the NDN architecture [6]. In LCE, a *D_pkt* will be cached in all routers existing between the producer (or, provider) and consumer. This scheme induces significant cache redundancy, i.e., the same content is cached at multiple nodes, which reduces the diversity of the cached contents in the whole system. Some existing cache placement algorithms for reducing the cache redundancy, are *Leaving copies with Probability* (LCProb), and *Leaving copies with Uniform Probability* (LCUniP) [36]. LCProb caches the content at the router using the caching probability *1/(hop count)* and LCUniP caches the content at the router using uniform probability.

In order to increase the utility of cached contents in the whole system, there is need to store popular contents at the network edge which will minimize the *content downloading latency* and will improve the whole network's *cache diversity* [36]. There is also need of an effective coordination scheme [36] among the routers for reducing cache redundancy and improving cache diversity. There are two types of cache coordination schemes - *explicit* and *implicit*. In the explicit scheme, content access frequency, cache network topology, and network cache's state information must be known apriori for making

placement decisions about content. This scheme can be further classified into three categories: *global coordination* (decision involves all cache nodes), *path coordination* (decision involves all the cache nodes along the path between the consumer and the producer (or, provider)), and *neighbourhood coordination* (involves all the neighbours of a cache node). In implicit cache coordination, each cache node does not need to know the state information of the other cache nodes and they exchange little information with others before taking the content placement decision.

We have classified cache placement into three main categories as *content popularity*, *cache partitioning*, and *selfishness* (as shown in Fig. 14). Content popularity is to cache only popular contents on the network for reducing the inter-domain traffic. Content popularity is calculated through the total number of requests for a content which remains stable for a short period of time. In NDN network, different types of traffics are stored in the CS of an NDN node. These network traffics of different characteristics require to share limited storage space efficiently. Therefore, cache partitioning is required to partition the cache space dynamically for different traffic/applications. In Selfishness, every node is considered to act selfishly because nodes will provide cache coordination only if their own access cost of content is reduced.

Recently, some cache mechanisms have been included in the NDN, using content popularity to minimize the inter-domain traffic [37][38][39], both inter and intra-domain traffic [40], and to support network load balancing [41]. Cache partitioning [42] has been studied using network traffic analysis, to increase the cache performance like, cache hit ratio. In recent work, selfishness is introduced for high cache hit [43] using neighborhood coordination.

A. Content Popularity

In popularity-based caching schemes, each access router (AR) maintains the access frequency of popular contents. Moreover, each router periodically calculates the content access profile from the previous request statistics. Li, et al. [37][38] have proposed a model for minimizing the inter-ISP traffic and number of access hops by maintaining replica of popular contents in dynamic networks. The main aim of these caching schemes is to cache frequently requested contents at selected routers inside the ISP. They outperform existing caching algorithms, such as LCE, LCProb, and LCUniP. Authors have extended their work in [38] and presented three coordinated caching algorithms for making cache placement decisions along the forwarding path. All three proposed caching schemes outperform LCE. Wu, et al. [39] have proposed a popularity-based coordinated caching strategy named as, *Effective Multi-path caching* (EMC) for inter-ISPs with multiple gateways and multi-path routing which reduces the inter-ISP traffic and content access latency. EMC finds the replica placement on the basis of most recent request statistics and supports online caching decisions for cache placement by knowing the global popularity of content. EMC outperforms LCE and LCProb schemes. The main overheads of coordinated caching decisions are, storage cost, communication cost, and execution time.

For reducing both the inter-domain and intra-domain traffic of an NDN backbone network, Wu, et al. [40] have proposed popularity-based coordinated caching strategy named as, *Distributed Caching with Coordination* (DCC). For finding the popular contents, a local content is ranked on the basis of its request rate at a router. Then, weighted popularity is periodically calculated using the content rank and its total number of requests. To determine the global popularity of content, local weighted popularity of that content at every router is added and defined as Summed Weighted Popularity (SWP). DCC outperforms LCE and Random Cache (RanCache) (randomly cache content along the path) in terms of scalability. Yeh, et al. [41] have shown that static and centralized caching algorithms which induce overhead under increasing traffic load. Therefore, there is a need of distributed and dynamic caching algorithm that can perform better under changing contents and user demands. The authors have presented a framework using Virtual Interest Packets (VIPs) which finds the demand of a particular content in the network. VIP framework uses virtual control plane to control VIPs and actual plane to handle normal *I_pkt*s and *D_pkt*s. A distributed control algorithm is developed using VIPs to increase the user demand rate acceptable in the network and a caching and forwarding algorithm is developed for actual plane using flow rates and queue lengths of the VIPs. Simulation results show that proposed framework outperforms LCE, LRU, LFU, FIFO (to choose first cached item for replacement), FIXP (cache on the basis of fixed probability), UNIF (to choose currently cached item, uniformly at random, for replacement), and BIAS (to choose two currently cached items uniformly at random, for replacement).

B. Cache Partitioning

Cache partitioning [42] has been studied using network traffic analysis, to increase the cache performance like, cache hit ratio. For better cache performance, the reusable contents of some applications like, videos and web, should not be replaced by non-cacheable items like, email, telephony apps, etc. For this, CS at the NDN router can be partitioned for different types of traffic class, e.g., Constant Bit Rate (CBR) which is used for multimedia applications and non-Constant Bit Rate (non-CBR). Cache partitioning can be of two types: *static* and *dynamic*. In static partitioning, CS is partitioned into fixed shares which cannot be used by other traffic classes. In dynamic partitioning, one traffic class can use another class-cache-space if other traffic class does not need cache space at that time. Rezazad, et al. [42] have introduced dynamic partitioning of CS for NDN traffics by using the cache miss equation which is used to dynamically partition database buffers. The main objective of the proposed scheme is to minimize the cache miss probability and ensure the fairness across different NDN traffics. The results show that dynamic partitioning reduces cache miss probability for edge routers.

C. Selfishness

Hu, et al. [43] have proposed a *Not So Cooperative Cache* (NSCC) scheme where a selfish node will cache the content iff cache coordination reduces its own access cost either by getting the data from its local cache or from its neighbor's cache. The main concern of the proposed scheme is to find what contents to cache at each node for minimizing the individual node access costs by some amount. For this, each NSCC node maintains four components, *Interest/Data processor*, *Request rate synchronizer*, *compute cache*, and *local cache*, where nodes in the network are selfish. Interest/Data processor processes *I_pkts* and keeps track of content's local popularity. Request rate synchronizer is the one which shares this local popularity with other NSCC nodes and learns popular contents of other NSCC nodes. The request rate synchronizer provides a view of content's popularity to the compute cache. The main objective of compute cache is to find what contents should be cached by local cache and what content would be cached by other nodes. To find the global object placement at NSCC nodes, compute cache uses a *game theory approach* (each nodes act independently and greedily caches most popular contents) developed by [44]. Local cache contains the contents suggested by compute cache. So, local cache retrieves the content from the content providers and stores them. Simulation results show that the proposed scheme (greedy NSCC nodes) achieves better cache hit ratio over NSCC nodes.

D. Miscellaneous

Zeng, et al. [45] have shown the impact of cached data distribution on mobile ad hoc NDN network performance. For distributing contents along a returning path, an integer data interval parameter is set in *D_pkt*. This parameter decides the cached data interval along the Interest-forwarded-path which is set by the data provider of the content. Each time content is forwarded downstream, data interval value is decremented by 1. When the interval value becomes 0 at any intermediate node, CS caches the content and resets the data interval value. This procedure is repeated till the *D_pkt* is received by the consumer. Simulation results show that the proposed caching scheme outperforms Ad-hoc On-Demand Distance Vector (AODV) provided the data interval parameter is tuned for cache distribution and better performance.

In addition, some authors have added caching with the routing and PIT to increase the performance of NDN forwarding daemon. Choi, et al. [46] have proposed Coordinated Routing and Caching (CoRC) scheme to minimize the effect of routing scalability and to increase the in-network caching efficiency. Dehghan, et al. [47] have analyzed two classes of TTL-based caching policies with PIT, where the timer could be set only once, or be reset with every request for the content.

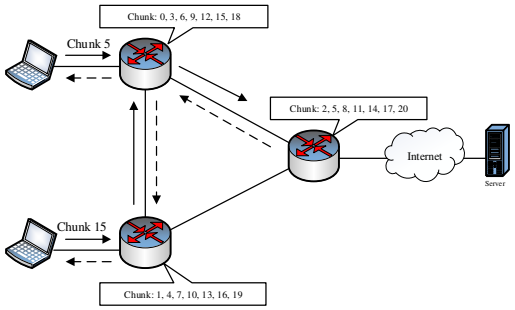

Fig. 15: Caching All Contents by Hashing (CACH) Caching Scheme [48]

Barakat, et al. [48] have proposed *off-path caching* over the pre-existing on-path caching techniques in order to increase the total volume of contents that can be cached. Instead of caching content on the path from content producer (or, provider) to consumer, content is cached in the off-path caches within the network and traffic is deflected towards these caches. This scheme improves the global hit ratio by utilizing the cache space in the network. Meanwhile, a hash based CACH [48] scheme is proposed which caches contents spread across the network. In CACH, each *CN* is hashed and its hash value decides the position of the caching router for content as shown in Fig. 15. When chunk 5 will arrive at router 1, it will calculate the hash value of chunk 5. If hash value points that this chunk should be stored at router 2 then, router 1 will forward chunk 5 to router 2. In CACH, for all content, randomly one single deflection point is selected (irrespective of its popularity). Experiments on real topologies, such as Rocketfuel [49], show that CACH (off-path caching) performs better as compared to on-path caching, in terms of bandwidth utilization and hit ratio.

4.2.2 Cache Replacement

A well-known cache replacement policy is *Least Recently Used* (LRU) (discard least recently accessed content) which performs well and increases the probability of a cache hit by storing most recent data for some more time. Another important cache replacement policy is *Least Frequently Used* (LFU) where less frequently used contents are discarded first. The cache decision timing can be based on the content arrival at the router and content replacement. For better network performance, content should not be deleted from the cache, it can be shifted one-level upstream in the cache hierarchy for caching. We have further classified cache replacement as *content popularity* and *content prioritization* where less popular and low priority content is replaced first.

A. Content Popularity

In recent years, content popularity has been addressed for cache replacement schemes by maintaining a data structure having *CN*s with its popularity, available at the router's cache [50][51]. In dynamic network, cache replacement also plays an important role for better network performance. Access-time-pattern based replacement policies like, LRU and LFU, cannot make full use of the popularity of contents in NDN network. In NDN network, there is need of content popularity based cache replacement schemes for an efficient caching algorithm. For better cache performance, Ran, et al. [50] have proposed a cache replacement scheme based on content popularity (CCP). A data structure, Content Popularity Table (CPT) is added with the CS which keeps the information like, *CN*, cache hit and previous and current popularity. On the basis of content access frequency and cache hit, periodically CCP calculates the current popularity of content and replace a content having minimum popularity among existing contents. The proposed scheme outperforms LRU and LFU. The proposed approach consumes most of the CPU time for updating the CCP table which will not perform well in large scale networks. Therefore, to minimize the resource consumption at the resource constrained routers, Dai, et al. [51] have introduced an online and space-efficient bloom-filter based method to determine the content popularity at line speed.

B. Content Prioritization

Recently, content priority has been adopted to minimize the access latency of cached content using knapsack problem [52]. A priority assigned to content decides which content will be exchanged first. Whenever two mobile nodes meet for a short time, during that encounter time they exchange their high priority contents. In highly dynamic network, content prioritization plays an important role in applications performance. High priority content will be available more in the network in compared to low priority contents. Low priority content will also suffer from high access latency. The big question is how to decide the priority of the content. The two main priority deciding factors can be content demand and the common information generated/exchanged among nodes.

Dron, et al. [52] have proposed in-network content prioritization policy for the cache replacement and has also pointed out the benefits of naming data for information-maximizing cached content delivery in ad hoc networks. Each item stored in the cache is labeled as either *hot* or *cold*. Whenever an encounter occurs, contents labeled as hot are exchanged first before cold contents. For finding the hot content among the cached data, the authors have formulated the knapsack problem where items in the knapsack must maximize utility across all user query replies. Those utility-maximizing items are labeled hot. The three performance metrics used for analyzing the information-maximizing policy are *coverage per query* (in case of sensor, amount of space covered by query response)*, responses per query* (throughput) and *average delay* (latency). Proposed scheme has better performance over LRU and Intention Caching (IC) (used in DTN, where intentionally data is cached at a set of network central locations) [53] in which it simultaneously delivers a higher coverage, supports a higher effective bandwidth, and offers a lower latency. We have summarized caching strategies in Table II listed below.

TABLE II COMPARISON BETWEEN DIFFERENT CACHING STRATEGIES

Caching Schemes	Caching Strategy	Decision Basis	Decision Timing	Cache Redundancy	Dissemination Speed
Zhang, et al. [6]	Cache Placement	No	Content Arrival	High	Fast
Li, et al. [37][38]	Cache Placement	Content popularity	Content Arrival	Medium	Medium
Wu, et al. [39]	Cache Placement	Content popularity rate	Content Arrival	Medium	Medium
Wu, et al. [40]	Cache Placement	Access cost	Content Arrival	Low	Fast
Yeh, et al. [41]	Cache Placement	Content popularity	Content Arrival	Medium	Medium
Hu, et al. [43]	Cache Placement	Selfishness	Content Arrival	Medium	Slow
Zeng, et al. [45]	Cache Placement	Data interval *D*	Content Arrival	Depends on *D*	Depends on *D*
Barakat, et al. [48]	Cache Placement	Chunk number	Content Arrival	Low	Fast
Ran, et al. [50]	Content Replacement	Content popularity	Content Arrival	High	Fast
Dai, et al. [51]	Content Replacement	Content popularity	Content Arrival	High	Fast
Dron, et al. [52]	Content Replacement	Content popularity	Content Arrival	High	Fast

4.3 Forwarding

As the NDN forwarding plane is able to detect failures (node, link, or packet) and perform recoveries, routing does not need to perform continuous FIB updates which improve the scalability and stability of the NDN routing plane [7][8]. We have classified NDN forwarding into *scalable forwarding* and *forwarding strategies*. Scalable forwarding supports intelligent and stateful forwarding. Through this, NDN routers can measure packet's RTT, throughput, packet losses, and alternative path during overhead and congestion.

Forwarding plane in NDN also acts as a control plane because forwarding strategies perform all the decisions needed for the *I_pkt* / *D_pkt* forwarding.

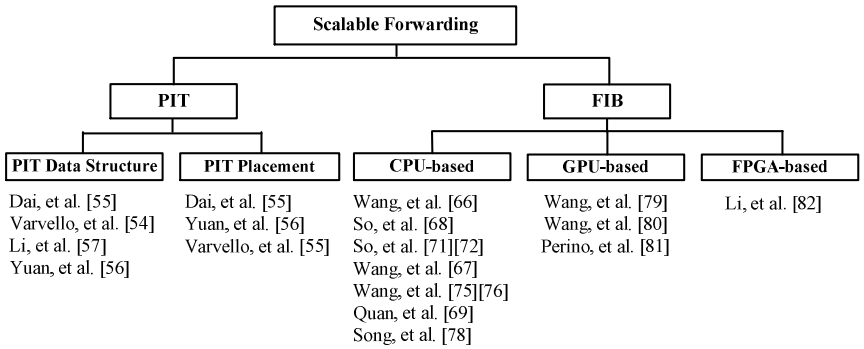

Fig. 16: Classification of Forwarding Techniques

4.3.1 Scalable Forwarding

NDN forwards the data on the basis of variable length unbounded names and has a read-write forwarding plane, which requires per-packet updates at the line speed. Therefore, the main challenges of NDN scalable forwarding are to provide fast name lookup with low memory cost.

There is trade-off between memory capacity and its access time. On-chip memory, SRAM has 4.25 MB capacity and takes 1 ns for access, Off-chip memory like SRAM has 25 MB capacity and takes 4 ns for access, RLDRAM has 250 MB capacity and takes 15 ns, and DRAM has 10-100 GB capacity and takes 55 ns [54]. For implementing PIT and FIB at the router, it is preferred to fit millions or billions of entries in the on-chip memory to achieve wire rate forwarding.

Name	**List of Nonce**	**List of Incoming Interfaces**		**List of Outgoing Interfaces**	
CN	Nonce	Interface ID	Timer	Interface ID	Send-time

Fig. 17: Structure of a PIT Entry [6]

A. Pending Interest Table (PIT)

NDN forwarding is considered to be stateful as an NDN router maintains a PIT entry for every *I_pkt*. Each PIT entry has five fields (Fig. 17) - *CN*, *nonce*, *incoming* and *outgoing interface(s)*, and *timer*. The nonce is 4-octet long byte-string, and in combination with *CN*, it uniquely identifies an *I_pkt* and prevents duplicate packet forwarding (loop detection). The <outgoing interface(s), send-time> tuple helps to estimate the corresponding name prefix's Round Trip Time (RTT) and hence the performance of the interface. Whenever an *I_pkt* is forwarded by the NDN router, a timer is associated with a corresponding PIT entry. If no *D_pkt* is returned till timer expiry, the matching PIT entry is deleted. There are three main PIT operations: *insert a new entry*, *delete a serviced entry*, and *update an existing entry* (add additional incoming interface(s)). All the operations require to search the PIT for the existence of a corresponding entry. So, the storage requirements and search time, increase proportionally with the PIT size, making it a bottleneck in NDN routing performance. We have broadly classified PIT design into two categories, *PIT data structure* and *PIT placement* as shown in Fig. 16. Researchers have adopted data structure like, trie [55], hash table [54][56], and bloom filter [57] to implement PIT on the resource constrained routers to reduce memory consumption and access time.

- ***PIT Data Structure.*** Dai, et al. [55] have presented a Name Component Encoding (NCE) scheme based on Encoded Name Prefix Trie (ENPT) which encodes the components of *CN* with 32-bit integers in order to reduce the PIT size and PIT access time. NCE uses the hash function to encode a component. After encoding the components, encoded names are inserted in the ENPT. The main drawback of the proposed technique is that code generation of n^{th} component depends on the code of the $(n-1)^{th}$ component which may degrade the system performance. Varvello, et al. [54] have proposed a PIT data structure, based on d-left open-addressed hash table (D-HT) [58]. Authors have performed experiments where D-HT outperforms Counting Bloom Filter (CBF) [59][60], hash table [61][62], and NCE [55]. To further reduce the on-chip memory and time of a PIT over the hash table and NCE technique, Li, et al. [57] have used modified bloom filter [63] and proposed a new mapping bloom filter (MBF). MBF is a data structure which supports querying and mapping of the set elements in the memory and minimizes on-chip memory cost. MaPIT performs better over NCE and hash table. Yuan, et al. [56] have focused the attention towards the compact storage of PIT design for core routers as they are mainly responsible for network performance. Authors proposed to use fixed length fingerprints (bit strings) using d-left hash table [58] for the PIT rather than *CNs* (name strings). In d-left hash table, hash buckets are grouped in d-subtables and to insert an item, all the d-subtable are checked and inserted in the lightly loaded sub-table. Edge routers support Interest aggregation and minimize the most of the traffic for core routers. Carofiglio, et al. [64] have presented an analytical model of PIT dynamics to analyze the average and maximum PIT size at steady state.

In Table III, we have provided, in detail, experimental environments and results, as discussed for different PIT's data structures.

TABLE III TECHNIQUES USED FOR PIT LOOKUP

Scheme Name	Data Structures	Algorithm Type	Performance	Pros	Cons
Dai, et al. [55]	Hash table, trie, arrays	Sequential and parallel	Performs better than hash table and NPT in storage.	Trie minimizes the impact of redundant information at storage.	Required memory access is proportional to number of component in the *CN*.
Varvello, et al. [54]	D-HT	Sequential and parallel	Able to sustain high load, i.e., high number of packets per second in compared to NCE, and CBF.	Hash table for implementing PIT is not reliable [55].	Uses DRAM, and perform poor w.r.to memory in comparison to NCE and CBF.
Li, et al. [65]	MBF	Sequential	Performs better than NCE and hash table in storage.	Uses SRAM to show whether *CN* is present or not.	The complete implementation consumes high memory.
Yuan, et al. [56]	64-bit CityHash	Hardware parallelism	Performs better for edge routers.	Uses SRAM or RLDRAM.	N/A

- ***PIT Placement.*** Although PIT data structures reduce memory consumption, during high link rates, the memory requirements can exceed the size of a single memory chip. So, there is a need to divide the PIT, one for each incoming/outgoing interface [54][55][56] of the router to handle user requests at the line speed. Dai, et al. [55] have proposed two PIT placements like, *output line-card*, and *input-output line-cards*. In output line-card, segregated PIT is placed at each outgoing interface. An *I_pkt* makes its entry at the output line-card where it is forwarded further (through FIB). In *input-output line-cards*, PIT is placed both at incoming and outgoing interfaces. While both the schemes have limited support for multi-path forwarding, the former does not at all support loop detection. Yuan, et al. [56] have proposed *input line-card* PIT placement where segregated PIT is placed at each incoming line-card. This scheme does not support loop detection and Interest aggregation and additionally, it needs to search whole PIT available at different incoming interfaces whenever a *D_pkt* arrives. Varvello, et al. [54] have introduced a third-party PIT placement where PIT is placed at each input-line-card. Whenever a request arrives for content *A*, a third party selects the PIT through hashing. There is no need of line-card at the output because same procedure is repeated for the *CN* of the *D_pkt*. This scheme supports multi-path forwarding, Interest aggregation and loop detection. However, it takes an extra switching for handling both *I_pkt* and *D_pkt*.

Name Prefixes	**Stale Time**	**Interfaces Ranked by Forwarding Policies**				
Name Prefix *N*	Stale time for *N*	Interface ID	Routing preference	RTT	Status	Rate limit

Fig. 18: Forwarding State in FIB [6]

B. Forwarding Information Base (FIB)

FIB is a table in an NDN router which helps in *I_pkt* forwarding. FIB maintains the following fields (Fig. 18) for each name prefix - *stale time*, *routing preference*, *RTT*, *status*, and *rate limit* [29][65]. These name prefixes are announced using NDN routing protocol and are added to FIB during the routing convergence time. Routing preferences are set by the routing protocol by using routing policy, static link metrics to path cost and network conditions. It helps ranking the interfaces in best to worst w.r.to data forwarding. A FIB also maintains the status (yellow, green, and red) of each interface for data retrieval, based on the RTT calculated through the PIT. Whenever a new interface is added to the router, FIB marks it yellow (signifying average performance), which changes into green (good performance) only when the data comes back through this interface. Otherwise, the interface's status is made red (poor performance). FIB periodically checks the status of each interface associated to a router through the *Interface probing*. The 'one *D_pkt* for each *I_pkt*' flow balance prevents congestion in NDN.

The data structure used to implement FIB must be scalable and efficient. Based on the underlying processing environment, we have broadly classified them into *CPU-based, GPU-based* and *FPGA-based* as shown in Fig. 16.

- ***CPU-based.*** They mainly use data structures, like trie [66], bloom filter [67], hash table with bloom filter [68], and trie with bloom filter [69]. Encoded Name Prefix Trie (ENPT) was the first name encoding scheme in FIB [66] where name components are encoded using a 32-bit integer. To further improve the lookup performance, they proposed a parallel architecture called Parallel Name Lookup (PNL) [70]. So, et al. [68] preferred to use a hash function and table with Bloom filter, and data pre-fetching for implementing the FIB. So, et al. [71][72] have also implemented a software forwarding engine for fast name lookup using hash table supporting fast collision-resistant hash computation. Another data structure that is popular for finding LPM in the IP's FIB is Bloom Filter (space-efficient probabilistic data structure) which consumes low memory and supports high lookup speed. But, it cannot be directly applied to NDN's FIB. Therefore, Wang, et al. [67] have proposed *NameFilter* - a two-stage Bloom filter-based scheme for fast longest name lookup in FIB. First stage maps the name prefixes into bloom filters using their lengths and it is used to find out

LPM of a name. Second stage divides the name prefixes into groups on the basis of their next-hop and finds out next-hop for the LPM calculated by the first stage. NameFilter performance mainly depends upon the distribution of the name prefix length and the number of interfaces/ports in the router. NameFilter outperforms Character Trie [73], ENPT [66], BloomHash [74], and Bloom Filter. Two main technical challenges to speed the LPM of name prefixes are (1) how to speed up the name prefixes' length calculation and (2) how to speed up the hash table operations. Wang, et al. [75][76] have proposed an adaptive greedy name lookup mechanism to handle the challenges. For handling the first challenge, it pre-computes the length of name prefixes of the FIB, and uses this distribution for performing LPM efficiently. For handling the second challenge, they proposed a new hash table which minimizes the computation time and memory. In trie based name lookup scheme, number of memory accesses are equivalent to length of a name while, bloom filter induces false positives. To minimize the negative impact of these data structures on the NDN performance, Quan, et al. [69] have proposed a name lookup engine, named as, *Adaptive Prefix Bloom filter* (NLAPB) which divides a name into two parts. The first part is B-prefix which is matched using Bloom Filter (BF) while the second part is T-suffix which is processed by Trie. NLAPB outperforms Character Trie [73], BloomHash [74], and hash table. Yuan, et al., [77] have used binary search of hash tables for reducing the lookup cost of the FIB. They have also proposed *level pulling* to reduce the number of hash lookup for LPM. Song, et al. [78] have proposed the use of binary Patricia trie to fit millions of names into fast memory of a line card. Song, et al. have also introduced *speculative forwarding* for core routers based on longest prefix classification (LPC). In LPC, packet is forwarded to the next-hop, either *CN* matches to FIB entries or not. Further, dual binary Patricia data structure is used for speculative forwarding where its size depends upon the number of rules instead of the length of the rules.

- ***GPU-based.*** When DRAM is used to store FIB, massively-parallel processing power of GPU is required to speed up the lookup time. The main data structures used for GPU-based FIB's are multi-stride aligned transition array [79], and trie [80] data structures. GPU-based name lookup is first introduced by Wang, et al. [79]. The proposed GPU-based name lookup scheme uses the Aligned Transition Array (ATA) which is changed into 4-stride Multi-stride-ATA (MATA) for reducing the number of memory accesses. Further, it is assumed that number of memory accesses can be decreased by storing the input name prefixes in an interleaved layout which is, named as, MATA-NW. MATA-NW outperforms 2D State Transition Table (STT), ATA, MATA, and MATA-NW. Wang, et al. [80] have implemented a GPU-based name lookup engine enabled with pipeline and CUDA multi-stream techniques. First of all, all the components of a name are encoded using 32-bit integer which is, named as, *global-code allocation mechanism*. But this solution consumes more memory. Therefore, to minimize the memory consumption, allocated codes of the components are locally optimized which is, named as, *local-code allocation mechanism*. In local encoding, all the components/transitions belonging to one node (Original Collision Set (OCS)) are allocated different codes and the identical components belonging to different OCS are allocated same code. This optimization minimizes the memory consumption by reducing the total number of allocated codes. The proposed scheme outperforms Traditional Character Trie (TCT) for network throughput, memory space, latency and name table update time. Perino, et al. [81] have proposed *prefix bloom filter* scheme using GPU for FIB. Authors have also proposed FIB placement on different interfaces of the router to reduce the lookup latency.
- ***FPGA-based.*** Li, et al. [82] have used Field-Programmable Gate Arrays (FPGA) to implement the hierarchical structure, Hierarchical Aligned Transition Arrays (HATA) for FIB.
- ***Miscellaneous.*** For analyzing, comparing and evaluating the different name lookup schemes, like character trie, component trie, and NCE, for the longest prefix match in the FIB, Zhang, et al. [83] have proposed a flexible platform, *NDNBench*. In order to enhance the performance of the NDN network, Rezazad, et al. [84] have proposed a memory architecture, named as, *NDN||mem* with content caching policy (CCndn) for matching the memory speed with the router's incoming request. The authors have assumed that FIB is slower than PIT and both modules are searched in parallel. For improving the performance of FIB, a new module FIB-cache, containing frequently accessed name prefixes, is integrated with the PIT table. The authors have also assumed that content is divided into chunks and routing of different chunks of a file remains same. Therefore, only first *I_pkt*'s routing information will be cached in the PIT and the rest *I_pkt* of a file should follow that. Another assumption on the basis of traffic measurement is that most of downloads are aborted in the middle. It can be assumed that starting chunks of a file are more popular than the end of a file. Therefore, starting chunks (or segments) should be cached near to the consumer and ending chunks should be cached near to the data producer (or, provider). Using this concept, authors have proposed CCndn caching policy that uses the hop distance (near to consumer or far) parameter to cache different chunks of a file in different routers between consumer and the producer (or, provider). FIB-cache and parallel search on both PIT and FIB reduces the searching time while CCndn minimizes the CS overhead by reducing the queue length. Whenever the number of non-aggregated name prefixes and named data increases in the NDN network, there is a need to store more name prefixes in the router's FIB which induces high control overhead and FIB explosion. To handle these problems of a routing scheme, Torres, et al. [85] have proposed a Controller-based Routing Scheme (CRoS) which runs on the top of NDN. CRoS uses NDN *I_pkt*/*D_pkt*, and supports NDN features, such as congestion control, network problem detection and path diversity. In this scheme, the main network routing issues like, mobility, security, network partition, and inter-domain routing, etc., are also discussed. In [86], Hsu, et al. proposed to use CRC-32 (Cyclic Redundancy Check) which encodes name prefix into 32-bit fixed length prefix to minimize the name lookup time in all three data structure of a router, i.e., CS, PIT, and

FIB. Afanasyev, et al. [87] have used the concept of map-and-encap for Secure NAmespace MaPping (SNAMP) as a solution of the NDN routing scalability issue (routing table sizes). So, et al. [88] have proposed forwarding and caching scheme with solid-state drives (SSD) so that both module, forwarding and caching can work together at high speed.

In Table IV, we have provided, in detail, experimental environments and performance, as discussed for different FIB's data structures.

TABLE IV PERFORMANCE OF THE FIB LOOKUP TECHNIQUES

Scheme Name	Data Structures	Algorithm Type	Performance	Pros	Cons
Wang, et al. [66][67]	Hash Table, trie, array	Sequential and parallel	Performs better than Name Character Trie (NCT) and Component Character Trie (CCT) in storage.	Trie minimizes the impact of redundant information at storage.	Uses STA. Whenever a new child is added there is need to shift child node(s) to other memory locations.
So, et al. [68]	Chained hash tables with linked list, and BF	Sequential	N/A	Use of hash table reduces name lookup cost per second.	Hash table induces high memory cost.
So, et al. [69][72]	Modified *SipHash*	Sequential	N/A	Use of hash table reduces name lookup cost per second.	Hash table induces high memory cost.
Wang, et al. [67]	CBF	Sequential	NameFilter outperforms Character Trie, NCE, BloomHash, and Bloom Filter.	N/A	NameFilter performance Depends upon prefix length and number of router interfaces.
Wang, et al. [75][76]	String oriented perfect hash-table	Sequential	Greedy name lookup mechanism with the improved string-oriented near-perfect hash table performs better for name lookup.	N/A	N/A
Quan, et al. [69]	CBF, Trie	Sequential	NLAPB outperforms Character trie, BloomHash, and hash table.	Minimizes the drawbacks of both data structures.	N/A
Song, et al. [78]	binary Patricia trie	Sequential	It outperforms [66], [67], [79], and [80]	Trie minimizes the impact of redundant information at storage.	N/A
Wang, et al. [79]	MATA	Parallel	MATA-NW outperforms 2D-STT, ATA, MATA for name lookup.	Uses DRAM. To increase name lookup per second uses massive parallelism.	Use of memory is not proper.
Wang, et al. [80]	Hash Table, Trie, Array	Parallel	Performs better than TCT.	Uses DRAM. To increase name lookup per second uses massive parallelism.	N/A

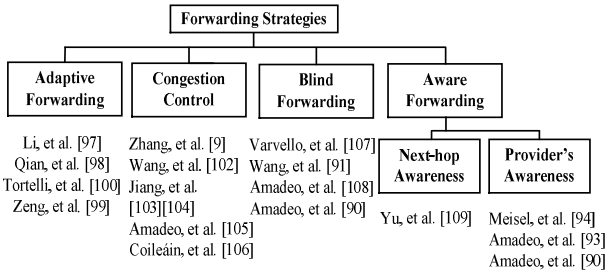

Fig. 19: Classification of Forwarding Strategies

4.3.2 Forwarding Strategies

Forwarding strategy in NDN decides how to use multiple forwarding options efficiently and choose the best interface(s) to forward the *I_pkt*. Design of a forwarding strategy depends upon network environment and context [29][65]. The key factors for designing a forwarding strategy are - *working path* (flooding or single best path), *context-based selection*, *on-demand multiple paths selection*, and *efficient interface(s) probing*. The three main questions arise for the NDN forwarding strategies [29][65] are (1) *how to discover working paths quickly and efficiently?* (2) *how to utilize multiple paths?* and (3) *how to probe unused interfaces?*There are some forwarding strategies used by the NDN forwarding plane [29][65].

- ***New Interest.*** Whenever a new *I_pkt* arrives at the router and its entry does not exist in the PIT, then a new entry is made in the PIT. If a green interface is available, then the *I_pkt* is forwarded to this interface. Else, it is forwarded to the available highest priority yellow interface and a retry timer is enabled. If a NACK comes from the upstream node before the timer expires, then remaining interfaces are tried to get the data. If there is no remaining low priority interface than the recently used one, a NACK is sent back to the downstream node with congestion code. NACK is used to notify downstream nodes that the requested content is not available. It is assumed in the NDN that NACK is beneficial and prevents from Interest flooding attacks. However, NACK have some drawbacks too as pointed out by Compagno, et al. [89]. They have categorized NACK into two *Content-NACKs* (cNACKs) (generated by content producer that requested content does not exist) and *Forwarding-NACKs* (fNACKs) (to inform downstream nodes that *I_pkt* cannot be satisfied).

fNACKs do not have any security issue while cNACKs are vulnerable to security attacks. In cNACK, adversary can send multiple closely spaced *I_pkt*s for the non-existent contents which may lead producer-focused DoS attack. As a countermeasure, Compagno, et al. [89] have proposed that for each *I_pkt* producer should check whether requested *I_pkt* CN is *plausible* (already exists or have possibility to exist in future), and non-existent. A cNACK will be generated iff *I_pkt* is plausible.

- ***Subsequent Interest.*** An *I_pkt* whose entry already exists in the PIT but *nonce* does not match, is called *subsequent Interest.* It may happen due to retransmission by the same content consumer and *I_pkt* may be generated by another user with the same name. If subsequent Interest arrives before the retry timer of the previous *I_pkt* expires, then the recent *I_pkt* will not be forwarded further because the router is already waiting for the matching content.
- ***Interest NACK.*** Whenever a router neither produce data for incoming *I_pkt* nor it is able to forward *I_pkt* upstream, then it sends *Interest NACK* to its downstream node. When an intermediate node receives Interest NACK, it forwards *I_pkt* to its remaining interfaces. When they also do not get corresponding *D_pkt*, an intermediate node sends NACK to the downstream node. The same procedure will be repeated until Interest NACK reaches to the consumer. Interest NACK will also maintain an error code like, *I_pkt* may be *duplicate*, network may be *congested* or *no data* is available for requested *CN*.
- ***Interface probing.*** For finding the alternative paths, FIB periodically sends probe and updates the interface status accordingly. If probing is pending, FIB chooses any available yellow interface and forwards the *I_pkt*.

We have broadly classified forwarding strategies into *adaptive forwarding*, *congestion control*, *blind forwarding* (BLF), and *aware forwarding* as shown in Fig. 19. Using the forwarding state information, each router can calculate packet delivery performance (i.e., RTT and throughput), link failures or network congestion, use multi-paths and maintain status of each outgoing interface. Thus, NDN forwarding decisions are adaptive to network conditions.

Congestion control strategies are used to control traffic entry in the network for increasing the link capabilities. Blind forwarding strategy [90][91] is a counter-based broadcasting scheme where *I_pkt* and *D_pkt* transmission is deferred using some policy to minimize collision probability. Also, it uses packet overhearing for minimizing the *I_pkt* redundancy. When a node *N* receives *I_pkt*, it listens to the channel for some time period. If it overhears ongoing broadcast/existence of a duplicate *I_pkt*, it drops the packet from the PIT. This strategy can minimize the effect of broadcast storm. Aware forwarding [90] uses some additional information (exchange of some packets [92] and/or some more fields in *I_pkt* / *D_pkt* [93][94][95][96]) of the content producer (or, provider) and/or neighborhood for making the forwarding decisions.

A. Adaptive Forwarding

Adaptive forwarding has been addressed by researchers using optimization techniques [97][98]to strike a balance between several performance metrics, and context-awareness to facilitate route adaptation [99][100]. Li, et al. [97] have proposed a Quality of Service aware Greedy Ant Colony Forwarding (GACF) algorithm for reducing the complexities associated with complex NDN networks (i.e., link failure, congestion, and frequent topology changes). It uses ISP-based aggregation to minimize the content naming space. It uses Ant Colony Optimization (ACO) algorithm as NDN forwarding for supporting the Quality of Service (QoS). GACF uses two types of ants for optimizing the routing and forwarding plane of the NDN. *Normal Interest Ant* uses greedy approach to retrieve data while, *Hello Interest Ant* collects routing and forwarding information, i.e., path overhead, bandwidth, RTT, number of hops, etc. However, GACF adaptively minimizes the impact of link failures, network congestion, and frequent changes in the network topology. Qian, et al. [98] have proposed a probability-based adaptive forwarding (PAF) scheme which also uses the ACO technique to calculate the selection probability of each interface for load balance among interfaces, minimizing delay, and detecting changes in network conditions (such as, network topology, network status, traffic pattern, traffic load, and content popularity). A statistical model is also proposed to calculate the packet retransmission timeout. PAF is adaptive and can be tuned on (or, to) different network conditions. Zeng, et al. [99] have also proposed rank-based routing strategy (RBRS), which assigns rank to each interface for route selection. In this scheme, cumulative use of each outgoing interface is calculated using FIB. Mostly used outgoing interfaces are assigned to low rank while, less used interfaces (less busy) have high rank. For forwarding an *I_pkt*, high ranked interface is preferred. RBRS reduces the number of access hops and the node load. Tortelli, et al. [100] have compared and evaluated three forwarding strategies *flooding*, *best-route with caching*, and *best-route without caching* under different traffic conditions. In flooding, incoming *I_pkt*s are forwarded to all outgoing interfaces except the incoming one. In best-route with caching, Dijkstra's algorithm is used to find best paths (minimum number of hops) among all available outgoing interfaces. In best-route without caching, in-network caching is removed and repositories are used for satisfying the consumer's request. Best-route without caching reduces file download time, and hit distance, while best-route with caching improves hit ratio. Flooding reduces overall communication overhead. Schneider, et al. [101] have discussed the simultaneous use of multiple access networks to support better QoS. They proposed three forwarding strategies: *lowest cost strategy* (LCS), *multiple attribute decision making* (MADM), and *selective parallel*. In LCS, three parameters (1) maximal packet loss, (2) maximal delay, and (3) minimal bandwidth are used to determine a interface (to forward *I_pkt*) with low cost and satisfying all three requirements according to the specific application. MADM is based on the cost-awareness and considers two parameters: maximum possible *QoE* of any application and minimum *QoS* for interface selection. Selective

parallel strategy uses same parameters used by LCS but if one of the requirement is not met, it simultaneously forwards *I_pkt(s)* to multiple interfaces until these requirements are satisfied by atleast one interface.

B. Congestion Control

Researchers have proposed congestion control schemes to regulate *I_pkt*s flow using Interest shaping schemes [102][103][104], and TCP's Additive increase/multiplicative decrease (AIMD) [8][105] to reduce network load and increase link utilization. Wang, et al. [102] have proposed a hop-by-hop Interest shaping congestion control mechanism for maximizing link utilization with no data loss. In Interest shaping algorithm, it cannot be assumed that the whole reverse path will remain dedicated to only *D_pkt*s. For an Interest shaping, there is a need to regulate the flow of *I_pkt*s so that enough *I_pkt*s may move in the reverse direction too. Therefore, congestion control through Interest shaping algorithms prefers to drop *I_pkt*s earlier instead of dropping *D_pkt*s. The authors have also shown that hop-by-hop Interest shaping algorithm is not enough to control congestion of the entire network. For this, the authors have proposed to use drop-tail policy at the Interest queue of the shaper and then send an Interest NACK to downstream nodes. Interest hop-by-hop shaping algorithm also facilitates congestion-aware re-routing for high cost paths and supports backpressure mechanism for NDN. Interest shaping algorithm performs well under the conditions of load and congestion. Jiang, et al. [103][104]have also proposed a scheme to regulate the *I_pkt*s flow. In this scheme, all *I_pkt*s associated to the same flow are aggregated into one packet named as, *Interest Set* which reduces the overall network overhead and round trip delay. It also reduces the number of FIB lookups and PIT size.

Most of the NDN congestion control schemes [29][65] have major limitations. The proposed solutions assume that network and traffic properties like, size of *D_pkt*s, the time required to retrieve *D_pkt*s, and bandwidth of each link, are known earlier. To overcome these problems, Dynamic Interest Limiting [8] is proposed which uses TCP's AIMD algorithm in NDN and can adjust the *I_pkt* limit of each interface dynamically. When *D_pkt* is received, the *I_pkt* limit increases, and when congestion is detected, *I_pkt* limit decreases. A Random Early NACK (REN) mechanism is also proposed for handling the queue length at a node to send explicit congestion notification to its downstream node. REN is not suitable for NDN overlay because it is difficult to manage the queues of underlying IP routers. Therefore, Link-layer Congestion Detection (LCD) is proposed where each router adds a link-layer header to each NDN *D_pkt*. At the other end, congestion can be quantitatively detected by the number of missing *D_pkt*s using each *D_pkt*'s sequence number. Jointly REN and LCD perform better than their individual performances. DIL also supports fairness at routers by distributing the *I_pkt* limit among several name prefixes. To enhance the performance of AIMD based transport strategies for wireless ad hoc network environments, Amadeo, et al. [105] have introduced a transport scheme, self-regulating Interest rate control (SIRC) for handling *I_pkt* transmission and retransmission. *I_pkt* transmission is analyzed by consumer on the basis of the inter-arrivals of *D_pkt*s, which indirectly indicates network bandwidth. *I_pkt* retransmission is based on the timeout at the consumer and controls the effect of content-multi-homing. SIRC is a robust feedback mechanism compared to timeout-based AIMD schemes. SIRC minimizes the effects of RTT / Re-transmission Timeout (RTO) fluctuations, freezing in dynamic network because of content multi-homing. A comparative analysis is done with respect to one Interest per RTT (issue an Interest iff, previous Interest either expired or satisfied) and timeout controlled AIMD strategies (window-based flow control, when the RTO expires, it halves the Interest Window) applicable in wired networks, where SIRC performs better for data retrieval. Coileáin, et al. [106] have introduced SAVANT framework for NDN to provide content feedback (from consumer to content provider to monitor future or ongoing operations) and accountability (to generate verifiable information about the content distribution process for finding responsible entity during any problem).

C. Blind Forwarding

Researchers have proposed to control the flooding in the broadcast medium, i.e., packet collision, using channel overhearing and random delay in *I_pkt* transmission [107], collision-avoidance timer [91], and deferring timers for *I_pkt*s and *D_pkt*s transmission [108]. In NDN, flooding is preferred in the wireless ad hoc environment (having intermittent connectivity), whereas flooding suffers from packet collisions in the broadcast medium. The possible solution is listening to the channel for passing packets and to abort the scheduled transmission if duplicate packets are detected. When a node wants to send a packet, it delays transmission by a random amount of time and drops the packet if it could overhear a same packet in the radio channel. Varvello et al., [107] have shown that this type of controlled flooding is mainly used in topology-based routing in MANETs, where the cost of maintaining topology information may diminish the impact of structured solutions. Wang, et al. [91] have used set of defer timers to control the packet redundancy in vehicular networking. Besides this, when neighbor cars receive same Interests, they use a collision-avoidance timer to disseminate traffic jam information. For providing the priority to *D_pkt*s over *I_pkt*s, Amadeo, et al. [108] preferred to set different defer timers for *I_pkt*s and *D_pkt*s. If a node *N* overhears the channel and finds out that another node has transmitted same data multiple times, then *N* drops *D_pkt* instead of forwarding. A blind forwarding [90], is similar to [91] and [108], which uses a set of different defer timers for minimizing the collision probability and supports overhearing to control *I_pkt* redundancy. Still, blind forwarding does not always guarantee that the selected node will be best for packet forwarding and does not ensure that *overhearing* will completely avoid packet collisions. The blind forwarding design is based on the packet overhearing and defers a timer which is not suitable to handle broadcast storm. To handle this problem and enhance the packet delivery performance, *provider-aware* forwarding design is proposed which appends provider's information in *I_pkt*s and *D_pkt*s.

D. Aware Forwarding

Aware forwarding [90] can be classified as *provider awareness* and *next-hop awareness* as shown in Fig. 19. In provider awareness forwarding (PAF), consumer discovers content provider(s) and selects best among them using some measures while, next-hop forwarding scheme, *I_pkt* is broadcasted to one hop neighbors and a node is selected as relay for further packet forwarding on the basis of direction and distance.

- ***Provider Awareness.*** Some research works have implemented an efficient provider aware forwarding using the hop distance [90][93][94] to minimize the content access latency in the wireless ad hoc network. PAF is based upon Listen First, Broadcast Later (LFBL) [94] and Enhanced-Content-centric multihop wireless NETwork (E-CHANET) [93] protocols. LFBL was basically designed for general wireless multi-hop networks with data-centric addressing. In LFBL, consumer selects provider using distance between consumer and content provider(s). It uses Distance Table (DT) which maintains *CN*, the provider identifier (ID) and the hop distance between consumer and communication end-point. It uses three types of packets: *Request*, *Response* and *Acknowledgement* (Ack). The data retrieval takes three steps: First is, request dissemination using flooding for selecting the data provider. Second is, response having information about available data provider(s). Third is, Ack from consumer to the selected (on the basis of distance) provider. E-CHANET is also a distance-based forwarding scheme like, LFBL but it does not use any explicit Ack. PAF [90] is implemented on top of the BLF where it floods the first *I_pkt* of the content using BLF. Whenever any provider gets this request, it appends its own ID and parameter *N* initialized to 1, with the *D_pkt*. Each intermediate node in the downstream increases *N* by 1 and forwards further. When a consumer receives a *D_pkt* corresponding to its request, it extracts additional information available in the *D_pkt* and maintains in the DT. Due to multi-path forwarding, multiple providers can response simultaneously. Then, for the same remaining contents, consumers will send requests to the nearest provider using minimum hop distance.
- ***Next-hop Awareness.*** Next-hop awareness forwarding has been addressed using both, *distance* and *I_pkt*s *incoming rate* [109] to select a relay node from the consumer's neighbors. Yu, et al. [109] have proposed a Neighborhood-Aware Interest Forwarding (NAIF) scheme for reducing the flooding overhead in which relay node is selected on the basis of data retrieval rate for a given name prefix and its distance from the consumer. An *I_pkt* is discarded either if the data retrieval rate is low or the distance between the consumer and the node is high. NAIF is compared with other Named Data MANETs (NDM) forwarding design, i.e., NDN Forwarding (NDNF), and LFBL. LFBL implicitly selects a path between consumer and provider. NDNF and NAIF perform better in multi-consumer data retrieval scenarios than LFBL, and NAIF reduces bandwidth usage in compared to NDNF. Performance of BLF and PAF are computed under different network conditions, node density and topologies. In case of one-to many (multiple users are interested in same content) scenario, BLF outperform PAF while, in terms of efficiency, and total number of *I_pkt* and *D_pkt* traversed in the network, PAF outperform BLF.

4.4 Security, Privacy and Trust

Content security is one of the most important basic requirements of any data centric networking [110]. The key challenges of NDN security are cost-effective security operations, trust management, and privacy protection.

- ***Security.*** In NDN, every *D_pkt* is signed with standard public key, so that, anyone can verify its authenticity [10]. The main requirements of trusted content are: *data integrity*, *origin authentication*, and *relevance of the data* (*against the requested one*). Producer selects an appropriate signature algorithm (like, RSA [111], DSA [112], or EC-DSA [113]) from a large fixed set to minimize the size of verifiable data, latency of verification, and computational cost of signature generation and verification [10][110]. The process of signature verification may have multiple rounds of certificates fetching and verification. Therefore, the possible solution is to cache the validated certificates required for verification which can be used till their expiry [10]. Each signed *D_pkt* keeps information of the public key, necessary for verification. Other information associated with the signed *D_pkt* is *public key's cryptographic digest*, or *fingerprint* and *key locator*, which shows the location of that key.

  NDN uses content *encryption* to support *access control* and *confidentiality*. However, there is no need of trusted servers or directories to implement access control policies. Also, there is no need to distribute decryption key as it can be passed along with the content. NDN supports both asymmetric and symmetric encryption. Content encrypted with the public key can only be decrypted using private key. If content is encrypted with symmetric keys, then there is a need to have a secret key which can be fetched from a signed *I_pkt*. A data producer can verify the consumer through its signature and returns encrypted secret key using a consumer's public key (visible to consumer only). Shang, et al. [114] have addressed that NDN packets are either encrypted or signed using symmetric key of two communicating parties. *Access control* is needed to differentiate between legitimate and malicious users. It supports producers to publish content under any namespace and consumers to access any content as long as they have the valid key. Hamdane, et al. [115] have modified an existing access control technique, called UCONABC [116], for optimal and secure data-centric access control in NDN. UCONABC is used to define access rights management over NDN. In this model, lock password and encryption are used for protecting data where access is centrally controlled by Access Control List (ACL). The proposed idea allows legitimate readers/writers to unlock/access secured data. Massawe, et al. [117] have proposed a Bloom-filter based Scalable and Privacy Preserving Routing Protocol for providing security

and privacy to NDN *I_pkt*s. Moreover, multicast key management protocol is also proposed which allows authorized users to access keywords of the content (known as multicast encryption). A content-dependent key tree is used for key distribution. Yu, et al. [118] have proposed an endorsement-based key management system based on the Web-of-Trust (WoT) to provide security to *ChronoChat* [119]. In this, a user can authenticate each other's membership, and other user's identity in the chat room, without any external public key infrastructure.

- ***Trust management.*** It is used to authenticate a given key for a particular packet in an application [8]. Pournaghshband, et al. [120] have allowed a user to securely retrieve the public key of content producer (K_p) from its own community of trust (C_T: individuals, the user personally knows in the real world). Whenever a *D_pkt* arrives at user, U, and K_p is invalid or not cached, then U asks for K_p in its C_T. The responses from C_T are validated using local policy. Moreover, a key revocation approach is also introduced for this model to do away with old keys which either expires or gets revoked. Application of trust management using NDN is Secure *Audio Conference Tool* (ACT) [121][122], lighting control system, building automation (BAS) and networked control systems (SCADA) [123][124][114][125] [127]. Yu, et al. [127] developed a set of trust schemas for several NDN applications with different trust models of varying complexity.
- ***Privacy.*** In NDN, there is need to encrypt both content and its name for providing the privacy, e.g., as ANDaNA [133] do. In the current Internet, through IP addresses, and payload, it is easy to guess "*who is consuming what content*". In NDN, through naming and caching, it is easy to track "*what data is requested*", but without destination address, it is very tough to know "*who has requested the content*" (unless there is any direct connection). NDN routers stores user's requests and are vulnerable to an attacker. Therefore, NDN requires privacy [7][8] at the router level.

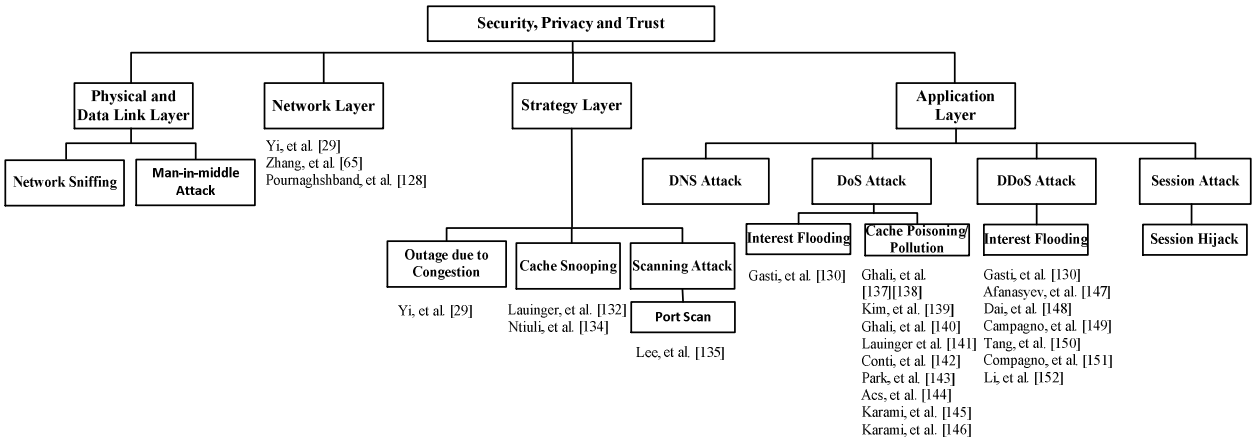

Fig. 20: Security Attacks and Their Countermeasures

4.4.1 Physical & Data Link Layer

We have discussed two major attacks at the physical and data link layer: *network sniffing* and *man-in-middle attack*. When the data is transferred in the network without encryption, it can be captured by the application which is known as network sniffing. In man-in-middle attack, an attacker may secretly change the content between two communicating parties. As a countermeasure for these attacks, NDN design supports an encryption mechanism similar to the TCP/IP architecture.

4.4.2 Network Layer

The main attack possible at the network layer is *prefix hijack*. In a prefix hijack attack [29][65][128], a malicious autonomous system (AS) advertises invalid prefixes. This corrupts the routing information and other existing AS starts forwarding their requests to this malicious AS. This is also known as *black-holing* as no request will be fulfilled by the malicious routers. *Black-hole prefix hijack* [29][65][129] is less effective in NDN. As its symmetric communication nature allows to keep track of unsatisfied *I_pkt*s at the PIT. These *I_pkt*s can be used to know which *I_pkt* is hijacked. Forwarding strategies at the router level also keep information about the performance of each interface (interface status) with respect to name prefix. Another feature like, multi-path forwarding also increases the probability of forwarding the *I_pkt* through different path and decreases the effect of this attack. Moreover, to completely eliminate the prefix hijack, routing updates must be cryptographically authentic [130]. For this, a mechanism for securing named data can be used to prevent from prefix hijack on NDN.

Each router's name can have a public key provided by the network operator. Now, each router's interface can also have an interface key signed by the router's public key. All routers' updated information will be signed by the interface key for supporting the authentication to the update information [131].

4.4.3 Strategy Layer

The main attack possible in the strategy layer is due to naming the contents. Someone in the middle of the communication can easily track the communicating parties.

A. Outage due to Congestion

To control the congestion, NDN routers use rate limit [29]. If congestion exists, then in-network caching reduces the overhead of handling the re-transmitted packets.

B. Cache snooping

Cache snooping is the process in which victim's cache is continuously probed to know the requests of the victim. Even, data in the intermediate routers remain encrypted, it is easy to learn about other information like, *CN*, size, etc. Cache snooping attacks may be of three types [132], such as, a *cache monitoring attack* (monitoring of victim's cache existing in the same access router), *object discovery attack* (discover objects in the victim's cache in the same access router), and *flow cloning attack* (to know the complete data flow in victim's cache). For cache monitoring attack prevention, one-time names can be used to make a *CN* unpredictable. Some tunneling mechanisms like, ANDaNA [133], can be used. Scope field of the *I_pkt* can be disabled to hide the cache hits. Moreover, some delay can be added for forwarding the cached data. In object discovery attack, the adversary can use the exclusion filter and prefix matching for getting the information of other contents in the victim's cache. For handling object discovery attack, exclusion filter can be disabled and a number of name prefix matching should be limited. In flow cloning attack, an adversary may know about the name of the requested content through object discovery attack and then, predict the next incoming *I_pkt*'s name. The possible solution for this attack is to encrypt the *CN*. If encryption is not possible, then the name should be difficult to understand. In [134], the authors have proposed an effective and efficient mechanism to identify cache snooping for low-level NDN routers.

C. Scanning Attack

One type of the scanning attack is *port scan*. In port scanning, an attacker sent message to each and every port on a system and then, wait for a response which returns some information like, port status, for attacks [135][129]. In NDN, for targeting any host, there is a need to target its namespace. The use of port is specific in NDN, to the applications as it supports different names for different services. Scanning of all possible names is very time consuming. Therefore, there is need of little attention to design the mechanisms, which can decide whether to answer an *I_pkt* due to scanning attack.

4.4.4 Application Layer

In the current Internet, the most common attacks are *denial of service* (DoS) and *distributed denial of service* (DDoS) [130] [136]. It is required for any new Internet architecture design to be robust for the existing DoS and DDoS attacks. In a DoS attack, only one computer system and one Internet connection are used to flood a server while in a DDoS attack, many computer systems distributed in location, are used to flood the network. We have identified attacks in application layer into the *Domain Name System* (DNS) attack, *DoS* attack and *DDoS* attack as shown in Fig. 20.

A. DNS Attack

In DNS attack, an attacker re-route the incoming traffic to a specified server. DNS attack is not possible in NDN. Re-routing the path between source and receiver, in NDN, does not make any effect if received content is valid.

B. DoS Attack

Gasti, et al. [130] have studied and explored NDN in-built security features which resists the NDN from certain attacks, such as *reflection attacks*, *bandwidth depletion*, *cache poisoning* (discussed below), and *black-hole prefix hijack*. In reflection attack, adversary uses another host IP to make attack which is not possible in NDN due its symmetric communication. In bandwidth depletion, adversary floods the victim's node with requests for already existing content. The effect of this attack is limited in NDN because after some time requested content will be available in the intermediate routers' caches. Interests for the same content will be satisfied earlier via cached copies. The two main DoS attack that target the contents are *content/cache poisoning* (i.e., overloading of malicious content into caches) and *cache pollution* (i.e., disrupting cache locality and false locality). We have discussed three new NDN-specific DoS attacks, such as *interest flooding*, *content/cache poisoning* and *cache pollution* with a set of countermeasures.

- ***Interest Flooding and Countermeasure.*** Interest flooding attacks can be further classified based on the type of content requested: (1) static, (2) dynamic, and (3) non-existent or fake (i.e., unsatisfiable interests). In these cases, adversary uses zombies to flood the targeted servers. Attack (1) and (3) are intended for *network infrastructure* while, due to (2), both *network and application-layer* functionalities suffer. NDN is resilient to both (1) and (3) attack. Attack (1) is similar to bandwidth depletion. In attack (3), zombies may issue unsatisfiable *I_pkts* to targeted servers. This attack targets the routers because these *I_pkts* will consume PIT space until they expire. NDN automatically alleviates this type of attack through interest aggregation at the router. In (2), adversary uses the *PIT states* for carrying out the DoS attacks. In this type of attack, the adversary can fill the PIT with fake requests and/or flood the specific content producer(s). However, in NDN, *I_pkt*s do not contain source address and are not secure. It is very difficult to know the adversary's origin. If the request is for dynamic content, it will be handled by content producer only (no support for caching). Zombie at adversary-side will issue *I_pkt* for victim server and the server will waste its resources to fulfil the zombie's request instead of requests from legitimate users. The possible countermeasure is to keep track of unsatisfied *I_pkt* at the PIT.

- ***Content/Cache Poisoning.*** An attacker can overload the cache through the corrupted (invalid signature) or fake (valid signature but wrong private key) *D_pkt*s. A consumer may easily verify the signature and identify cache poisoning. This feature introduces two main challenges, such as, signature verification overhead and trust management. The authors have also focused for strong binding of *I_pkt* with its corresponding *D_pkt* and proposed countermeasure based on heuristics, inter-router communication and user feedback. In [137], authors have discussed about NDN's elements of trust and some optimization techniques for reducing the signature verification overhead. It mainly focuses towards the countermeasures of content poisoning attacks in NDN. The authors have also presented some trust management goals for supporting the content authentication in NDN router where Interest-Key Binding (IKB) rule is implemented for consumers, providers and NDN routers for checking authenticity of received data. One of the effective content poisoning countermeasures is light-weight statistical content-ranking algorithm [138] for cached content in the NDN router which uses content ranking to differentiate between good (valid) and bad (fake) contents. The ranking of content is based on the statistics computed at the consumer-end through the delivery of content. Each cached-content is ranked between 0 and 1. Initially 1 (highest) is assigned to each content which decreases gradually with the time. Rank of any content is decided by how many times it has been excluded (exclusion filter) by the consumers in their requests. A content having more number of recent exclusions is ranked lower. In order to prevent from the cache poisoning, Kim, et al. [139] have proposed an effective solution based on content verification for valid contents with less computational cost. Ghali, et al. [140] have focussed the attention that due to signature verification overhead and low trust in public keys for verification, content signature verification is optional in NDN. This creates high possibility of content poisoning attacks.
- ***Cache Pollution.*** Two countermeasures, *naive* and *selective* have been proposed [141] against cache-based privacy attacks. Naïve countermeasure further can be categorized as *detecting an attack* and *preventing an attack*. Challenges in detecting the cache pollution attacks are that attackers may change their behaviour and routers have less processing capability. To prevent data from the attacker, the *hop count* feature could be disabled and some *delay* can be added for fetching the cached data to give an impression that cache is shared by many users. *Selective countermeasures* show that content available in the network can be classified as *sensitive* and *non-sensitive* (not private). Selective countermeasure is also classified as *selective caching* on the basis of *content popularity* and *selective tunnelling* for sensitive content. Content popularity may depend on the context in which it is used and the location. E.g., a video popular in one area may not be useful for other geographically separated areas. Therefore, selective caching is proposed to cache popular contents, for maximizing the cache performance. On the other hand, sensitive contents always have low popularity because of their nature. Selective tunnelling uses enabled flag bit for the sensitive content only which will not be cached in the network. This feature will enable a better security and will enhance network performance. In [142], authors have considered a new type of cache pollution attack in NDN where the adversary attempts to interrupt cache locality for legitimate users. They have proposed a countermeasure for detecting pollution attacks on different topologies which performs better than the existing [143] one. To guarantee the cache privacy in NDN, Acs, et al. [144] have categorized data into *privacy-sensitive* and *non-sensitive*. For sensitive data, provider puts one extra bit in the content header and consumer enables one bit in Interests. Karami, et al. [145] have proposed a cache replacement scheme using Adaptive Neuro-Fuzzy Inference System (ANFIS) to alleviate the cache pollution attacks in NDN.

In [146], Karami, et al. proposed a hybrid algorithm using Particle Swarm Optimization (PSO) with the radial basis function (RBF) neural network to proactively detect and mitigate DoS attacks.

C. DDoS Attack

DDoS attacks on NDN have been studied by Gasti, et al. [130] and Afanasyev, et al. [147] who have shown that per-packet state update and symmetric nature of NDN communication mitigate the effect of the DDoS attack. They have proposed four countermeasures - (1) *token bucket with per interface fairness*, (2) *intelligent attack mitigation*, (3) *satisfaction-based Interest acceptance*, and (4) *satisfaction-based pushback*. First approach modifies the token bucket of packet switched network and guarantees that next hop will get a fair mix of *I_pkt*s. Second approach uses the statistics of Interest satisfaction ratios (current flow of outgoing *I_pkt* and incoming *D_pkt*s) to make decisions for forwarding incoming *I_pkt*. Third approach uses the Interest satisfaction ratio based on direct probability to accept or reject the incoming *I_pkt*. In fourth approach, each router announces its Interest limit for each incoming interface to their downstream router(s). Evaluation showed that the fourth approach outperforms the rest. Dai, et al. [148] have also used symmetric nature of NDN to mitigate the effect of DDoS attack and proposed Interest traceback scheme which can trace the origin of the *I_pkt*. However, the scheme cannot identify *I_pkts* having fake names. Compagno, et al. [149] have proposed a scheme to track consumers and eavesdroppers within the network. Tang, et al. [150] have classified Interest flooding attack in *FIB-based* and *Broadcast-based* Interest flooding. In FIB-based Interest flooding, a particular name prefix is targeted and many times requests are forwarded for same name prefix from adversary side. This does not only exhaust PIT but also, wastes FIB resources. In broadcast based Interest flooding, adversary selects a non-existent name prefix for *I_pkt* and floods it. A *two-phase detection scheme* has been proposed to track the normal name prefixes used in Interest flooding. The two phases are *rough detection phase* and *accurate identification phase*. In rough detection phase, NDN symmetric nature is used to mitigate the attack effect. If the number of *I_pkt* and number of *D_pkt* varies over time, it shows the presence of abnormal name prefixes. In accurate identification phase, each router interface is monitored for incoming *I_pkt*s and expired *I_pkt*s to identify the

abnormal name prefixes. Another proposed countermeasure against the Interest flooding DDoS attack on NDN network is *Poseidon* [151]. Poseidon uses both local metric and collaborative techniques for detecting the Interest flooding at the early stage. Li, et al. [152] have proposed an application-based countermeasure, *Interest Cash* for interest flooding to handle dynamic contents. In Interest Cash, the consumer is required to solve the static meta-puzzle created by the content provider before it sends an *I_pkt*. So, the adversary will consume more resources than a consumer.

D. Session Attack (Session Hijack)

Due to data-centric nature of NDN, many existing Internet-based applications, such as ACT, live streaming, and vehicle-to-vehicle communication, on the Internet can be implemented without maintaining sessions. On the other hand, NDN application such as, ANDaNA [133] uses sessions. But still, in NDN these attacks are less effective as security is applied at data level instead of channel.

TABLE V VARIOUS SECURITY ATTACKS IN NDN AND THEIR POSSIBLE COUNTERMEASURES

Layers	Different Attacks in NDN	Attack Resilience	Possible Countermeasures for NDN
Physical and Data Link Layer	Network Sniffing	Yes	Both content and its name are encrypted in NDN.
	Man-in-middle Attack	Yes	
Network Layer	Black-hole through Prefix Hijack	Limited	*Symmetric* nature of NDN communication.
Strategy Layer	Outage due to Congestion	Limited	NDN forwarding-*rate limit* feature.
	Cache Snooping		
	Cache Monitoring Attack	Limited	Through unpredictable *CN*s [132].
			Tunnelling mechanism [132].
			Through a disabled scope field of *I_pkt* [132].
			Delay for forwarding the cache data [132].
	Object Discovery Attack	Limited	Disable Exclusion filter [132].
			Limit the number of name prefix [132].
	Flow Cloning Attack	Yes	Encrypt the *CN* [132].
Application Layer	***DoS Attack***		
	Interest Flooding	Limited	To keep track of unsatisfied *I_pkts* in the PIT [130].
	Content/Cache Poisoning	Limited	Content authentication [136].
			Content ranking [137].
	Cache Pollution	Limited	Naïve Countermeasure [141] • Disable hop count • Add delay for cached data
			Selective Countermeasure [141] • Selective caching • Selective tunnelling
	DDoS Attack		
	Interest Flooding	Limited	Token bucket with per interface fairness [142].
			Intelligent attack mitigation [142].
			Satisfaction-based Interest acceptance [142].
			Satisfaction-based pushback [142].
			FIB-based Interest flooding [143].
			Broadcast-based Interest flooding [143].
	Detection of Interest Flooding at Early Stage	Limited	Poseidon [145].
	Interest Flooding for Dynamic Contents	Limited	Interest Cash [146].
	Session Attack		
	Scanning Attack	Yes	Scanning all possible names in NDN is difficult.

E. Others

Hamdane, et al. [153] have proposed PKI and HIBC integrated scheme, to improve the security and explore the naming system in NDN. Min, et al. [154] have proposed a new format of *I_pkt* and *D_pkt* well-matched with present crypto technologies and embedded with the modern security communication system CMS. Kusunoki, et al. [155] have proposed, a PKI-based two-way content management scheme based on content copyright violation in NDN. This scheme shows that if copyrighted content illegally exists in NDN, then the authentic owner can inform to supervisor to delete content from the whole network. The supervisor does the same by checking the authenticity of owner's request. Paknezhad, et al. [129] have presented a comprehensive study on security and privacy for cloud services in NDN.

In Table V, we have summarized the possible security attacks and their countermeasures on NDN on a layer-by-layer basis.

4.5 Mobility

In IP-based environment, devices require an IP address (to communicate with other devices) for every networking interface they use (or for topological change) and do not guarantee the continuation of an ongoing connection. Therefore, for mobile devices, communication is not feasible until they acquire a new IP address every time when there is a change in location. For handling this problem, proposed solutions are Mobile IP [156] and Host Identification Protocol (HIP) [157], but they do not directly deal with content mobility issue. These solutions rely on topological information and on indirection points for traffic redirection [158]. However, NDN supports data access by *CN*s instead of IP addresses. This enables mobile users to have better data access as there is no need for them to repeatedly acquire an IP address and they may continue their communication during the content flow or unavailability of published content. Below, we summarize the basic differences between IP-mobility and NDN mobility issues [159].

- ***Host Multi-homing.*** In the Internet, individual applications are required to establish individual connections with multiple network interfaces (Bluetooth, Wi-Fi, etc.). Each interface needs a different IP address to facilitate communication and every time there is a need to maintain TCP connection. Therefore, IP address assignment to each interface makes switching difficult between the network interfaces. NDN supports the naming. Therefore, NDN applications can explore multiple network interfaces by sending requests to all interfaces at the same time.
- ***Network Address Consistency.*** In IP, routing protocols must maintain consistent routing tables in the network, which induces high overhead in large scale network, e.g., slow convergence in BGP and limited scalability in OSPF.
- ***Removal of Connection Oriented Sessions.*** Whenever a node moves in the current Internet, there is need to re-establish TCP connection for resuming the communication. However, for communication, NDN does not establish a connection to the data source. Also, NDN does not maintain sessions.
- ***Scoping of What and Where.*** Consumers should not be recognized by their location. E.g., BBC iPlayer service can only be accessed using UK IP addresses which do not support mobility of legitimate UK consumers. In NDN, content is accessed on the basis of *CN* and does not bind any consumer to the location.
- ***Resilience through Replication.*** In IP, if host identification fails during connection establishment or any intermediate router fails during data transfer, there will be no data. While in NDN, an intermediate router caches the requested data and fulfils the request if have valid content.

Meisel, et al. [160] were the first to argue that current Internet architecture and its protocols are not suitable for the highly mobile environment like, Mobile Ad-hoc Network (MANET). The authors then discussed the main failing reasons of IP-routing based approach w.r.to mobility as we have already discussed earlier. In order to solve these problems, some opportunistic IP-routing based protocols were also proposed for taking the full advantage of wireless broadcast nature. But still, IP address assignment to the mobile node and their management problems were remaining. A new solution DTN was introduced where data units are framed within bundles and allows late binding of data names to node IP address. But still, each bundle is delivered using the traditional IP addresses.

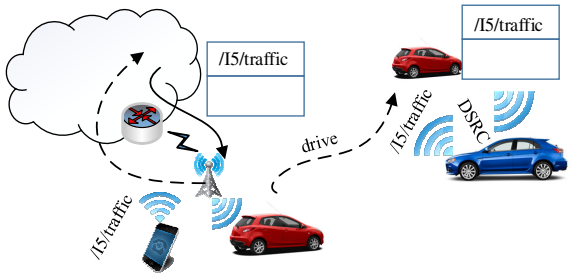

Fig. 21: An Example of DTN and Connected Networks [161]

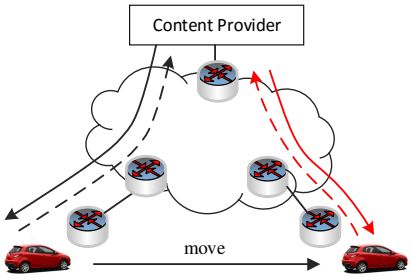

Fig. 22: Data Flow Back to a Mobile Consumer along Reverse Path of *I_pkt* [161]

NDN architecture design supports the DTN and MANET data delivery while in the current Internet, these are separate sets of solutions for highly dynamic network [160][161]. Moreover, the NDN special feature is that, devices may communicate if physically they are in range of each other or along device-to-device paths without having any Internet Service Provider (ISP), which is not present in TCP/IP [162]. To analyze how mobile nodes communicate under different connectivity conditions in NDN, a scenario is presented in the Fig. 21. Suppose a mobile phone requests traffic related information using Wi-Fi where mobile-nodes / vehicles near to that mobile phone may overheard reply and opportunistically cache the content for future requests for the same. Whenever these mobile nodes / vehicles encounter to other vehicles requesting the same traffic information, even without any wireless coverage using Dedicated Short Range Communications (DSRC) [163], may serve them through cached data.

NDN supports the user-side mobility without any extra efforts through in-network caching. It supports the smooth hand-off because if consumer moved to another location, then requested content will be cached in-between routers as shown in Fig. 22. While consumer mobility is supported inherently by NDN, support for content producer mobility is absent [161][164][165].

Azgin, et al. [164][165] proposed a dynamic framework to gauge the impact of producer/ consumer mobility on the network performance. Two main NDN forwarding strategies used are *flooding* and *smart-flooding*. In flooding, *I_pkt*s are forwarded to all active interfaces, whereas in smart-flooding, *I_pkt*s are forwarded to all the interfaces iff there is no green interface. There is a third forwarding strategy, called semi-flooding, which is a hybrid of the previous two and it works based on the nature of the network which performs flooding in access networks and smart-flooding in others. The authors proposed to use Constant Bit Rate (CBR) traffic model which is good for modeling delay sensitive as well as delay tolerant traffics. In delay sensitive traffic, *I_pkts* are forwarded at a fixed interval while delay tolerant traffic is opportunistic, i.e., whenever a node gets a network connection, it forwards the data. The authors have analyzed the impact of mobility in NDN with respect to *intra-* and *inter-AS mobility*.

In intra-AS mobility, flooding induces higher overhead, but it still works well for mobile nodes. However, smart flooding and semi-flooding do not perform well in the case of mobility. Flooding performs better for producer mobility whereas smart/semi flooding performs better for consumer mobility. Producer mobility is having higher overhead compared to consumer mobility, because in the latter case, data will be available within a few hops due to in-network caching. But, in producer mobility, overhead will increase due to Interest flooding done by all intermediate nodes. In inter-AS mobility, producer mobility performance degrades; while, consumer mobility performance remains constant. In producer mobility, flooding overhead will be more, as there is no common router(s) in between the producer and the consumer. Mobility in data consumer means to send *I_pkt*s before the hand-off and to retrieve corresponding data after hand-off. In data producer mobility, there is need to reach the *I_pkt* to data producer new location.

Another main issue is to support smooth hand-off for mobile nodes moving fast in NDN. Therefore, we have broadly classified NDN's mobility into *producer's mobility* and *hand-off management*, as shown in Fig. 23.

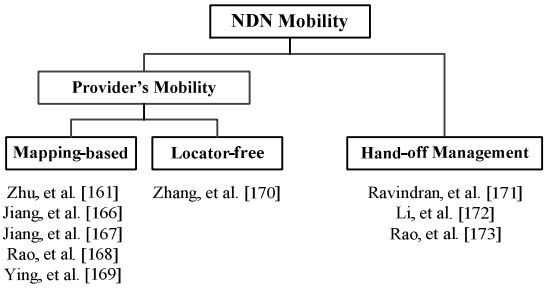

Fig. 23: Classification of NDN Mobility Techniques

4.5.1 Provider's Mobility

Content provider's mobility in NDN is supported through a mapping-based technique which is aimed at providing mobile content providers current location against specific user query [161]. This technique works following the well-known locator/identifier split principle adopted in IP mobility support. Since, a namespace can always be associated with a particular content provider, we can use the namespace as the stable identifier of that provider even if it moves to a different location. Similarly, the name prefix of the access router can be used as the location of the mobile content provider. For tracking the mobile content provider's location, either NDN broadcast support or intermediate DNS servers can be used. In NDN broadcast, *I_pkt* can be broadcasted to all the requested user's neighbors. If a broadcast is not available then, mobile content provider may update its current location to the intermediate DNS. NDN is broadcast in nature and works well in wireless LAN (broadcast-nature communication media). If multiple consumers want same data, then only one *I_pkt* is transmitted which provides data to all multiple consumers at the same time. When the broadcast feature is not available, DNS mapping between identifier and locator can be used. Some researchers have proposed to use mapping technique like, DNS-based mapping between identifier and locator as discussed earlier, while some have proposed locator-free approaches for supporting provider's mobility in the NDN. We further divided provider's mobility into *mapping-based* and *locator-free* as shown in Fig. 23.

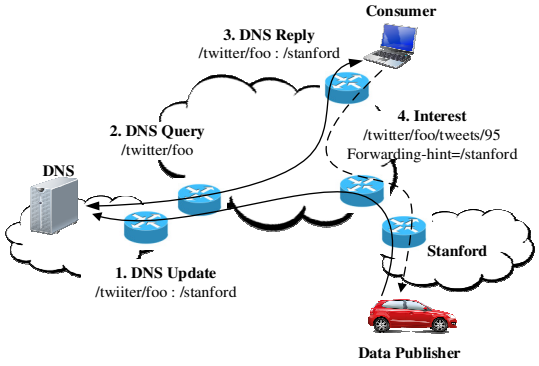

Fig. 24: DNS-Based Mapping for Data Fetching [161]

A. Mapping-based Approach

Mapping based approach has been widely adopted as a solution of provider's mobility using DNS mapping (forwarding hint) [161][166][167], and unique locator [168][169].

Zhu, et al. [161] have presented a DNS-based solution to achieve seamless mobility support for content providers as shown in Fig. 24. Intermediate DNS servers can be used to maintain the current location information of moving data provider for its supported name prefix. If any user wants to access data from the mobile content provider, then the user may ask to the intermediate DNS server for updated location of the intended content provider. The provided location information can be used as a forwarding hint with the *I_pkt CN*, to retrieve the data from the mobile content provider. In forwarding hint, a routable prefix is added in the *I_pkt* as a hint and it is used for forwarding an *I_pkt* if no name prefix match exists in the FIB.

Jiang, et al. [166] have extended the concept of forwarding hint to support the content provider mobility and presented a depth analysis to show that only popular content get profits from caching. The authors have also explored that the benefits of the NDN architecture depends upon the application nature like, retransmission, on-demand or instant. Further, Jiang, et al. [167] highlighted the problem (situation of triangle routing) associated with DNS mapping based scheme used for data provider mobility support in NDN. The *forwarding hint* can contain any name prefix of the content provider in place of ISP prefix. As a mobile node moves to another ISP, name prefix announcement can be distributed and local which support the scalability and minimizes the name prefix update time.

The scope of the locator also plays an important role. Whenever *I_pkt* moves to new AS, the ingress router may change the locator on the basis of AS strategies. If AS links are overloaded, then other paths may be used for *I_pkt* forwarding. If the ingress router of AS knows about the nearest cached copy, then it may forward *I_pkt* towards the cached node. In Novel Locator Based Approach (NBLA) [168][169] each access router is assigned with unique locator and with some additional features and functionalities to support provider mobility. An optional field is also included in NDN *D_pkt* and the original FIB outgoing interface entries are modified to keep track of mobility status and location of the provider. This helps the consumer to avoid the situation of sending the *I_pkt* to the old location of the provider or re-issuing the *I_pkt* to the new location of the provider. NBLA minimizes the handover cost and latency compared to existing provider's mobility support approaches.

B. Locator-free Approach

Zhang, et al. [170] have proposed provider's mobility support using PIT states (Interest's states) as trace to reach mobile nodes, called '*Kite*'. Kite supports *location freeness* and *scenario-awareness*. In case of location freeness, the location of a mobile node, MN, is made transparent to all the nodes communicating with MN. This is achieved by exploiting the forwarding strategy of NDN. Scenario awareness gives the flexibility to the application programs for mobility support specific to their scenarios/requirements.

In location freeness, Kite support two types of mobility: *direct Kite* and *indirect Kite*. In direct Kite, *MN* establishes a path with an immobile corresponding node (CoN) by issuing the traced Interest (leaves the trace back to a *MN*), which allows *CoN* to send an Interest back along the trace. In indirect Kite, *MN* sends Interest to an immobile trace anchor (application-specific). Then, *CoN* (mobile or immobile) can send tracing Interest (traverse along the trace of the previous Interest) towards the *MN* to the anchor, using FIB which ensures reachability of anchor and using PIT trace which guarantees reachability of *MN*. In scenario-awareness, authors have presented five mobile application specific scenarios: upload the data on the server, download data from the server, pull data from *MN*, push data to a *MN*, and share data. The Upload scenario is supported by direct Kite and Push / Pull application scenario is supported by indirect Kite.

Authors made some assumptions that Kite is designed to support infrastructure-based networks for trace generation and maintenance. Also, long-term relocation of the content provider is not supported. When the *MN* moves frequently, Kite shortens the path stretch (ratio of the actual path length to the shortest path length between the *MN* and the *CoN*), minimizes delay in compared to mapping-based mobility solutions. Kite increases the uploading rate for the *Upload* application-specific

scenario and minimizes traffic generation in *Share* scenario. Kite also provides a platform to design new NDN application protocols.

4.5.2 Hand-off Management

Researchers have assessed mobility support during hand-off process for real time (RT) applications [171] and delay-sensitive (DS) applications [172]. Proactive caching mechanism [173] is also proposed to minimize the access latency of content during handover process.

Ravindran, et al. [171] have introduced three schemes to support seamless mobility in RT NDN applications based on following four components – (1) *mobility agent* (MA) associated with individual mobile nodes (MN) identified with unique mobile IDs, (2) *network proxy agent* (NPA) associated with each *Point of Attachment* (PoA) (router through which the node is connected to the network), (3) *rendezvous point agent* (RA) (which is a variant of NPA and exists in all other non-PoA router(s)), and (4) *mobility controller* (which maps MNs to their current locations). Moreover, the authors have proposed three schemes to minimize the *I_pkt* and Data loss during a handoff of consumer, data provider, and both.

The first scheme is based on *Point of Attachment* (PoA). In this scheme, the *MN* initially registers itself to a PoA through the NPA. Whenever a *MN* starts moving towards another PoA, its MA will de-register it from the old PoA and will register with the new PoA.

The second scheme is *Rendezvous Point* (RP) which is a higher level aggregation router for handling the *MN* associated with multiple PoA(s). In this scheme, mapping of a *MN* to an RP is maintained by the mobility controller. *MN* registers itself to a PoA and the PoA chooses the RP to handle the seamless mobility of that *MN*.

The third scheme is based on *multicast* and it uses - multi-path Interest routing (MIR) and multi-point content (prefix) publishing (MCP: which supports provider's mobility through publishing prefixes from several points in the network). In this scheme, the NPA does not maintain the *MN* states while NPA collaborate with all PoA's NPA for supporting MIR and MCP. This scheme reduces complexity associated with forwarding state, but performs poor for PIT state overhead and transient FIB entries.

Li, et al. [172] have introduced a novel mobility support scheme for delay sensitive applications in NDN using the Session Initiation Protocol (SIP). SIP is used for making voice and video calls in the IP application layer. In this paper, the authors have advocated the use of SIP for supporting mobility in Voice-over Content-Centric Networking (VoCCN) which is an implementation of VoIP over NDN and is proposed by Van Jacobson, et al. [174]. In SIP, INVITE message is used to start the session and message body contains information about the session description and URL of end host. VoCCN has implemented SIP into the NDN network layer by including the INVITE message into the *I_pkt*. During an ongoing voice communication process between two hosts *M* and *N*, if *M* moves from one access router to another, then there will be a change in the current name prefix. *M* informs *N* about its new name prefix by sending a modified re-INVITE message as *I_pkt*. Two approaches have been introduced by the authors as enhancements of their proposed SIP-based mobility support scheme for NDN discussed above. The first such scheme aims to reduce the hand-off latency, by assigning name prefix before a link layer hand-off takes place. This approach is similar to the Fast Handoff scheme for Mobile IPv6 [175] and it uses L2 triggers initiated by the link-layer specific events, such as link signal strength. The second enhancement of the proposed scheme is proxy-assisted network based technique for fast hand-off.

Rao, et al. [173] have also proposed Proactive Caching approach for NDN (PCNDN) for supporting the user mobility during the handover process. Each access router supports some additional features like, pre-fetching and caching the contents before the user handover. PCNDN reduces handover cost, minimizes handover latency, and increases packet delivery ratio.

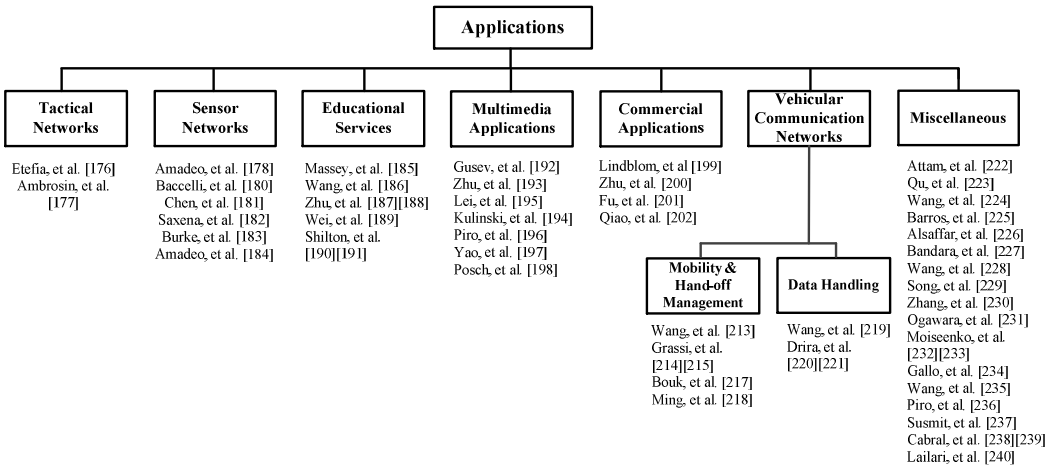

Fig. 25: Classification of NDN Applications

5. APPLICATIONS

In this Section, we shall survey different NDN applications to understand how several architectural and functional features of NDN can benefit networking applications. We shall discuss various naming schemes with respect to different types of applications. We shall also study how the NDN functional features like, multi-path forwarding, security, name data routing, scalable forwarding, in-network caching, infrastructure-less mobility, etc., are supporting NDN applications and providing them advantages over the current Internet. We have studied and found out the applications of NDN spread across a wide variety of application domains, such as *Smart System Development* (Building Automation System (BAS)), *Wireless Sensor Networks* (WSN), *vehicular communication*, *multimedia systems*, *conferencing and educational systems*, *entertainment* and *gaming*, to name a few as shown in Fig. 25.

5.1 Tactical Networks

Tactical networks are an ad-hoc communication network which connects soldiers and military equipments in the battlefield with the military headquarters. Due to IP-based communication, tactical networks face challenges in connectivity and suffer from limited bandwidth and high latency. In order to improve challenged military/tactical communication networks efficiently, authors in [176], have proposed an NDN network model and analyzed its performance. Ambrosin, et al. have presented the covert ephemeral communication (CEC) [177] which allows two or more nodes to exchange covert ephemeral messages (CEM). These CEM becomes disabled after a time limit. CEC is implemented on network layer instead of application layer which gets to this scheme robust, reliable and suited for tightly-controlled environments like, military. To exchange the CEM among multiple receivers, a new robust technique, Common-Prefix-based Covert Communication (CPC) using NDN longest prefix matching is also proposed.

5.2 Sensor Networks

A WSN is composed of sensor nodes, which are deployed densely, prone to failures, and are resource-constrained. WSN works in a data-centric manner as queries are addressed with required data. Moreover, in a WSN, unique addressing is not possible as a sensor node does not have global identification (global-ID) because of high overhead and the presence of a large number of sensor nodes. The data-centric nature of WSN and the absence of global-ID shows that NDN and its features can be easily applied to WSN to increase overall system performance. Some of the NDN benefits over WSN [178] are fast data retrieval as NDN uses hierarchical naming, scalability as NDN inherently supports broadcast communication, caching, easy application development, and flexible deployment.

Any physical object can be attached to the sensors and may seamlessly sense, communicate and share the data. Smart sensors can be placed anywhere and can be controlled either locally or remotely in various applications like, smart healthcare, home automation or smart home, smart environment, etc. This computing concept is known as Internet of Things (IoT). The benefits of NDN-IoT are manifold: First, NDN uses *CN*, e.g., */ndn/iitr.ac.in/apps/healthcare/CSE/IS_lab* for applications which remain consistent across facilities and installations (helps in network setup and maintenance). Naming data provide inherent support for delay tolerant networking (handles intermittent connectivity) unlike the Internet. Moreover, in IoT, as heterogeneous devices are being used, NDN naming can support access restrictions instead of having a separate policy language for individual devices [6]. Second, NDN does not face *address exhaustion*, *address assignment* and *management* problems because it uses unbounded namespaces. Third, NDN reduces the complexity of auto-configuration mechanisms compared to layered network architecture [180]. Fourth, NDN supports in-network caching, which may reduce power consumption (saves energy) and usage of radio resources depending on the caching strategy adopted. It further supports load balancing and fault tolerance and thus helps resource-constrained devices. Fifth, NDN supports multi-path routing which provides much sought-after communication reliability for safety-critical systems. Sixth, NDN has in-built support for user-side mobility which will help mobile nodes or devices to keep communicating with the server. Seventh, NDN inherently provides confidentiality, authenticity and integrity by signing each content which is of utmost importance of securing named data. Eighth, NDN uses a trust model to authorize an application and authenticate each and every command (for fixtures) for performing required operations in the application which enhance the privacy, security and reliability of the system. In support of these benefits, the authors [179] have implemented a lighting control system using NDN architecture to solve the problems associated with existing Building Management System (BMS) like, configuration management and security. Baccelli, et al. [180] have done experiments to explore the challenges, features and advantages of NDN in IoT systems. In [181], Chen, et al. have presented a cloud-assisted Wireless Body Area Network (WBAN) for supporting healthcare services using NDN. Saxena, et al. [182] have proposed *SmartHealth-NDNoT*, an NDN-based smart healthcare IoT, or NDNoT, for remote monitoring and diagnosis of patient's healthcare and wellness. The authors have also proposed naming conventions to locate the healthcare services. Briante, et al. [183] have proposed a framework *namEd Data netwOrking for hoMe-USer interaction* (eDomus), which uses the popular social network, i.e., Facebook (FB) functionalities to allow a user to remotely control the home network. Recently, Amadeo, et al. [184] have proposed multi-source data retrieval in NDN-IoT based systems.

5.3 Educational Services

The main aim of these services is to present a basic introduction of the network architecture development over the time. The major applications are to set up the educational networks like, conference rooms and to provide ad-doc communication

during meetings, conferences, and lectures. For better understanding of two network architecture developments, IP-based networking and data-centric networking, a detailed study is presented in [185].

In [186], Wang et al. have discussed NDN major features like, data-centric dissemination, request-reply model, multi-cast data delivery, and security at data level to design the conference applications. As an instance, a voice conference system is developed for studying the real benefits of NDN architecture over the current Internet. The authors have also discussed the challenges associated with the voice conference system and their solutions through NDN. Zhu, et al. [187] have implemented a distributed and robust Audio Conference Tool (ACT) which have done away with any central controller unlike conventional online audio conversation applications. ACT introduces high scalability and support for mobility. In [188][189], Extensible Messaging and Presence Protocol (XMPP) enabled Multi-User Chat (MUC) based two conference tools (audio and whiteboard) are developed over CCNx. Shilton, et al. [190][191] have presented the social values associated with NDN design, its technical contents and data privacy and anonymity.

5.4 Multimedia Applications

Teleconferencing is a most popular multimedia application, e.g., Video-conference [192], tele-teaching, tele-presentation, tele-musical rehearsal, video phone and audio chat. A distributed, server-less, multi-user chat application, *Chronos* [193], is implemented over NDN. Chronos can work with any chat client supporting XMPP. It allows a participant to share a hashed collection of names with other participants in the same chat room.

A live video streaming application, *NDNVideo* [194] for NDN architecture is also implemented using the GStreamer open-source media framework and PyCCN (Python bindings for the CCNx software router). The main design goals are to provide live and pre-recorded video to multiple users with random access, synchronize playback of multiple consumers, no session semantics or negotiation and security. This approach could be used to enable server-less video publishing from resource-constrained mobile devices. Lei, et al. [195] have proposed NDN-Hippo's control layer based on the IP-based P2P media streaming system. Hippo media system have two independent layers: *control layer* responsible for system operations and *media layer* responsible for data streaming.

In [196], an NDN-based architecture for supporting the crowd-sourced and real-time media contents is implemented. The authors carried out extensive experiments to show the effect of various factors like, chunk loss ratio, the percentage of interest received by providers, Peak Signal-to-Noise Ratio (PSNR) w.r.to number of available video contents for maintaining QoS for end users. The four main components of this architecture are user groups for event content share, user group interested in the shared content, distributed Event Management System, and NDN communication infrastructure. In [197], long-term Interest packet design is presented to efficiently and effectively deal with the real time data in NDN network. In this mechanism, an *I_pkt* for real time applications will have longer lifetime to reduce the number of *I_pkt*s and PIT size. This solution also supports mobility by re-issuing the *I_pkt*s from the new access router. Posch, et al. [198] have shown the challenges associated with the use of MPEG-DASH for client-driven adaptive streaming in NDN. To handle the issues of previous approach like, client starvation (clients do not know how many users concurrently watching the same video and their received video quality), authors have proposed in-network adaptation based on scalable content which will be developed in future.

5.5 Commercial Applications

The main services supported under these applications are electronic payments from anywhere, consistency maintenance among different datasets, file hosting services for file synchronization, vehicle communication like, data collection from mobile vehicles, data dissemination, e.g., news, road conditions, accident information, weather, etc., among vehicles.

In FileSync [199], a P2P file sync using NDN is proposed. Compared with existing file synchronization methods like, Dropbox, which is based on client-server model and depends on a centralized server, FileSync uses a P2P approach based on a distributed file sharing model. In order to reduce the traffic in IP-based P2P approach, NDN is employed for file synchronization which uses CCNx-SYNC synchronization protocol for maintaining the consistency of content among NDN network nodes. It also ensures security, version control and conflict resolution. In [200], a robust and distributed *ChronoSync* protocol is proposed to synchronize the state of a dataset among a distributed group of users in the NDN networks. It encodes the state of the data set into the digest to exchange the state of digests among all parties in a synchronization group whenever a change occurs in the dataset. Fu, et al. [201] have presented, a high performance synchronization protocol, *iSync* which supports efficient data reconciliation by representing the synchronized datasets using a two-level invertible Bloom filter (IBF) structure.

NDNBrowser [202] is implemented to support wide variety of NDN-based web applications like, e-government, e-health, e-commerce and social networking services using the open source WebKit. The already existing NDN.JS [203][204] based on JavaScript can run only on the TCP/IP networks, and does not support development of extensive web applications. While, NDNBrowser supports of HTTP/URL to guarantee that it can run at the same time on the NDN networks, TCP/IP networks, or NDN and TCP/IP hybrid networks. NDNBrowser is able to support for the HTML, images, CSS, JavaScript, audio/video tags in HTML5, JavaScript-based AJAX and developer-friendly API.

5.6 Vehicular Communication Networks

Vehicular networks have received significant attention in the last few years [205], especially with the advent of the DTN concept [206][207]. Recently, researchers are keen to investigate the feasibility of employing a content-centric approach to improve the efficiency of vehicular networks [208][209][210]. Amadeo, et al. [211] have proposed a content-centric framework for vehicular networks, which is implemented on top of IEEE 802.11p standard. Navigo [212], a location based packet forwarding mechanism for vehicular Named Data Networking (NDN).

5.6.1 Mobility and Hand-off Management in Vehicular Network

In recent years, vehicular NDN is exponentially growing for efficiently providing mobility support in the infrastructure-less networks [213][214][215][216] [217].

Wang, et al. [213] have proposed a mobility solution with respect to a practical NDN application in the V2V data dissemination system. Every vehicle has an embedded NDN forwarding engine which communicates with its peers and road-side units (also NDN embedded) through multiple network interfaces, such as 802.11p (DSRC/ Wireless Access in Vehicular Environments (WAVE) and Wi-Fi, to gather traffic information. Moreover, the NDN forwarding engine will send the *I_pkt* to other vehicles using one hop broadcast through available network interfaces. There are three different roles identified for a vehicle in terms of data *producer*, *consumer*, and *data mule*. Data producers are equipped with camera, sensors, and safety systems for producing road and traffic events. In comparison to other smart devices, cars usually have higher storage capacity. A data mule pre-fetches data which might be useful for consumers in the future. It caches data items by overhearing. They can serve even when the original data producer has left. The authors have also proposed a suitable naming scheme for the V2V NDN application. For effectively fetching the required data from other vehicles, names should include some key parameters like, geographical area, time, application type, and nonce, i.e., /traffic/geolocation/ timestamp/data type/nonce.

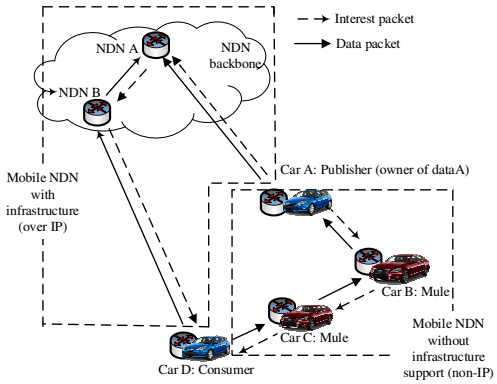

Fig 26: V-NDN [214]

Grassi, et al. [214][215] have proposed Vehicular NDN (V-NDN), a common vehicular NDN framework for both infrastructure and infrastructure-less networks to upload and retrieve data. A vehicle can support many wireless interfaces, such as ad-hoc, 3G/LTE, Wi-Fi, WiMAX, etc. Using these interfaces, a vehicle may communicate to other vehicles or servers by choosing the best interface according to the need of applications (Fig. 26).

A car can use infrastructure support (3G/WiMAX/Wi-Fi) to communicate with NDN routers. As in Fig. 26, car D communicates with car B using cellular networks by establishing an IP tunnel over NDN routers, which is an example of using infrastructure support. Also, car D can communicate with car C locally without any infrastructure support (using Wi-Fi ad-hoc /802.11p). In such cases, the *I_pkts* are forwarded hop-by-hop until it reaches the car having the requested content. This gives the flexibility to a car in the V-NDN to play one role at a time, producer, consumer, forwarder and data mule (physical carriers of packets).

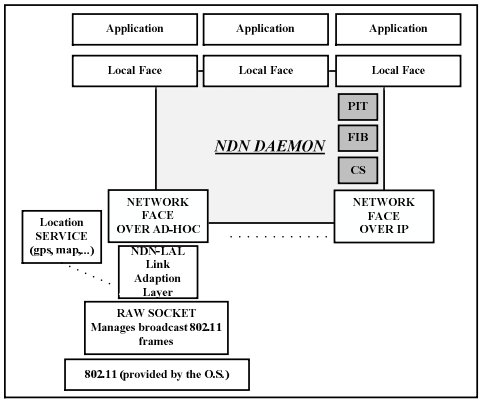

Fig 27: V-NDN Architecture [215]

The main components of the V-NDN architecture is *NDN Daemon*, *NDN Local face*, *NDN-Link Adaptation Layer (LAL)*, *Location Service*, and *Cache* as shown in Fig. 27. *NDN Daemon* provides main NDN functionalities like, naming and forwarding decisions and contains FIB, PIT and CS. *NDN Local face* works as a bridge between applications and NDN daemon. *NDN Network Face* provides adoption functions, coupled with the technology needed for communication. *LAL* is equivalent to layer-2.5 which takes maximum benefits from layer-2. *Location Services* provides the support for the application and LAL. *Cache* is associated with each node to cache the content. Authors have implemented V-NDN on ndnSIM [216] where simulation shows that when high number of cars requests the same content, the overall performance of the system improves. As later, consumers play the role of data mule and distribute the data efficiently in the network.

Ming, et al. [218] have carried out empirical studies to compare handover events in NDN and IP networks with respect to fast moving trains. They have introduced relative delay coefficient (RDC) as a metric to compare relative performance of NDN and mobile IP. For accessing the first packet after the handover, NDN performs better than mobile-IP even though there will be a cache miss. The reason may be that, handover in mobile-IP requires MNs to first register its IP with the new access point, which incurs significant handover latency. While in NDN, named data does not require any prior registration. For regular packets, NDN performs poor in comparison to mobile IP if no or little cache hit is there. In NDN, there is a need to send a request first for retrieving the required data while IP continuously retrieve data from the servers.

5.6.2 Data Handling in Vehicular Networks

Recently, some mechanisms for data collection in vehicular network, have been studied in the NDN using naming [219], Pub/Sub integration [220], and query's response aggregation [221], to reduce the number of messages required for data collection.

DMND [219] is a vehicle data collection system from MNs which take care of intermittent connectivity, data security and scalability in NDN. The simulation performed in Qualnet shows that DMND achieves better request-reply ratio compared to traditional Mobile IP. Drira, et al. [220] have proposed Pub/Sub integration with NDN for supporting data collection and dissemination in V2X (Vehicle-to Vehicle/Infrastructure) networks. Since V2X communication is event-based and NDN is a request-reply, integration of event-based Pub/Sub results in reduction of messages in the network because Pub/Sub allows the freedom for the subscribers to receive many messages for one subscription without sending the *I_pkt* again. Simulations on ndnSIM show that the method reduces the number of messages dispersed for subscribers in vehicles. Drira, et al. [221] have also presented a distributed query mechanism, named as NDN-Q for reactive data collection in vehicular networks. NDN-Q is used to collect and aggregate data from vehicles distributed in the network. NDN-Q is divided into two parts: query dissemination and response collection and aggregation. When consumer needs any data, it broadcast the Interest and any vehicle having the requested data may respond. Several vehicles may response together and increases unnecessary traffic volume. So to minimize the number of *D_pkts* transferred, data are aggregated or reduced by the intermediate nodes and towards the query source.

5.7 Miscellaneous

Many researchers have studied the use of NDN in different other domains, such as home and enterprise networking [222], gaming [222][224][225] and entertainment [226], multi-sensor collaborative sensing [227], instant messaging [228], traffic control for power saving [229][230], emergency communication support during disaster [231], developing communication APIs [232][233], Inter-process communication (IPC) [234], handling malicious nodes [235], smart city [236], climate research and high energy and nuclear physics (HEP) [237], etc.

Several researchers have worked on developing handy tools for experimentation over NDN platform. Cabral, et al. [238][239] have introduced a cost effective Mini-CCNx tool developed for different experiments on NDN architecture and evaluated using dynamic routing protocols (OSPFN). Moreover, it is user friendly to configure the various parameters and supports large simulation environments. Lailar, et al. [240] have presented emulated NDN testbed in Open Network Laboratory (ONL).

6. COMPARISONS BETWEEN INTERNET/IP AND NDN

Researchers of FIA [9] have analyzed some fundamental limitations of the current Internet architecture in terms of functional, structural, and performance related properties [15] and we have summarized them in Table VI. For the sake of easy understanding, the limitations have been classified into four distinct categories related to *processing and failure handling*, *storage*, *transmission*, and *control*. In processing and failure handling, we discuss the limitations associated with the processing of data at the routers and handling (or access of) the data. In storage, we analyze the limitations with the buffers, caches, memory and disks, etc., available at the network. In transmission, we point to the limitations associated with the transfer and exchange of data. Finally, control discusses the limitations associated with the controls required for processing, storage and transmission functions.

Table VII shows the inherent limitations of the Internet with respect to support for content dissemination, user mobility, network security, content caching capability, and application deployment. All these problems cannot be completely fixed with an ad hoc manner. Therefore, there is need of a new Internet paradigm which can address the drawbacks of the current Internet and root out the challenges with simple and efficient solutions.

In Internet, packets identify the end host using the destination IP address, while NDN packets refer the data using the *CN*. IP routers use IP prefixes for forwarding the IP packets and NDN routers use name prefixes for forwarding the *I_pkt*. IP depends on the routing protocol to provide a loop-free paths whereas, in NDN, request packet name and its nonce entry information is enough to eliminate the looping of the packets.

IP forwarding is stateless because, packets are not recorded in the routing table. But, NDN forwarding is stateful [7][28][65] because NDN router keeps record of every *I_pkt* passing through them, till the lifetime of the *I_pkt* expires. NDN observes the data plane (it is the part of a network that carries user traffic) performance continuously through the PIT states, Interest NACK and RTT calculation while IP cannot observe data plane performance because of one-way traffic and stateless forwarding. Internet routing protocols handle failures in packet forwarding. Therefore, routing protocols must maintain consistent routing tables in the network, which induces high overhead in large scale network [6], e.g., slow convergence in BGP and limited scalability in OSPF. However, in NDN, forwarding strategies handle the failures locally through periodically probing the router's outgoing interfaces.

TABLE VI FUNDAMENTAL LIMITATIONS OF THE CURRENT INTERNET ARCHITECTURE

Limitations	Classification	Internet	NDN
Processing and Failure Handling	Support for Internet problems diagnosis	During failure, a host cannot identify what happened, why it happened and what are the required actions.	√
	Methods for handling of network and systems infrastructure and essential services	Lack of proper services for these infrastructures, e.g., healthcare, transportation, agreement with legal regulations, etc.	√
	Data and identity service	Lack of proper content property rights which leads unfair charging model.	√
Storage	Context/content aware storage management	Data information is available at the end-points, but not during data transfer. Therefore, it does not guarantee fast storage management, fast data mining and retrieval.	√
	Inherited user and data privacy	Encrypted data cannot be stored efficiently while unencrypted data sacrifices user and data privacy.	√
	Data integrity, reliability, provenance, and trust	Loss of integrity due to failures and attacks.	√
	Efficient caching (on-path)	Still, several cache servers will request the same document from the original site of publication.	√
Transmission	Efficient transmission of content-oriented traffic	Content Distribution Networks (CDN) reduces the traffic by providing a distributed cache, but still cannot meet Internet scale.	√
	Security requirements of the transmission links	Communication privacy is not only protecting the data exchanged, but also not disclosing that communication took place.	√
	Security of the whole Internet architecture	Security in the Internet is provided through several add-ons. Protocols may be secure, but the overall architecture is not self-protected against malicious attacks.	√
Control	Support for mobility	There is a need to acquire an IP address for each wireless interface and every time the user changes location.	√
	Efficient congestion control	Existing congestion control schemes work through cooperation of end systems and the network which induces plenty of overhead.	√

In Internet, IP packets may follow any path to reach the destination which may lead to congestion in a dynamic environment. Therefore, there is a need to explicitly maintain congestion control schemes. While in NDN, router manages congestion by regulating the *I_pkt* rate by allowing a single *D_pkt* against every *I_pkt*. Network security in the Internet is ensured by providing a secure channel between two communicating hosts while in NDN, security is in-built in the architecture. Every *D_pkt* in NDN is cryptographically signed by the content provider with its name for securing the data.

Internet does not support caching of the requested data in the network, whereas, NDN supports in-network caching of the requested data.

TABLE VII FUNCTIONAL DIFFERENCES BETWEEN INTERNET AND NDN [7]

Features		Internet	NDN
Addressing		IP addresses	Named data
Routing	**FIB**	• IP prefixes • Contains single best next-hop • contains nothing but the next-hop information	• Name prefixes • contains a ranked list of multiple interfaces • FIB entry records information from both routing and forwarding planes.
		Forwarding is stateless	Forwarding is stateful
	Loop-free Path	Routing protocol	Name + Nonce
Forwarding and their Strategies	**State Maintenance**	Stateless forwarding	Stateful forwarding
	Failure in Packet Forwarding	Consistent routing table	Probe messages
	Congestion Control	Explicit congestion control schemes	FIB-rate limit
Security		Channel	Data
In-network Caching		Not supported	Supported

Some of the data transfer functions [158][232] on the basis of layers are discussed in Table VIII. Data transfer functions can be divided into two parts: data manipulation and transfer control. Data manipulation functions are used to read or modify the data, whereas, transfer control operations are used to regulate the data transfer process itself. These operations are associated with different layers in the Internet architecture while the same operations may associate with separate libraries and architectural modules in NDN. The layers responsible for performing these operations are discussed in Table VIII.

TABLE VIII COMPARISONS OF INTERNET AND NDN FOR DATA MANIPULATION AND DATA TRANSFER [232]

Data Manipulation	Internet	NDN
Error Detection	Link and transport layer.	Link and network layer.
Buffering for Retransmission	Transport layer protocols use the send and receive buffers in the kernel-space.	NDN uses the *Application level framing*, send and receive buffers are done in the application layer.
Encryption	Session layer (TLS/SSL).	It uses the user specific libraries in the network layer.
Presentation Formatting	Presentation layer.	*D_pkt* naming.
Transfer Control	**Internet**	**NDN**
Detecting Network Transmission	No	Yes
Acknowledgement	Uses ACK.	*D_pkt* itself.
Flow/Congestion Control	Uses transport protocols.	*D_pkt* follows the reverse path of *I_pkt*.
Multiplexing	Uses process-to-process multiplexing/de-multiplexing between two socket endpoints.	*I_pkt* or *D_pkt*s are de-multiplexed through the *CN*.
Timestamp & Sequence Number	Transport layer.	Network layer.

7. OPEN RESEARCH CHALLENGES

As NDN is an emerging area of research, it offers plenty of open research challenges for current as well as future researchers. We have pointed out some major research challenges in this Section categorized based on the taxonomy tree, we introduced in Fig. 3.

7.1 Naming

NDN architecture should support globally unique, human-readable, secure and location-independent names [10]. Therefore, the major issue is to develop a naming mechanism that can satisfy all these requirements. Currently existing naming approaches, like flat, hierarchical, and attribute-value, supports some of the requirements. Flat names provide uniqueness, and induce no overhead for finding the longest prefix matching. Flat names can be self-certifying and can be easily handled with highly scalable structure, like DHTs. However, flat names do not support name aggregation. Due to this, the use of flat name increases the routing table size and reduces network scalability. But still, there is no specific research whether flat names can provide required performance or not. Hierarchical names are human-friendly and supports name aggregation. Thus, it minimizes the routing table size and update time, and makes network scalable. However, because of name aggregation, hierarchical names do not fully support persistence. As in NDN, *CN*s shows content properties explicitly. Any change in the content hierarchy, either through owner change or any modification through a content provider, changes the *CN* [158]. In hierarchical naming mechanism, each name prefix is unique and supports forwarding of the packets as in the Internet. A major problem associated with hierarchical naming is that hierarchical names show content properties explicitly. Bari, et al. [13] pointed out that it is better to use self-certifying flat names because they support authenticity, uniqueness and

persistence. However, we still lack general agreement regarding which naming mechanism should be used and it remains an open research issue. Users, on the other hand, are interested in retrieving the content by using human-friendly names that do not satisfy network requirements. Therefore, naming mechanisms for NDNs are still an important challenge to be investigated.

NDN forwards the packets using *CN*, where the name has variable and unbounded length, and consumes high memory. Therefore, the high FIB update rate induces a scalability challenge for the name resolution. Another solution is to apply hashing to map the *CN*s into a fixed integer like, IP address and, then, use IP based routing algorithms for forwarding. Although many such schemes are proposed, there are still needs to develop one or more flexible and practical scheme(s) to support low processing time, high throughput and reliability at the router level.

7.2 Routing

In NDN, developing inter-domain routing protocols is still an open research area. Moreover, the main challenge is to map proposed routing solutions at the Internet level. The protocol used in Internet for inter-domain routing is BGP which is based on policies rather than shortest paths. The development of inter-domain routing is a challenging effort because policy agreement frequently changes in NDN.

In NDN, forwarding plane is responsible for fault detection, recovery and error control. On the other hand, in NDN, forwarding plane has reduced the role of routing only up to bootstrapping forwarding, and long-term topology dissemination. Therefore, there is a need to more study the forwarding and routing areas which are not present in the current Internet. Moreover, for solving the routing scalability problem, conventional routing can be replaced with geometric routing, i.e., hyperbolic routing [7][8] for NDN. This method requires a service for mapping names to hyperbolic coordinates instead of a global routing table. But, so far, very little attention has been paid towards hyperbolic routing performance and how hyperbolic routing can handle complex routing policies between ISPs.

In highly dynamic networks, nodes may move frequently. In NDN, *D_pkt*s trace the path followed by its *I_pkt* in reverse. If an intermediate node changes its location after the *I_pkt* transfer, reverse path will also change. Therefore, handling NDN mobility in highly dynamic network is an open research issue.

7.3 Caching

When several types of traffic compete for the limited cache space, cache-space management becomes a major concern for the NDN. Though some selective caching schemes have been proposed for NDN architecture, the area is still wide open for researchers. In selective caching, whole network-based traffic is classified into popular and unpopular contents which are resource consuming to implement. Other requirements for selective caching are not to cache privacy-sensitive content, cache only highly popular contents (in the network), and replacement of less popular contents. Therefore, the main research challenge is to find out content popularity correctly while incurring least possible resources. At the same time, another problem associated with selective caching is to deal with the fake popularity of contents. Moreover, the deployment of caching and replication mechanisms inside the network opens up the possibility of jointly optimizing routing, forwarding and in-network cache management. For instance, routing decisions could be affected by cache locations, the cache-ability of information and/or indications of cache contention.

The routers maintain limited space as a cache because of cost and performance. Content replacement policies are used to make room for new most popular content and remove less popular content from the cache. There is a trade-off between the processing capability of the router and cache replacement policies complexity. These policies must be less complex due to processing constraints at the router. Many complex content replacement policies exist, but still, there is need to develop the content replacement policies which can be scaled at the Internet level.

The management of these cached replicas increases the cost in mobile environments like, MANET and DTN whereas, some recent mobile routing algorithms support opportunistic on-path caching. Therefore, designing routing algorithms that can mitigate the overhead of keeping track of routing information to access cached contents is a research challenge. Another open issue is to deal with the unpopular contents as they do not get any benefits from caching. Varvello, et al. [107] have presented that the presence of unpopular contents reduces the overall performance benefits of the structured routing protocols.

7.4 Forwarding

Many data structures [55][60][62][66] exist for implementing the PIT and FIB at the NDN router. Still, this research area is open for developing efficient data structures supporting less memory consumption, and high-speed lookups.

As already discussed, forwarding strategies explores the multi-path capability of NDN and choose best outgoing interface(s) to forward the *I_pkt*. It supports load balancing across paths, congestion control, link failure, and detect several attacks, such as prefix hijack and DDoS [29][65]. There are still open challenges to design effective and efficient forwarding strategies for different contexts and networks.

NDN does not have any transport layer, the main responsibility of IP's transport layer has been shifted into the NDN forwarding plane. In NDN, in-network caching, replication, multi-path routing, new delivery modes like, multicast, unicast, and anycast, make data transport more complicated than in the Internet. Although, many techniques have been proposed for exploring the functionalities of these new mechanisms, still there is need to explore the design of flow, congestion and error

control functions. Another open issue is to explore that how to utilize the multi-path capability in a better way.

7.5 Mobility

Current Internet architecture still facing problems for supporting the mobility while, mobility in NDN, has not been explored too much. Consumer mobility is in-built in the NDN while provider's mobility induces overhead and scalability issues. Whenever a provider moves, there is a need to update locator information. The impact of provider's mobility can be minimized through caching and replication of popular contents but still unpopular contents suffers. Therefore, better solutions for the provider's mobility are a major research challenge.

NDN mobility minimizes the hand-off delay because of the availability of cached and replicated copies of the content hosted by some intermediate node between the consumer and the actual provider. But, the main problem is that many real-time communications has less use of caching (e.g., a voice call). Therefore, real-time multimedia support for NDN is a major research challenge.

Most of the time, the vehicles use information through crowd-sourcing. Therefore, the major challenge is to authenticate data without revealing provider's identity.

7.6 Security, Privacy and Trust

As NDN deals directly with the *CN*s, content properties remain available explicitly. Therefore, privacy and security of the content are an open research challenge as discussed in Section 5.1. In NDN architecture, encryption with keys associated with *CN* is used for security. There are very few literature for key management, i.e., who will be responsible for key creation, distribution and revocation. Another largely unexplored research area is access control mechanisms, i.e., definition of access control policies, applications of access control policies for cached data and user authentication [10]. Moreover, an attacker may track user information by monitoring their content requests when it is available to all intermediate nodes between the user node and the content provider. A permanent solution is not provided yet. In conclusion, security, trust management and privacy are still an open research area in NDN.

7.7 Applications

As NDN is a new research area, a big scope exists for the development of several applications. Many applications for NDN has been developed as discussed in Section 6. Nevertheless, there is need to develop applications for the various network environments and contexts to explore the NDN benefits, such as multicast, in-network caching, intelligent and stateful forwarding, scalability, easy application development and deployment nature.

REFERENCES

[1] www.history-computer.com/Internet/Maturing/TCPIP.html

[2] YouTube. Available: https://www.youtube.com

[3] Netflix. Available: https://www.netflix.com/

[4] Amazon, Available: www.amazon.com/

[5] iTunes. Available: www.apple.com/itunes/

[6] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, K.C. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. (2010). Named Data Networking (NDN) Project. [Online]. Available: http://named-data.net/project/annual-progress-summaries/

[7] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, K.C. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. (2011). Named Data Networking (NDN) Project. [Online]. Available: http://named-data.net/project/annual-progress-summaries/

[8] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, K.C. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh. (2012). Named Data Networking (NDN) Project. [Online]. Available: http://named-data.net/project/annual-progress-summaries/

[9] Future Internet Architecture. [Online]. Available: www.nets-fia.net/

[10] V. Jacobson, D. K. Smetters, Thornton, D. James, P. F. Micheal, B. H. Nicolas, B. L. Rebeeca, "Networking named content " In Proceedings of the 5th International conference on Emerging networking experiments and technologies (CoNEXT), 2009.

[11] Ahlgren, Bengt, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," IEEE Communications Magazine, 50(7), (2012): 26-36.

[12] Z.A. Jaffri, Z. Ahmad, and M. Tahir, "Named Data Networking (NDN), New Approach to Future Internet Architecture Design: A Survey," In International Journal of Informatics and Communication Technology (IJ-ICT), 2(3), (2013): 155-165.

[13] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks," In IEEE Communications Magazine, 50(12), (2012): 44–53.

[14] H. Hu, J. Bi, T. Feng, S. Wang, P. Lin, and Y. Wang, "A Survey on New Architecture Design of Internet," In Proceedings of third IEEE International Conference on Computational and Information Sciences (ICCIS), 2011, pp. 729–732.

[15] European Community Future Internet Architecture (FIArch) Experts Group. (2011, March) Fundamental limitations of current Internet and the path to future Internet. [Online]. Available: http://www.future-internet.eu/publications/view/article/fundamental- limitations-of-current-internet.html

[16] J.S. Khoury, and C. T. Abdallah, "A Survey of Novel Internetwork (and Naming) Architectures," In *Internet Naming and Discovery*, Springer, London, 2013, pp. 13-33.

[17] J. Pan, S. Paul, and R. Jain, "A Survey of the Research on Future Internet Architectures," In IEEE Communications Magazine, 49(7), (2011): 26–36.

[18] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, "A survey of mobility in information-centric networks: challenges and research directions." In Proceedings of First ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications, 2012, pp. 1-6.

[19] NSF NeTS FIND Initiative, Available: http://www.nets-find.net.

[20] NEBULA Project, Available: http://nebula.cis.upenn.edu.

[21] eXpressive Internet Architecture Project, Available: http://www.cs.cmu.edu/~xia/.

[22] Global Environment for Network Innovations (GENI) Project, Available: http://www.geni.net/.
[23] The FP7 4WARD Project, Available: http://www.4ward-project.eu/.
[24] FIRE: Future Internet Research and Experimentation, Available: http://cordis.europa.eu/fp7/ict/fire/
[25] AKARI Architecture Design Project, Available: http://akari-pro- ject.nict.go.jp/eng/index2.htm
[26] JGN2plus- Advanced Testbed Network for R&D, Available: http://www.jgn.nict.go.jp/english/index.html.
[27] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, and B. Liu, "Fast name lookup for Named Data Networking," In Proceedings of the 22ndIEEEInternational Symposium on Quality of Service (IWQoS), 2014, pp. 198-207.
[28] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the Role of Routing in Named Data Networking," Tech. Rep. NDN-0016, 2013.
[29] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," In Proceedings of the ACM Conference on Computing Communication (SIGCOMM), 36(7), (2013): 779-791.
[30] M. Vahlenkamp, "Threats on Information-Centric Networking," Master Thesis, Department of Computer Science, Hamburg University of Applied Sciences, 2013.
[31] L. Wang and A. K. M. M. Hoque, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," Tech. Rep. NDN-0003, 2012, pp. 1–15.
[32] H. Dai, J. Lu, Y. Wang, and B. Liu, "A Two-layer Intra-domain Routing Scheme for Named Data Networking," In Proceeding of IEEE Global Communications Conference (GLOBECOM), 2012, pp. 2815–2820.
[33] A.K.M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, and L. Wang, "NLSR: Named-data Link State Routing Protocol," In Proceedings of the 3rd ACM workshop on Information-centric networking (SIGCOMM), 2013, pp. 15–20.
[34] S. Dibenedetto, C. Papadopoulos, and D. Massey, "Routing Policies in Named Data Networking," In Proceedings of the ACM workshop on Information-centric networking (SIGCOMM), 2011, pp. 38–43.
[35] NDN testbed. http://www.arl.wustl.edu/~jdd/ndnstatus/ndn_prefix/tbs_ndnx.html.
[36] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: a survey," Computer Networks, 57(16), (2013): 3128-3141.
[37] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, and X. Wang, "Popularity-driven Coordinated Caching in Named Data Networking," In Proceedings of the 8th ACM/IEEE symposium on Architectures for networking and communications systems, 2012, pp. 15–26.
[38] J. Li, H. Wu, B. Liu, and J. Lu, "Effective caching schemes for minimizing inter-ISP traffic in named data networking," In Proceedings of the 18th IEEE International Conference on Parallel and Distributed Systems (ICPADS), 2012, pp. 580 –587.
[39] H. Wu, J. Li, Y. Wang, and B. Liu, "EMC: The Effective Multi-path Caching Scheme for Named Data Networking," In Proceeding of the 22nd IEEE International Conference on Computer Communications and Networks (ICCCN), 2013, pp. 1-7.
[40] H. Wu, J. Li, T. Pan, and B. Liu, "A novel caching scheme for the backbone of named data networking," In the Proceedings of the IEEE International Conference on Communications (ICC), 2013, pp. 3634-3638.
[41] E. Yeh, M. Burd, and D. Leong, "VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networks," In Proceedings of the 1st ACM International Conference on Information-centric networking, 2014, pp. 117-126.
[42] M. Rezazad, and Y. C. Tay, "A cache miss equation for partitioning an NDN content store," In Proceedings of the 9th ACM Asian Internet Engineering Conference, 2013, pp. 1–8.
[43] X. Hu, C. Papadopoulos, J. Gong, and D. Massey, "Not So Cooperative Caching in Named Data Networking," In Proceedings of IEEE Global Communications Conference (GLOBECOM), 2013, pp. 2263–2268.
[44] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Dis- tributed selfish replication," In IEEE Transactions on Parallel Distributed Systems, 17(12), pp. (2006), 1401–1413.
[45] Y. Zeng, and X. Hong, "A Caching Strategy in Mobile Ad Hoc Named Data Network," In Proceeding of the 6th International Conference on Communications and Networking (CHINACOM), 2011, pp. 805–809.
[46] H. Choi, J. Yoo, T. Chung, N. Choi, T. Kwon, and Y. Choi, "CoRC: coordinated routing and caching for named data networking." In Proceedings of the 10th ACM/IEEE symposium on Architectures for networking and communications systems, 2014, pp. 161-172.
[47] M. Dehghan, B. Jiang, A. Dabirmoghaddam, and D. Towsley, "On the Analysis of Caches with Pending Interest Tables," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 69-78.
[48] C. Barakat, A. Kalla, D. Saucez, and T. Turletti, "Minimizing bandwidth on peering links with deflection in named data networking," In Proceeding of the 3rd International Conference on Communications and Information Technology (ICCIT), 2013, pp. 88–92.
[49] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," In Proceedings of the Second ACM International Conference onInternet Measurement Workshop (IMW), 2002.
[50] J. Ran, N. Lv, D. Zhang, and Z. Xie, "On Performance of Cache Policies in Named Data," In Proceedings of International Conference on Advanced Computer Science and Electronics Information, 2013, pp. 668–671.
[51] H. Dai, Y. Wang, H. Wu, J. Lu, and B. Liu, "Towards Line-Speed and Accurate On-line Popularity Monitoring on NDN Routers," In Proceedings of 22nd IEEE International Symhosium on Quality of Service (IWQoS), 2014.
[52] W. Dron, A. Leung, M. Uddin, S. Wang, T. Abdelzaher, R. Govindan, and J. Hancock, "Information-maximizing Caching in Ad Hoc Networks with Named Data Networking," In Proceeding of 2nd IEEE Network Science Workshop (NSW), 2013, pp. 90–93.
[53] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa. Supporting cooperative caching in disruption tolerant networks. In Proc. of ICDCS, 2011.
[54] M. Varvello, D. Perino, and L. Linguaglossa, "On the design and implementation of a wire-speed pending interest table," In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2013, pp. 369-374.
[55] H. Dai, B. Liu, Y. Chen, and Y. Wang, "On pending interest table in named data networking," In Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems, 2012, pp. 211-222.
[56] H. Yuan, P. Crowley, and B. Case, "Scalable Pending Interest Table Design: From Principles to Practice Interfaces," In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2014, pp. 2049–2057.
[57] Z. Li, et al., "MaPIT: An Enhanced Pending Interest Table for NDN with Mapping Bloom Filter," IEEE Communications Letters, 18(11), (2014): 1915-1918.
[58] A. Kirsch, M. Mitzenmacher, and G. Varghese, "Hash-based techniques for high-speed packet processing" In *Algorithms for Next Generation Networks*, pages 181–218. Springer, 2010.
[59] W. You, B. Mathieu, P. Truong, J.-F. Peltier, and G. Simon, "DiPIT: a distributed bloom-filter based PIT table for CCN nodes," In Proceedings of 21st International Conference on Computer Communications and Networks (ICCCN), 2012, pp. 1-7.
[60] Z. Li, J. Bi, S. Wang, and X. Jiang, "Compression of pending interest table with bloom filter in content centric network," In Proceedings of the ACM 7th International Conference on Future Internet Technologies, (CFI), 2012.
[61] D. Perino and M. Varvello, "A reality check for content centric networking," In Proceedings of the ACM workshop on Information-centric networking (SIGCOMM), 2011, pp. 44-49.
[62] H. Yuan, T. Song, and P. Crowley, "Scalable NDN forwarding: Concepts, issues and principles," In Proceedings of 21st International Conference on Computer Communications and Networks (ICCCN), 2012, pp. 1-9.
[63] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practiceof bloom filters for distributed systems," IEEE Commun. Surveys Tuts., 14(1), 2012, pp. 131–155.
[64] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Pending Interest Table Sizing in Named Data Networking," In Proceedings of the 2nd

International Conference on Information-Centric Networking, 2015, pp. 49-58.
[65] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," In Proceedings of the ACM Conference on Computing Communication(SIGCOMM), 42(3), 2012, pp. 62-67.
[66] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable Name Lookup in NDN Using Effective Name Component Encoding," In Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems (ICDCS), 2012, pp. 688-697.
[67] Y. Wang, T. Pan, Z. Mi, H. Dai, X. Guo, T. Zhang, and B. Liu, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage Bloom filters," International Conference on Computer Communications (INFOCOM), 2013, pp. 95–99.
[68] W. So, A. Narayanan, D. Oran, and Y. Wang, "Toward fast NDN software forwarding lookup engine based on hash tables," In Proceedings of the 8th ACM/IEEE symposium on Architectures for networking and communications systems, 2012, pp. 85-86.
[69] W. Quan, C. Xu, J. Guan, H. Zhang, and L. A. Grieco, "Scalable Name Lookup with Adaptive Prefix Bloom Filter for Named Data Networking," IEEE communication, 18(1), (2014): 102–105.
[70] Y. Wang, H. Dai, J. Jiang, K. He, W. Meng, and B. Liu, "Parallel Name Lookup for Named Data Networking," In Proceedings of the IEEE Global Telecommunication Conference (GLOBECOM), 2011, pp. 1–5.
[71] W. So, A. Narayanan, D. Oran, and M. Stapp, "Named Data Networking on a Router: Forwarding at 20Gbps and Beyond Categories and Subject Descriptors," In ACM SIGCOMM Computer Communication Review, 43(4), 2013, pp. 495-496.
[72] W. So, A. Narayanan, and D. Oran, "Named data networking on a router: Fast and DoS-resistant forwarding with hash tables," In Proceedings of the ninth ACM/IEEE symposium on Architectures for networking and communications systems, pp. 215-226, 2013.
[73] E. Fredkin, "Trie memory," ACM Communication, 3(9), (1960):pp. 490–499.
[74] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest Prefix Matching using Bloom Filters," In Proceedings of the International Conference on Applications, technologies, architectures, and protocols for computer communications, 2003, pp. 201-212.
[75] Y. Wang, D. Tai, T. Zhang, J. Lu, B. Xu, H. Dai, and B. Liu, "Greedy name lookup for named data networking," In Proceedings of the ACM International Conference on Measurement and modeling of computer systems, 2013, pp. 359-360.
[76] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, and B. Liu, "Fast name lookup for Named Data Networking," In Proceedings of the IEEE 22nd International Symposium of Quality of Service (IWQoS*), 2014*, pp. 198-207.
[77] H. Yuan, and P. Crowley, "Reliably Scalable Name Prefix Lookup." In Proceedings of the 11th ACM/IEEE Symposium on Architectures for networking and communications systems, 2015, pp. 111-121.
[78] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable Name-Based Packet Forwarding: From Millions to Billions," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 19-28.
[79] Y. Wang, H. Dai, T. Zhang, W. Meng, J. Fan, and B. Liu, "GPU-accelerated name lookup with component encoding," Computer Networks, 57(16), (2013): 3165–3177.
[80] Y. Wang, Y. Zu, T. Zhang, K. Peng, Q. Dong, B. Liu, W. Meng, et al., "Wire Speed Name Lookup: A GPU-based Approach," In 10thUSENIX Symposium on Networked Systems Design & Implementation, 2013, pp. 199-212.
[81] D. Perino, M. Varvello, L. Linguaglossa, R. Laufer, and R. Boislaigue, "Caesar: a content router for high-speed forwarding on content names," In Proceedings of the 10th ACM/IEEE symposium on Architectures for networking and communications systems, 2014, pp. 137-148.
[82] Y. Li, D. Zhang, X. Yu, W. Liang, J. Long, and H. Qiao, "Accelerate NDN name lookup using FPGA: Challenges and a scalable approach," In Proceedings of the 24th IEEE International Conference on Field Programmable Logic and Applications (FPL), 2014, pp. 1-4.
[83] T. Zhang, Y. Wang, T. Yang, J. Lu, and B. Liu, "NDNBench: A Benchmark for Named Data Networking Lookup," In Proceedings of IEEE Global Telecommunication Conference (GLOBECOM), 2013, pp. 2174–2179.
[84] M. Rezazad and Y. C. Tay, "ndn||mem: an Architecture to Alleviate the Memory Bottleneck for Named Data Networking," In Proceedings of the CoNEXT Student Workshop, 2013, pp. 1–4.
[85] J. Torres, L. Ferraz, and O. Duarte, "Controller-based Routing Scheme for Named Data Network," Tech. Rep. Electrical Engineering Program, COPPE/UFRJ, 2012.
[86] J.M. Hsu, J.Y. Chang, "A CRC-32 Name Prefix Encoding in Named Data Networking," International Journal of Advanced Information Technologies (IJAIT), 8(2), (2014): 125-130.
[87] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure Namespace Mapping to Scale NDN Forwarding," In Proceedings of the 18thIEEE Global Internet Symposium, 2015.
[88] W. So, T. Chung, H. Yuan, D. Oran, and M. Stapp, "Toward terabyte-scale caching with SSD in a named data networking router," In Proceedings of the 10th ACM/IEEE symposium on Architectures for networking and communications systems, 2014, pp. 241-242.
[89] A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "To NACK or not to NACK? Negative Acknowledgments in Information-Centric Networking,"2015.
[90] M. Amadeo, C. Campolo, and A. Molinaro, "Forwarding Strategies in Named Data Wireless Ad hoc Networks: Design and Evaluation," In Journal of Network and Computer Applications, (2014): 1–12.
[91] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, and L. Zhang, "Rapid Traffic Information Dissemination Using Named Data," In Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications, 2012, pp. 7–12.
[92] SY Oh, D. Lau, M. Gerla, "Content centric networking in tactical and emergency MANETs," In Wireless days, IFIP; 2010.
[93] M. Amadeo, A. Molinaro, G. Ruggeri, "E-CHANET: routing forwarding and transport in information-centric multihop wireless networks," Computer Communication, 36(7), (2013): 792-803.
[94] M. Meisel, V. Pappas, L. Zhang, "Listen first broadcast later: topology-agnostic forwarding under high dynamics," In Annual Conference of International Technology Alliance in Network and Information Science; 2010, pp. 8.
[95] F. Angius, G. Pau, M. Gerla, "BLOOGO: BLOOm filter based GOssip algorithm for wireless NDN," In Proceedings of the1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications, 2012, pp. 25–30.
[96] Y. Lu, B. Zhou, L-C. Tung, M. Gerla, A. Ramesh, L. Nagaraja, "Energy-efficient content retrieval in mobile cloud," In Proceeding of the ACM SIGCOMM workshop on mobile cloud computing ser. (MCC), 2013, pp. 21-26.
[97] C. Li, L. Wenjing and O. Koji, "Greedy Ant Colony Forwarding Algorithm for Named Data Networking," In Proceedings of the Asia-Pacific Advanced Network, 34, 17–26, 2012.
[98] H. Qian, R. Ravindran, G.Q. Wang, and D. Medhi, "Probability-Based Adaptive Forwarding Strategy in Named Data Networking," In ACM SIGCOMM computer communication review, 42(3), (2012): 62-67.
[99] X. Zeng and Z. H. Gao, "Rank-Based Routing Strategy for Named Data Network," In Applied Mechanics and Materials, 543, (2014): 3320–3323.
[100] M. Tortelli, L. A. Grieco, and G. Boggia, "Performance Assessment of Routing Strategies in Named Data Networking," GTTI Session on Telecommunication Networks, 2013.
[101] K.M. Schneider, and U.R. Krieger, "Beyond Network Selection: Exploiting Access Network Heterogeneity with Named Data Networking," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 137-146.
[102] Y. Wang, N. Rozhnova, U. Pierre, and D. Oran, "An Improved Hop-by-hop Interest Shaper for Congestion Control in Named Data Networking," In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013, pp. 55–60.
[103] X. Jiang, and J. Bi, "Interest set mechanism to improve the transport of named data networking," In Proceedings of the ACM SIGCOMM, 2013, pp.

515-516.
[104] X. Jiang, “Interest Set Mechanism to Support Flow Transmission in NDN,” 2013.
[105] M. Amadeo, A. Molinaro, C. Campolo, M. Sifalakis, and C. Tschudin, “Transport Layer Design for Named Data Wireless Networking,” In Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM), 2014, pp. 464–469.
[106] D. Ó Coileáin, and D. O' Mahony, "SAVANT: Aggregated feedback and accountability framework for named data networking," In Proceedings of the 1st ACM International Conference on Information-Centric Networking, 2014, pp. 187-188.
[107] M. Varvello, I. Rimac, U. Lee, L. Greenwald, V. Hilt, “On the design of content-centric MANETs,” In International Conference on wireless on-demand network systems and services (WONS), 2011.
[108] M. Amadeo, C. Campolo, A. Molinaro, “Enhancing content-centric networking for vehicular environments,” Computer Networks, 57(16), (2013):3222–34
[109] Y. Yu, R. B. Dilmaghani, S. Calo, M. Y. Sanadidi, and M. Gerla, “Interest Propagation In Named Data MANETs,” In Proceedings of IEEE International conference on Computing, Networking and Communications (ICNC), 2013, pp. 1118–1122.
[110] L. Zhang, et al., "Named data networking, "In ACM Computer Communication Review (SIGCOMM), 44(3), (2014): 66-73.
[111] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21:120–126, 1978.
[112] NIST. FIPS 186-3, June 2009.
[113] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The elliptic curve digital signature algorithm (ecdsa). Int. J. Inf. Sec., 1(1):36–63, 2001.
[114] W. Shang, Q. Ding, A. Marianantoni, J. Burke, and L. Zhang, “Securing Building Management Systems Using Named Data Networking,” IEEE, 28(3), (2014): 50–56.
[115] B. Hamdane, M. Msahli, A. Serhrouchni, and S. Guemara El Fatmi, “Data-based access control in Named Data Networking,” In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Work sharing (Collaboratecom), 2013, pp. 531-536.
[116] J. Park, R. Sandhu, “The UCONABC usage control model,” In ACM Transactions on Information and System Security (TISSEC), 7(1), (2004): 128-174.
[117] Massawe, E. Alphonce, S. Du, and H. Zhu, "A Scalable and Privacy-Preserving Named Data Networking Architecture Based on Bloom Filters," In Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW), 2013, pp. 22-26.
[118] Y. Yu and A. Afanasyev, “An Endorsement-based Key Management System for Decentralized NDN Chat Application,” University of California, Los Angeles, Tech. Rep. NDN-0023, 2014.
[119] ChronoChat, Available: https://github.com/named-data/ChronoChat.
[120] V. Pournaghshband, and K. Natarajan, “A Robust Trust Model for Named-Data Networks,” University of California, Los Angeles, 2011.
[121] C. Bian, Z. Zhu, E. Uzun and L. Zhang, “Deploying Key Management on NDN Testbed,” UCLA, Peking University and PARC, Tech. Rep. (2013).
[122] Z. Zhu, J. Burke, L. Zhang, P. Gasti, Y. Lu, and V. Jacobson, “A new approach to securing audio conference tools,” In Proceedings of the 7th Asian Internet Engineering Conference (AINTEC), 2011, pp. 120–123.
[123] J. Burke, P. Gasti, and N. Nathan, “Securing Instrumented Environments over Content-Centric Networking: the Case of Lighting Control and NDN,” arXiv preprint: 1208.1336, 2012.
[124] J. Burke, A. Horn, and A. Marianantoni, “Authenticated Lighting Control Using Named Data Networking,” University of California, Los Angeles, Tech. Rep. NDN-0011, October, 2012.
[125] V. Perez, M. T. Garip, S. Lam, and L. Zhang, “Security Evaluation of a Control System Using Named Data Networking,” pp. 1–6, 2013.
[126] W. Granzer, W. Kastner, G. Neugschwandtner and F. Praus, “Security in Networked Building Automation Systems,” 2006.
[127] Y. Yu, A. Afanasyev, D. Clark, V. Jacobson, and L. Zhang, "Schematizing Trust in Named Data Networking," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 177-186.
[128] H. Ballani, P. Francis, and X. Zhang, “A Study of Prefix Hi- jacking and Interception in the Internet,” ACM SIGCOMM Comput. Commun. Rev., 37(4), (2007): 265–276.
[129] M. Paknezhad and M. Keshtgary, “Security and Privacy Issues of Implementing Cloud Computing on NDN,” Journal of Basic and Applied Scientific Research, 4(3), (2014): 270–279.
[130] P. Gasti,G. Tsudik, E. Uzun, and L. Zhang, “DoS and DDoS in named data networking,” In Proceeding of the 22nd IEEE International Conference on Computer Communications and Networks (ICCCN), 2013, pp. 1-7.
[131] NSF FIA PI Meeting, 2011. Named Data Networking. May 22, 2013, Available: www.named-data.net.
[132] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, and E. Biersack, “Privacy Implications of Ubiquitous Caching in Named Data Networking Architectures,” Tech. Rep. TR-iSecLab-0812-001.
[133] S. DiBenedetto, P. Gasti, G. Tsudik, andE. Uzun, "ANDaNA: Anonymous named data networking application." arXiv preprint: 1112.2205(2011).
[134] N. Ntuli and S. Han, “Detecting Router Cache Snooping in Named Data Networking,” In Proceedings of the IEEE International Conference on ICT Convergence (ICTC), 2012, pp. 714–718.
[135] C.B. Lee, C. Roedel, and E. Silenok, “Detection and Characterization of Port Scan Attacks,” Univeristy of California, Department of Computer Science and Engineering, 2003.
[136] S. Al-Sheikh, M. Wählisch, and T.C. Schmidt, "Revisiting Countermeasures Against NDN Interest Flooding," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 195-196.
[137] C. Ghali and G. Tsudik and E. Uzun, “Elements of Trust in Named-Data Networking,” In arXiv preprint (2014): 1402.3332.
[138] C. Ghali and G. Tsudik, “Needle in a Haystack: Mitigating Content Poisoning in Named-Data Networking,” In Proceedings of the NDSS Workshop on Security of Emerging Networking Technologies (SENT), 2014.
[139] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient Content Verification in Named Data Networking." In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 109-116.
[140] C. Ghali, G. Tsudik, and E. Uzun, "Network-layer trust in named-data networking," *ACM SIGCOMM Computer Communication Review*, 44(5), (2014): 12-19.
[141] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, “Privacy risks in named data networking,” In ACM SIGCOMM Computer Communication, 42(5), (2012): 54-57.
[142] M. Conti, P. Gasti, and M. Teoli, “A lightweight mechanism for detection of cache pollution attacks in Named Data Networking,” Computer Networks, 57(16), (2013): 3178–3191.
[143] H. Park, I. Widjaja, H. Lee, “Detection of cache pollution attacks using randomness checks,” In Proceedings of the IEEE International Conference on Communications(ICC), 2012, pp. 1096–1100.
[144] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik, “Cache Privacy in Named-Data Networking,” In Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems (ICDCS), 2013, pp. 41-51.
[145] A. Karami, and M. Guerrero-Zapata, "An ANFIS-based cache replacement method for mitigating cache pollution attacks in Named Data Networking," Elsevier Computer Networks, 80, (2015): 51-65.
[146] A. Karami, and M. Guerrero-Zapata, "A hybrid multiobjective rbf-pso method for mitigating dos attacks in named data networking," Neurocomputing, 151, (2015): 1262-1282.
[147] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, “Interest Flooding Attack and Counter measures in Named Data Networking,” In

Proceedings of the Federation for Information Processing Networking (IFIP), 2013, pp. 1-9.
[148] H. Dai, Y. Wang, J. Fan, and B. Liu, “Mitigate DDoS Attacks in NDN by Interest Traceback,” In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2013, pp. 381-386.
[149] A. Compagno, M. Conti, P. Gasti, L.V. Mancini, and G. Tsudik, "Violating Consumer Anonymity: Geo-locating Nodes in Named Data Networking," 2015.
[150] J. Tang, Z. Zhang, Y. Liu, and H. Zhang, “Identifying Interest Flooding in Named Data Networking,” In Proceedings of the IEEE International Conference on Green Computing and Communications (GreenCom), 2013, pp. 306–310.
[151] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, “Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking,” In Proceedings of the 38th IEEE Conference on Local Computer Networks (LCN), 2013, pp. 630-638.
[152] Z. Li, and J. Bi, “Interest Cash: An Application-based Countermeasure against Interest Flooding for Dynamic Content in Named Data Networking,” In Proceedings of Ninth ACM International Conference on Future Internet Technologies, 2014, pp. 1.
[153] B. Hamdane, A. Serhrouchni, A. Fadlallah, S. Guemara, and E. Fatmi, “Named-Data Security Scheme for Named Data Networking,” In Proceedings of the Third IEEE International Conference on Network of the Future (NOF), 2010, pp. 1-6.
[154] E. Min, Z. Chen, R. Chen, and D. Wang, “Reformat Named Data Networking with Cryptographic Message Syntax,” In Proceedings of the Third International Conference on Networking and Distributed Computing (ICNDC), 2012, pp. 21–25.
[155] K. Kusunoki, Y. Kawahara, and T. Asami, “Supervisor application for content management in named data networking,” In Proceedings of the 11th IEEE International Symposium on Autonomous Decentralized Systems (ISADS), 2013, pp. 1–6.
[156] C. E. Perkins, "Mobile networking through mobile IP." Internet Computing, IEEE2.1 (1998): 58-69.
[157] Host Identification Protocol (HIP), Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-1/121_host.html.
[158] B. De, M. Gabriel, B. Pedro Velloso, and M. Igor Moraes, “Information Centric Networks: A New Paradigm for the Internet,” John Wiley & Sons, May 2013.
[159] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe, “A Survey of Mobility in Information-Centric Networks,” Communications of the ACM, 56(12), (2013): 90-98.
[160] M. Meisel, V. Pappas, and L. Zhang, “Ad hoc networking via named data,” In Proceedings of the 5th ACM International workshop on Mobility in the evolving Internet architecture, 2010, pp. 3-8.
[161] Z. Zhu, A. Afanasyev, and L. Zhang, “A New Perspective on Mobility Support,” Named-Data Networking Project, Tech. Rep., 2013.
[162] K. Shilton, J. Burke, C. Duan, and L. Zhang, “A World on NDN: Affordances & Implications of the Named Data Networking Future Internet Architecture,” Tech. Rep. NDN - 0018, 2014, pp. 1–19.
[163] J.B. Kenney, “Dedicated Short-Range Communications (DSRC) Stan- dards in the United States,” In Proceedings of the IEEE, 99(7), 2011, pp. 1162–1182.
[164] A. Azgin, R. Ravindran, and G. Wang, "A Scalable Mobility-Centric Architecture for Named Data Networking,” arXiv preprint arXiv: 1406.7049, (2014).
[165] A. Azgin, R. Ravindran, and G. Wang, "Mobility study for named data networking in wireless access networks," In Proceedings of IEEE International Conference on Communications (ICC), 2014, pp. 3252-3257.
[166] X. Jiang, J. Bi, and Y. Wang, “What Benefits Does NDN Have in Supporting Mobility,” In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), 2014, pp. 1-6.
[167] X. Jiang, J. Bi, Y. Wang, P. Lin, and Z. Li, “A Content Provider Mobility Solution of Named Data Networking,” In Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP), 2012, pp. 1-2.
[168] Y. Rao, D. Gao, and H. Luo, “NLBA: A Novel Provider Mobility Support Approach in mobile NDN environment,” In Proceedings of the 11th IEEE Consumer Communications and Networking Conference (CCNC), 2014, pp. 188-193.
[169] R.A.O. Ying, L. U. O. Hongbin, G. A. O. Deyun, Z. Huachun, and Z. Hongke, “LBMA: A novel Locator Based Mobility support Approach in Named Data Networking," Communications, China, 11(4), (2014): 111-120.
[170] Y. Zhang, H. Zhang, and L. Zhang, “Kite: A Mobility Support Scheme for NDN,” In Proceedings of the 1st ACM International Conference on Information-centric networking, 2014, pp. 179-180.
[171] R. Ravindran, S. Lo, X. Zhang, and G. Wang, “Supporting Seamless Mobility in Named Data Networking,” In Proceedings of the IEEE International Conference on Communications (ICC), 2012, pp. 5854-5869.
[172] Y. Li, Z. Zhao, T. Lin, H. Tang, and S. Ci, “A SIP-based Real-time Traffic Mobility Support Scheme in Named Data Networking,” Journal of Networks, 7(6), (2012): 918–925.
[173] Y. Rao, H. Zhou, D. Gao, H. Luo, and Y. Liu, “Proactive Caching for Enhancing User-Side Mobility Support in Named Data Networking,” In Proceeding of 7th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013, pp. 37-42.
[174] 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 ACM workshop on Re-architecting the Internet, 2009, pp. 1-6.
[175] R. Koodli, Ed, “Fast Handovers for Mobile IPv6,” RFC 4068, July 2005.
[176] B. Etefia and L. Zhang, “Named Data Networking for military communication systems,” In Proceedings of the IEEE International Aerospace Conference, 2012, pp. 1–7.
[177] M. Ambrosin, M. Conti, P. Gasti, and G. Tsudik, “Covert Ephemeral Communication in Named Data Networking,” In Proceedings of the ninth ACM Symposium on Information, Computer and Communications Security (ASIACCS), 2014, pp. 15-26.
[178] M. Amadeo, C. Campolo, A. Molinaro, and N. Mitton, “Named Data Networking: a Natural Design for Data Collection in Wireless Sensor Networks,” In Proceedings of the IFIP Wireless Days (WD), 2013, pp. 1-6.
[179] J. Burke, A. Horn, and A. Marianantoni, “Named Data for Control Systems : NDN Lighting,” 2011.
[180] E. Baccelli, et al. "Information centric networking in the IoT: experiments with NDN in the wild." arXiv preprint: 1406.6608 (2014).
[181] M. Chen, “NDNC-BAN: Supporting rich media healthcare services via named data networking in cloud-assisted wireless body area networks,” In Information Sciences, 284, (2014): 142-156.
[182] D. Saxena, V. Raychoudhury, and S. Nalluri, "SmartHealth-NDNoT: Named Data Network of Things for Healthcare Services, " In Proceedings of the ACM MobiHoc Workshop on Pervasive Wireless Healthcare, 2015, pp. 45-50.
[183] O. Briante, M. Amadeo, C. Campolo, A. Molinaro, S. Y. Paratore, andG. Ruggeri, “eDomus: User-home interactions through Facebook and Named Data Networking,” In Proceedings of the Eleventh IEEE International Conference on Sensing, Communication, and Networking (SECON), 2014, pp. 155-157.
[184] M. Amadeo, C. Campolo, and A. Molinaro, "Multi-source data retrieval in IoT via named data networking," In Proceedings of the 1stACMInternational Conference on Information-centric networking, 2014, pp. 67-76,.
[185] D. Massey, C. Papadopoulos, L. Wang, B. Zhang, and L. Zhang, “Teaching Network Architecture through Case Studies,” ACM SIGCOMM Education workshop, 2011.
[186] S. Wang, J. Wu, and J. Bi, “Application Design over Named Data Networking with its Features in Mind,” In Proceedings of eleventh International Conference on Networks (ICN), 2012, pp. 121–124.
[187] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, “ACT: Audio Conference Tool Over Named Data Networking,” In Proceedings of the ACM SIGCOMM workshop on Information-centric Networking (ICN), 2011, pp. 68-73.

[188] Z. Zhu, C. Bian, and L. Zhang, “XMPP over Named Data Networking: Design,” 2012, pp. 1–8.
[189] J. Wei, D. Nguyen,J.J. Garcia-Luna-Aceves, and K. Nichols, “Experience with Collaborative Conferencing Applications in Named-Data Networks,” In Proceedings of the tenth IEEE Consumer Communications and Networking Conference (CCNC), 2013, pp. 241-246.
[190] K. Shilton, H. Bldg, C. Park, and J. A. Koepfler, "Making space for values: communication & values levers in a virtual team," In Proceedings of the 6th ACM International Conference on Communities and Technologies, 2013, pp. 110-119.
[191] K. Shilton, “Social Values in a Future Internet: Analyzing the Named Data Networking Protocols,” 2013.
[192] P. Gusev, and J. Burke, "NDN-RTC: Real-Time Videoconferencing over Named Data Networking," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 117-126.
[193] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang, “Chronos: Serverless Multi-User Chat Over NDN,” University of California, Los Angeles, Tech. Rep. NDN - 0008, 2012, pp. 1–12.
[194] D. Kulinski and J. Burke, “NDNVideo : Random-access Live and Pre-recorded Streaming using NDN,” University of California, Los Angeles, Tech. Rep. NDN – 0007, 2012, pp. 1–17.
[195] K. Lei, Y. Longyu, and W. Jun, "Scalable control panel for media streaming in NDN," In Proceedings of the 1st ACM International Conference on Information Centric Networking, 2014, pp. 207-208.
[196] G. Piro, V. Ciancaglini, R. Loti, L. A. Grieco, and L. Liquori, "Providing crowd-sourced and real-time media services through a NDN-based platform," In proceedings of modelling and processing for next generation big data technologies and applications, 2013, pp. 1–37.
[197] C. Yao, L. Fan, Z. Yan, and Y. Xiang, “Long-Term Interest for Real time Applications in the Named Data Network,” In Proceedings of ACM AsiaFI, 2012, pp. 0–7.
[198] D. Posch, C. Kreuzberger, B. Rainer, and H. Hellwagner, "Client starvation: a shortcoming of client-driven adaptive streaming in named data networking." In Proceedings of the 1st ACM International Conference on Information-centric networking, 2014, pp. 183-184.
[199] J. Lindblom, M. Huang, J. Burke, and L. Zhang, “FileSync / NDN: Peer-to-Peer File Sync over Named Data Networking,” University of California, Los Angeles, In Tech. Rep. NDN - 0012, 2013, pp. 1–6.
[200] Z. Zhu, and A. Afanasyev, “Let’s ChronoSync: Decentralized dataset state synchronization in Named Data Networking,” In Proceedings of 21st IEEE International Conference on network protocols, 2013, pp. 1–10.
[201] W. Fu, H.B. Abraham, and P. Crowley, "Synchronizing Namespaces with Invertible Bloom Filters." In Proceedings of the 11thACM/IEEE Symposium on Architectures for networking and communications systems,2015, pp. 123-134.
[202] X. Qiao, G. Nan, Y. Peng, L. Guo, J. Chen, Y. Sun, and J. Chen, “NDNBrowser: An extended web browser for named data networking,” In Journal Network and Computer Applications, (2014): 1–14.
[203] W. Shang, J. Thompson, M. Cherkaoui, J. Burke, and L. Zhang, “NDN.JS: A JavaScript Client Library for Named Data Networking,” In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2013, pp. 399-404.
[204] W. Shang, J. Thompson, J. Burke, and L. Zhang, “Development and Experimentation with NDN-JS, a JavaScript Library for Named Data Networking,”University of California, Los Angeles, Tech. Rep. NDN-0014, Revision 1, 2013.
[205] M.G. Rubinstein, F.B. Abdesslem, S.R. Cavalcanti, et al., “Measuring the capacity of in-car to in-car vehicular networks,” In IEEE Communications Magazine, 47(11),(2009): 128–136.
[206] M.J. Khabbaz, C.M. Assi, W.F. Fawaz, “Disruption tolerant networking: a comprehensive survey on recent developments and persisting challenges,” IEEE Communications Surveys & Tutorials, 14(2), (2012): 607–640.
[207] K. Fall, S. Farrell, “DTN: an architectural retrospective,” IEEE Journal on Selected Areas of Communications (JSAC), 26(5), (2008): 828–836.
[208] M. Amadeo, C. Campolo, A. Molinaro, “Content centric networking: is that a solution for upcoming vehicular networks?,” In Proceedings of the ninth ACM International workshop on Vehicular inter-networking, systems, and applications, 2012, pp. 99-102.
[209] G. Arnould, D. Khadraoui, Z. Habbas, “A self-organizing content centric network model for hybrid vehicular ad-hoc networks,” In Proceedings of the First ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet), 2011, pp. 15-22.
[210] P. Talebifard, V.C. Leung, “A content centric approach to dissemination of information in vehicular networks,” In Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet), 2012, pp. 17-24.
[211] Amadeo M., Campolo C., Molinaro A., “CRoWN: content-centric networking in vehicular ad hoc networks”, In IEEE Communications Letters, 16(9), 2012, pp. 1380–1383.
[212] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, "Navigo: Interest Forwarding by Geolocations in Vehicular Named Data Networking," arXiv:1503.01713, 2015.
[213] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang, “Data naming in vehicle-to-vehicle communications," In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2012, pp. 328-333.
[214] G. Grassi, D. Pesavento, L. Wang, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, “Vehicular Inter-Networking via Named Data,” In ACM SIGMOBILE Mobile Computing and Communications, 17(3), (2013): 23-24.
[215] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, and L. Zhang, "VANET via Named Data Networking," In Proceedings of the IEEE INFOCOMM Workshop, 2014, pp. 410-415.
[216] A. Alexander, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," University of California, Los Angeles, Tech. Rep., 2012.
[217] S.H. Bouk, M. A. Yaqub, S.H. Ahmed, and D. Kim. "Evaluating interest/data propagation in vehicular named data networks," In Proceedings of the ACM Conference on research in adaptive and convergent systems, 2015, pp. 256-259.
[218] Z. Ming, H. Wang, M. Xu, and D. Pan, “Efficient handover in railway networking via named data,” In Springer International Journal of Machine Learning and Cybernetics, (2014): 1-7.
[219] J. Wang, R. Wakikawa, and L. Zhang, “DMND: Collecting data from mobiles using named data,” In Proceedings of the Second IEEE Vehicular Networking Conference (VNC), 2010, pp. 49–56.
[220] W. Drira and F. Filali, “A Pub / Sub Extension to NDN for Efficient Data Collection and Dissemination in V2X Networks,” In Proceedings of the 15th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014, pp. 1-7.
[221] W. Drira and F. Filali, “NDN-Q: An NDN Query Mechanism for Efficient V2X Data Collection in Smart Cities,” In Proceedings of the Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops), 2014, pp. 13-18.
[222] A. Attam and I. Moiseenko, “NDNBlue: NDN over Bluetooth,” Tech. Rep. NDN - 0012, 2013, pp. 1–3.
[223] Z. Qu and J. Burke, “Egal Car: A Peer-to-Peer Car Racing Game Synchronized Over Named Data Networking,” Tech. Rep, 2012, pp. 1–13.
[224] Z. Wang, Z. Qu, and J. Burke, "Demo overview-Matryoshka: design of NDN multiplayer online game," In Proceedings of the 1st ACM International |Conference on Information-centric networking, pp. 209-210, 2014.
[225] D.G. Barros, and M.P. Fernandez, "NDNGame: A NDN-based Architecture for Online Games."In Proceedings of the 2ndACM International |Conference on Information-centric networking, 2015, pp. 77.
[226] A.A. Alsaffar, and E.N. Huh, "A framework of N-screen services based on PVR and named data networking in cloud computing," In Proceedings of the ACM 7th International Conference on Ubiquitous Information Management and Communication, 2013, pp. 100.
[227] H.M.N.D. Bandara and A. P. Jayasumana, “Distributed, multi-user, multi-application, and multi-sensor data fusion over named data networks,” In Elsevier Internet Computer Networks (ICN), 57(16), (2013): 3235-3248.
[228] J. Wang, C. Peng, C. Li, E. Osterweil, R. Wakikawa, P. C. Cheng and L. Zhang, “Implementing instant messaging using named data,” In Proceedings of the Sixth ACM Asian Internet Engineering Conference (AINTEC), 2010, pp. 40–47.

[229] Y. Song, M. Liu, and Y. Wang, “Power-Aware Traffic Engineering with Named Data Networking,” In Proceedings of the Seventh IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN), 2011, pp. 289–296.
[230] M. Zhang, C. Yi, B. Liu, and B. Zhang, “GreenTE: Power- aware traffic engineering,” In Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP), 2010, pp. 21-30.
[231] T. Ogawara, Y. Kawahara, and T. Asami, “Information Dissemination Performance of a Disaster-tolerant NDN-based Distributed Application in Disrupted Cellular Networks,” In Proceedings of the Thirteenth IEEE International Conference on Peer-to-Peer Computing (P2P), 2013, pp. 1-5.
[232] I. Moiseenko, and L. Zhang, "Consumer-producer API for named data networking." In Proceedings of the 1st ACM International Conference on Information-centric networking, 2014.
[233] I. Moiseenko, L. Wang, and L. Zhang. "Consumer/Producer Communication with Application Level Framing in Named Data Networking." In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 99-108.
[234] M. Gallo, L. Gu, D. Perino, and M. Varvello, "NaNET: socket API and protocol stack for process-to-content network communication," In Proceedings of the 1st ACM International Conference on Information-centric networking, 2014, pp. 185-186.
[235] K. Wang, H. Zhou, J. Chen, and Y. Qin, "RDAI: Router-Based Data Aggregates Identification Mechanism for Named Data Networking," In Proceedings of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013, pp. 116-121.
[236] G. Piro, I. Cianci, L. Alfredo Grieco, Gennaro Boggia, and Pietro Camarda, "Information centric services in smart cities." Journal of Systems and Software, 88 (2014): 169-188.
[237] S. Susmit, A. Barczuk, C. Papadopoulos, A. Sim, I. Monga, H. Newman, J. Wu, and E. Yeh, “Named Data Networking in Climate Research and HEP Applications,” In Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015), Okinawa Japan, April 2015.
[238] C. Cabral, C.E. Rothenberg, and M.F. Magalhães, "Mini-ccnx: fast prototyping for named data networking," In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013, pp. 33-34.
[239] C. Cabral, C. E. Rothenberg, and M. F. Magalhães, “Reproducing real NDN experiments using mini-CCNx," In Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013, pp. 45-46.
[240] Z. Lailari, H.B. Abraham, B. Aronberg, J. Hudepohl, H.Yuan, J. DeHart, J. Parwatikar, and P. Crowley, "Experiments with the Emulated NDN Testbed in ONL," In Proceedings of the 2nd ACM International Conference on Information-Centric Networking, 2015, pp. 219-220.

Divya Saxena is working towards the Ph.D. degree at the Indian Institute of Technology (IIT), Roorkee, India. Her research interests include wireless communication, mobile computing, and future Internet architecture.

Vaskar Raychoudhury received the B.Tech. degree in Information Technology from the B.P. Poddar Institute of Management & Technology, Kolkata (*affiliated to The University of Kalyani, West Bengal*), in 2003, and the MS degree in Information Technology from the School of Information Technology, Indian Institute of Technology, Kharagpur in 2006. He obtained his PhD in Computer Science from The Hong Kong Polytechnic University in 2010. Later he continued his post doctoral research in the same university for a year before moving to the the Institut Telecom SudParis, in France where he worked for another year as a post-doctoral research fellow. He is currently an Assistant Professor in the Department of Computer Science and Engineering, Indian Institute of Technology Roorkee. His research interests include mobile and pervasive computing, context awareness, and (Mobile) social networks, and he keeps publishing high-quality journals and conference papers in these areas. He has served as program committee member in ASE/IEEE Socialcom, ICDCN, and many others. He serves in the capacity of the reviewer for many top IEEE and Elsevier journals. He is a member of the ACM, and a senior member of the IEEE.

Neeraj Suri received his Ph.D. from the University of Massachusetts at Amherst. He currently holds the TUD Chair Professorship at TU Darmstadt, Germany and is also affiliated with the Univ. of Texas-Austin and Microsoft Research. His earlier appointments include the Saab Endowed Chair Professor-ship, and Professor at Boston University. His research spans distributed systems, mobile computing and OS’s tackling the design, analysis and assessment of trustworthy web scale services.

Christian Becker received his Ph.D. in Computer Science from the Universität Frankfurt. He currently holds the Chair for Information Systems II (Wirtschafts informatik II) Universität Mannheim, Germany. His research spans Self-Aware Computing, Adaptive Systems, especially Proactive Adaptation, Context-Aware Computing, Smart Cities, especially traffic and power management, and Middleware and communication protocols.

Jiannong Cao is currently a chair professor and head of the Department of Computing at Hong Kong Polytechnic University, Hung Hom, Hong Kong. He is also the director of the Internet and Mobile Computing Lab in the department. Dr. Cao's research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He has co-authored a book in Mobile Computing, coedited 9 books, and published over 300 papers in major international journals and conference proceedings. Dr. Cao has won numerous prizes and awards, and is very active in professional activities locally and internationally. He is a senior member of China Computer Federation, a senior member of IEEE, and a member of ACM. He was the Coordinator in Asia and now the Chair of the Technical Committee on Distributed

Computing of IEEE Computer Society. Dr. Cao has served as an associate editor and a member of the editorial boards of many international journals, including IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Networks, Pervasive and Mobile Computing Peer-to-Peer Networking and Applications, and Journal of Computer Science and Technology. He has also served as a chair and member of organizing / program committees for many international conferences, including PERCOM, INFOCOM, ICDCS, IPDPS, ICPP, RTSS, DSN, ICNP, SRDS, MASS, PRDC, ICC, GLOBECOM, and WCNC. Dr. Cao received the BSc degree in computer science from Nanjing University, Nanjing, China, and the MSc and the Ph.D degrees in computer science from Washington State University, Pullman, WA, USA.