EPF—An Efficient Forwarding Mechanism in SDN Controller Enabled Named Data IoTs

Asadullah Tariq and Rana Asif Rehman and Byung-Seo Kim

1 Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan; realasadullahtariq@gmail.com (A.T.; r.asif@nu.edu.pk (R.A.R.)

2 Department of Software and Communication Engineering, Hongik University, Sejong City 30016, Korea

* Correspondence: jsnbs@hongik.ac.kr

Received: 1 September 2020; Accepted: 20 October 2020; Published: 30 October 2020

Abstract: The vision of the Internet of Things (IoT) is that it connects all kinds of things by leveraging the creation of increasingly affordable and small devices that can be embedded for sensing, processing, wireless communication, and actuation. Named data networking (NDN) is a newly emerging Internet paradigm that may replace the current Internet architecture and that fulfills most of the expectations of the IoT. Software-defined networking (SDN) is an emerging paradigm of technology that is highly capable of managing overall networks efficiently and transforming complex network architectures into manageable, simple ones. The combination of the SDN controller, NDN, and IoT can be lethal in the overall performance of the network. Broadcast storms, due to the flooding nature of NDN’s wireless channel, are a serious issue when it comes to forwarding interest and data packets. Energy consumption of sensor nodes in dense IoT scenarios causes problems in forwarding as well as unnecessary delays, decreases network performance, and increases the cost and packet delay for important packets. We took these problems as our baseline and proposed an energy-efficient, priority-based forwarding (EPF) in SDN-enabled NDN–IoT. Our scheme EPF used the efficient flow management of the SDN controller to control the broadcast storm and efficiently forward the priority-based packets. A defer timer mechanism was used to prioritized the packet upon its arrival to the node. An energy threshold mechanism was used to control energy consumption and improve overall energy efficiency. We compared our scheme with the traditional flooding mechanism and geographic interest forwarding; EPF outclassed the other schemes and produced the best results in terms of total number of interests and retransmissions, content retrieval time, total number of priority interests, energy consumption, and network lifetime.

Keywords: named data networking; software defined networking; Internet of Things; energy; internet

1. Introduction

Internet of Things (IoT) technology is evolving and starting to make a huge impact in a widespread range of applications, including industrial environments and smart cities. Such real world IoT applications combine communication, sensing, computation and actuation to remotely control and monitor the environment. Gartner states that the number of IoT devices will reach 20.4 billion by 2020 [1]. Even human beings will connect to devices and each other through IoT in the near future. There is a great market opportunity offered by the IoT for Internet service providers, manufacturers, and application developers. Figure 1 shows the wide range of IoT applications. There is a critical need for a flexible, layered architecture for the IoT because that can connect the trillions of heterogeneous devices through the Internet. Realizing the IoT vision using a host-to-host (H2H) communication standard Internet protocol (IP) is challenging in the context of constrained sensing and smart devices.
that connect densely with one or more networks, are highly heterogeneous, and frequently use several interfaces [2]. There is a need to correspond with numerous security administrations and have both global and local communication ability. It is challenging for the IP to deal with this increase in devices while still integrating the minimum resource usage [3].

Figure 1. Internet of Things (IoT) applications.

Named Data Networking (NDN) is a new and prominent communication paradigm. IoT based data communication efficiency can be improved using advantages of NDN. Communication in NDN is content centric. The main motivation behind the vision of NDN is the limitations of current internet architecture [4]. Both NDN and IP use the same hourglass architecture, end-to-end principle, their own namespace and send datagram for the delivery of data [5]. Named Data layer in NDN hourglass model consist of Content store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB). The NDN hourglass architecture is presented in Figure 2.

Figure 2. Named Data Networking (NDN) hourglass model.

Named data networking as an emerging paradigm on the Internet fulfills most of the expectations of the IoT. There are always points to address among the many open aspects of research. Dense and
different traffic patterns of the IoT system compared to current Internet applications, such as video streaming and web browsing, need a proper procedure to identify the devices, services, and management [6]. IoT devices are resource constrained ones that use little power and energy, but they do not provide reliable communication in the wireless environment, so the optimization techniques for reliable data delivery must be there in NDN.

Software Defined Networking (SDN) is an emerging technology that makes computer networks more programmable. SDN is offering new ways to interact within networking devices by adding flexibility to existing networking operations and designs. SDN provides an innovative approach towards the design, implantation and management of networks that separates the control plane and data plane. SDN provides a centralized control plane for the control and management applications, and removes the decision-making part from the router that makes it different from traditional networking [7]. Two primary components of SDN are the controller and switches. Management of the entire network is the responsibility of the SDN controller while, on the other hand, instruction-based operation is done by the switches. Software defined networking is a good solution for emerging technologies and efficient resource utilization and communication [8].

There are open problems and challenges, like scalability, integration, management of flow table, and real-time SDN updates. In urban scenario, hundreds of IoT devices, actuators and sensors are sensing and transmitting the data for processing in an heterogenous environment. In IoT environment drastically increase in population and IoT devices makes communication and data management more challenging [9]. Real-time efficient management and retrieval of data in the IoT environment is a challenge that can be solved using SDN [10]. Software defined networking can benefit the IoT system because the control intelligence moves to the centralized controller and is removed from the data plane devices. The controller itself interacts with the devices of the data plane. The SD-IoT architecture consists of three layers: the physical infrastructure layer, the application layer and the control layer [11]. The SDN architecture is presented in Figure 3. There are benefits for environment communication and performance when these three paradigms—NDN, SDN, and IoT—are connected to develop a new joinable SD-NDN–IoT architecture.

**Figure 3.** Software Defined Networking (SDN) architecture.

Forwarding strategies in References [12–14] used SDN controller to forward the packets. Some forwarding strategies used native flooding mechanism aided with controller to forward the packet that brings high overhead and broadcast storm issues on network. There is no strategy to prevent the energy consumption with the use of controller in related work. They did not handle the priority and emergency packets efficiently in WSNs and IoTs. In References [15–17], proposed forwarding mechanisms to mitigate the broadcast storm issue but lack of work to find efficient
forwarding path and more overhead on the controller. In References [18–20], authors proposed energy efficient forwarding schemes in IoT. Proposed energy efficient mechanism in this paper results in improved average energy consumption and network life time. In this paper an efficient forwarding mechanism in NDN based IoTs is proposed using the centralized controller of SDN. The main purpose of our proposed scheme is to ensure the in-time and fast packet transmission of high priority packets in dense IoT environment with maximum satisfaction rate. The focus is on the energy efficiency of the whole network to guarantee the maximum output. This paper is organized as follows: Section 1 introduces the three paradigms—IoT, NDN, and SDN. Section 2 provides a brief explanation of NDN, and its architecture with IoT is presented. Section 3 consists of the motivation behind integration of SDN and ICN along with the advantages of SDN-NDN-IOT combination. Section 4 consists of related work. Problem scenarios are presented in Section 5. Section 6 proposes a scheme and discusses it in detail. Finally, Sections 7 and 8 depict the simulation graphs and conclusion, respectively.

2. Background

2.1. Named Data Networking

Named data networking (NDN) is an emerging Internet paradigm. Current Internet architecture is the base line behind the design principles of NDN. Design principle of NDN reflects our understanding towards the strength and the limitations of current internet architecture [4]. Communication in NDN focuses on content rather than content container and is based on content names. NDN offers innovative and advanced way to think about the internet and making a beneficial impact on the overall performance of the network. Separation of forwarding and routing planes, built-in security, state full forwarding, and end-to-end principle are the key features of NDN architecture [5].

2.1.1. Named Data Networking Architecture

NDN is based on the receiver-driven communications [21]. Two types of packets are used in NDN communication: Interest and Data packets [4]. The NDN router has three main architectures: forwarding information base (FIB), pending interest table (PIT), and content store (CS). An FIB table is used to forward the interest packet to the producer node by routers. The PIT keeps track of the interest packet’s path. Aggregation and congestion control mechanisms can also be supported by the PIT. The CS is a temporary cache for the data objects. In the CS, all the satisfied interests and content entries are present.

2.1.2. Named Data Networking Communication Model

In NDN, the consumer node first sends an interest packet requesting data to the data publisher node using receiver-driven communication. When a node receives interest, it first checks its CS and matches the recently satisfied interest entries and the content names entries. If the data or content names match, the data will be forwarded to the consumer node that requested the data. In the case of no entry being found in the CS table, interest will go to the PIT. If an interest matches the PIT entries, then it will drop the recently requested interest because some other interest with the same request has already been forwarded to the network. If there is no matching entry in the PIT, then, after entering that interest entry, it will go to the FIB where the incoming and outgoing faces forward the interest and data for content retrieval toward the data producer.

After finding the data from the producer, the producer adds the data into the packet with its name and signature. Producer signatures authorize and secure the content. The producer sends the data packet to the consumer following the trail of the interest packet. Pending interest table entries help the data packet keep track of the path of interest packet. Data centricity, in-network discovery of name, data security, flow balance (hop by hop), data centricity, data immutability, universality, and hierarchical naming, is the main design principles of NDN. A complete forwarding and communication model of NDN is presented in Figure 4.
2.2. NDN–IoT Architecture

A high level of NDN–IoT architecture was devised in References [22,23]. The NDN–IoT architecture consists of three layers: the things layer, the NDN layer, and the application layer. Figure 5 shows the NDN–IoT architecture. The things layer represents the multitude of devices in the IoT system. The NDN layer is the networking layer that hides the complexity and diversity of IoT applications from the users. The NDN layer in NDN–IoT architecture consists of two main components: management and the control plane and data plane. The control layer handles the operational functionalities of both interest packets and data packets, including security, naming, caching, and the strategies that adequately meet the IoT features.

3. Motivations of Integrating SDN and Information Centric Networking

Although excellent research work has already been presented in SDN and ICN separately, these two promising futuristic paradigms are complementary, benefiting each other by tackling different kinds of networking problems. The research community has started investing their efforts in SDN and ICN. Benefits of SDN over current Internet, including centralized control, powerful data
management, and monitoring tool and global views of the whole network, also apply to ICN [24]. ICN can handle distribution of data in upgraded networks, smooth the transition phase, and simplify the deployment with the help of SDN. A lot of surveys and research ideas have been proposed in SDN and ICN on issues of energy efficiency [25], security, mobility, caching, and forwarding [26]. The centralized SDN controller has powerful control of data transfer as well as a global view of the status of network. SDN is a promising way to manage networks simply. The combined SDN-ICN can benefit the network by opening new opportunities after shifting the question from “where” to “what.” SDN can take full advantage of the main feature of ICN—that is, in-network caching. SDN is able to exploit the multipath routing and in network caches. Routing of content can be optimized because SDN can monitor in network caches efficiently. New data centric services can be enabled using SDN over ICN. ICN can remove the security threats over northbound and southbound communication in SDN due to its nature of protecting content as opposed to communicating between channels. SDN can provide powerful control over distributed elements of ICN. Centralized management, revocation of keys, and distribution of SDN can strengthen the security of ICN. SDN can benefit ICN with its agility and programmability in named based forwarding.

**Software Defined Networking enabled Named Data IoTs (SDN-NDN-IoT)**

SDN enabled NDN–IoT architecture is an emerging research combination that uses the benefits of NDN with IoT systems to improve efficiency using the SDN model. High-priority messages can support NDN–IoT in case of any alarming situation by notifying governments, emergency departments, hospitals, or law enforcement authorities using push-based forwarding in NDN. Support of periodic and event triggered data and content pushing must be overhauled in NDN–IoT architecture. Device configuration procedures and device discovery is an important requirement in the IoT system. In NDN–IoT architecture, service publish and discovery protocols (SPDPs) [27] and neighbor discovery protocols (NDPs) [22] can be use to discover and configure the device in the IoT system. In the global IoT environment, named-based routing of NDN appears to be extremely useful. Packets can communicate in the network without resolving any network addresses because of the uniqueness of the content name. Specific application-based naming can clearly benefit the NDN solution for the IoT. The current IoT system needs IoT names to be able to identify the services, tasks, and applications ensured by the device. For example, sensor type, sensor position, type of the data, and time when data is needed. User readability and user-friendly NDN names can cause serious security-related issues in IoT applications such as spoofing. NDN offers an amazing solution for the secure content dissemination in the IoT environment by putting publisher signatures in the data packet. In NDN–IoT, there is a need to develop efficient authorization mechanisms, trust models, and access control policies to support security and minimize risks. In-network caching in NDN can be crucial to provide the scalability, readability, and robustness to the IoT environment [28]. The NDN strategy layer allows the named data network to interact with the IoT. The IoT data passes through heterogeneous devices during communication within the network, so NDN–IoT architecture must develop a mechanism to support the heterogeneous environment of the IoT system. A combination of NDN and IoT is a good solution that benefits the communication of smart devices and enhances the overall performance of the IoT system. Named data networking takes care of the security of data packets by securing them on the network layer. A producer signature, binding data payloads and data names in a cryptographic way, authentication and reconstruction chains for the verification of the data and recorded information, and signing key certificates in key information locators are methods used to secure the data.

The name-based forwarding in NDN enables the IoT and named data networks to communicate directly with the packets, unlike the IP network. Name-based stateful forwarding can benefit other features of the IoT system, like fast local recovery, congestion control (hop by hop), and delay tolerant communication style. The in-network storage ability of NDN is a big advantage to IoT devices: in case of sensors deployment with limited storage capacity in an agriculture field, sensors can quickly transfer the data they monitor to the nearby repository. A controller can retrieve the desired data from
the repository as well. The in-network storage can reduce overhead from the network because the nodes can cache the data packets they forward and can efficiently cache the popular data and fast local recovery.

By Default, SDN does not support Sensor and IoT networks. With the decoupling of control plane from data plane, SDN benefits the IoT with intelligent management. One or multiple SDN controllers can be there in IoT environment, in which devices can be manage in the range of specific SDN controller. To attain the resource management and flexibility benefits, many authors proposed the concepts of root controller and hierarchical controller. Energy consumption is a huge problem for companies around the world because of the need for continuous availability and the enormous architecture of IoT devices. Companies spend a huge amount of revenue to solve this problem of energy consumption. One solution is SDN, which can reduce the consumption of overall energy. The programmability, openness, and feasibility of SDN reduces the implementation complexity for software and hardware. It is a candidate by which we can limit our resources, and reduction in resource usage in turn reduces overall energy consumption.

Energy-saving techniques and power-saving schemes can be applied to reduce energy consumption. Energy consumption is not consistent at all times. For example, during the day, Internet traffic is high while at night, traffic is significantly reduced. In scenarios with low Internet traffic, most of the nodes are unutilized. Flexible programming in SDN can solve this problem and can make intelligent decisions according to the situation of the traffic. Unutilized nodes can be sent to sleep mode while the activation period in network for the nodes can be defined through SDN. A content-based method can also be a solution for SDN resource management. This method ensures reduced power consumption with proper resource utilization because content length can be known to the forwarder in NDN and ICN, so the required resources can be delivered to the most fitted nodes. Named data networking can also be beneficial to the SDN–IoT combination. Load balancing and flow management techniques in the SDN can reduce overall energy consumption. Traffic management based on information from the nodes and switches can reduce collisions of interest and data packets. Constant and timely updates to the SDN controller about energy consumption and resource allocation can enhance network performance against energy consumption, interest satisfaction rate and network life time. Finally, the combination of SDN and IoT with NDN can be a great addition to research advancements and the development of science because of their fruitful results.

4. Related Work

Current NDN architecture updates its FIB using flooding mechanism. Software defined networking based NDN architecture reduces the effect of such problem on consumption of network resource. An SDN-based routing scheme (SRSC) in Reference [12], also proposes a NDN-based forwarding scheme over SDN. Then a control process controls and manages the FIB with the help of an SDN controller at the node level. Controller announcement and neighbor discovery are the two phases at the controller level. In this routing protocol, flooding is used to forward packets. This scheme has no efficient forwarding mechanism but has a controller attachment only.

A controller-based routing strategy for NDN (CRos-NDN) in Reference [13] is another SDN controller–enabled scheme. The aim of CRos-NDN is to define specific procedures and names for efficient forwarding. It consists of two processes, bootstrapping and named data routing. During the bootstrapping process, a hello packet is transmitted across the whole network. Nodes connect with the controller after recognizing it. Neighboring routers acquire information by sending an interest packet. The routing step gives permission to fill the FIB nodes, which then send the packet as relay nodes. CRos-NDN improves the mobility efficacy.

In Reference [14], NDN improved the forwarding process with topology discovery and the management of tables. It used all three tables: CS, PIT, and FIB. It is a waste of time if there is no name prefix in the CS when dealing with huge tables. Named data networks improve the forwarding mechanism by adding the FlowFIB table, which minimizes the access number for prefixes and tables.
There are two types of action, local and external, after redirecting the interest toward FlowFIB. Local action is all about the existence of a prefix in the CS. External action refers to the port that follows the forwarding process after prefix matching. Named data networks resolve problems like consumption of bandwidth, scalability, and delay. IT outperforms SRSC and CRoS-NDN. The above discussed forwarding strategies do not handle the priority packets, communication, or energy consumption of sensor nodes. There is no efficient forwarding model in the above discussed strategies.

Adaptive forwarding based link recovery for mobility support (AFIRM) in Reference [23] is an efficient novel solution for mobility issues among producers and makes the data availability more effective. The focus of AFIRM is on data availability requirements.

Markov Decision Process Based Interest Scheduling (MDP-IOT) in Reference [29] is an efficient technique to satisfy IoT applications that are delay intolerant. In this scheme, the interest schedule moves towards the right interface with low latencies after prioritizing the IoT traffic that results in less RTT.

Controller based Selective forwarding (CSF) in Reference [15] is an efficient scheme to mitigate the broadcast storm problem using forwarding flows calculated in SDN controller. Consumer nodes can request these forwarding flows whenever needed, and only selective nodes perform forwarding instead of the native flooding mechanism. Broadcast Storm Avoidance Mechanism (BSAM) in Reference [16] is a distance based forwarding mechanism that controls the packets propagation using controller. Each node calculates a counter value by calculating its distance from a sender node. The farthest node will then forward the packet. SDN controlled VNDN in Reference [17] is an efficient forwarding mechanism that uses SDN controller for greater view of network for user. Packet classification triggers the Emergency packet dissemination. Road side unit has been equipped with Policy based bifold classifier for packet classification. Hybrid swarm optimization and cuckoo search algorithms are used for the best path selection. In Reference [30] an SDN aided forwarding strategy in ICN is presented that copes with the mobility issues and dynamically updates the controller whenever a consumer or a producer detaches from the network.

SD-EAR in Reference [18] is an energy efficient software defined forwarding mechanism. In the mechanism, network is divided in multiple clusters, controlled by SDN controller. Each controller already knows the topology inside the cluster zone. Outside the cluster, SDN is in the charge of controlling nodes to send their packets. SDN controller discovers the optimum energy efficient path after checking the residual energy of the nodes and flow tables. In this paper, a sleeping mechanism has been used for the exhausted nodes.

GIF in Reference [19] proposes an energy efficient forwarding strategy. Sink nodes are used to keep record of neighbor ID, position and remaining energy of all the nodes in cluster. In this scheme, the sink node having more resources than the sensor nodes, but, the high loads in the sink node results in the more energy consumption and eventually in the death. Such sink nodes’ death causes to path loss and packet drops. There is no such criteria to prevent the energy of sink nodes, having most of the information of the cluster that will cause serious management problem.

The DMIF in Reference [20] comprises of two collective forwarding modes, in which several energy efficient mechanisms including flooding scope control, flexible mode shift, packet suppression, broadcast storm avoidance and energy weight factors are designed to balance and save the energy consumption.

5. Problem Statements

In this section, problems in wireless networks, IoT, and NDN are discussed. These problems include broadcast storms because of the flooding nature of wireless channels and NDN. Energy problems, which cause a decrease in network performance, cost and delay important packets, are discussed as well. Some NDN-based IoTs can take befit from the broadcasting nature of NDN and the wireless medium to propagate interest packets. More than one neighboring node can listen to the packet transmission of an interest. These neighbors are the possible relay nodes and intermediate
forwarders. These multiple forwarders increase the probability of packets being forwarded to the producer. The way content is requested, and the broadcast nature of the wireless medium, imposes challenges to the deployment of NDN-IoTs, particularly the “broadcast storm problem.” For NDN, after an interest is transmitted and received by neighbor nodes, the nodes will repeat the same process. Nodes are equipped with one wireless interface. In this scenario, the same interface is used to forward the packet, and all the nodes that are in the transmission range will receive the packet, even the nodes that have already received it. This leads to the broadcast storm problem. In this case, the consequences increase collisions and medium access time. Nodes using the same path can collide. The broadcast nature of NDN makes it even harder for the development of NDN based IoTs. Figure 6 shows the scenario of broadcast storm problem. In our scenario case, C (consumer node) broadcasts an interest packet to P (Producer node). Node A and B receive the interest and broadcast it further. Then all the other nodes, C, D, E, F, G, and H receive the interest and broadcast the same request in the network in same manner that causes a broadcast storm issue.

![Diagram](Figure 6. Broadcast storm, high energy consumption and delay problems in IoT scenarios.)

Energy is one of the most critical aspects in real deployments of sensor networks and IoTs. A sensor node participates in sensing, processing, and communication subsystems. Optimization of energy consumption in sensor networks is generally a hard task, since it effects on maximizing the network life time along with reduction in energy consumption. The minimizing of sensor networks’ power consumption is becoming increasingly significant, especially in situations where many countries are independently developing programs to improve the efficiency of power networks and power supplies. The energy efficiency and power consumption of various devices are defined in several different conditions of work. In the case of load applications, it would imply efficiency at different levels relative to the rated power of the device. The energy consumption of the sensor nodes largely depend on protocols the sensors use for communication.

In an IoT system, the access communication of the regular and periodic interest and data packets sensed and sent by the sensor nodes can consume most of the energy and resources of the sensor nodes and network [31]. In fact, in IoT scenarios, the less important regular sensing data flow regularly in access in the network and probability of emergency data with high priority is very low. This scenario can be a problem for the high-priority and reactive interest packets and data packets. So efficient routing and forwarding mechanisms are needed to preserve the energy and power of sensor nodes, enhance the network’s lifetime, minimize the overall energy consumption, and improve overall network performance. Figure 6 shows the sensor node’s energy problem. In our scenario case, consumer node C broadcasts a less important interest packet to producer node P. Nodes A and B receive the interest and broadcast again. Then nodes C, D, E, F, and G receive the interest and broadcast
the same request in the network. The communication of the less important packet consumes the energy of the sensor node. When a high-priority packet arrives, the node energy of E goes down very critically because the regular flow of low priority packet consumes most of the energy of the node. Hence, the packet faces a path loss or packet drop issue. Large data packets create an unacceptable delay for important packets because of the head-of-line (HOL) problem. The broadcast storm problem caused by the broadcast nature of wireless networks as well as the NDN forwarding model and the energy problem caused by the access in periodic movement of unimportant sensed data can be harmful to the in-time, efficient, and fast forwarding of the important, high priority, and emergency interest and data [22]. Figure 6 show the delay problem in which low priority packets create the delay problem for high priority packets on node E. The emergency and critical messages have higher priority than a chat between friends or the regular patterns of the traffic.

6. Energy-Efficient, Priority-Based Forwarding (EPF)

To cope with the broadcast storms and communication delays in the network, and to improve overall energy consumption, an Energy efficient Priority based Forwarding strategy has been proposed in this paper. Two main architectures are presented in our system model: 1. Node side architecture and 2. Controller side architecture.

6.1. Node Side Architecture

Like the traditional NDN architecture, Node side architecture contains CS that keeps the record of the content. The PIT keeps the information of unsatisfied packets. The additional part of EPF is its Local Information Table (LIT) and priority forwarding table. The LIT stores information of controller, neighbor IDs, residual energy and content names. Figure 7 shows the node side architecture.

![Node side architecture in energy-efficient, priority-based forwarding (EPF).](image)

6.2. Controller Side Architecture

Controller side architecture in Figure 8 consists of the Routing Information Table (RIT), FIB and Global Information Table (GIT). In the GIT, the entries are against each node’s ID, neighbor nodes, residual energy of the nodes, and content names presented in the CS. The RIT in the controller computes the best and the shortest possible path toward the producer node by keeping the track of the hop count to a minimum.
6.3. Packet Formats

Four types of packets are used in the EPF. Controller ID packets (C_ID pkt) contain information on the controller ID. This packet is broadcasted once at the start of the topology. New nodes in the network can communicate with neighbor nodes and send their information to the controller node after syncing with it. Local information packets (L_info pkt) have information on the node’s ID, the node’s residual energy, and the content names, which are broadcasted to the controller node and neighbor nodes for the population of the GIT and LIT. Interest packets in EPF (EPF I_pkt) have a naming scheme in which some priority criteria is presented: the path information, requested content name, and the nonce for the unique identifier of the packet. Data packets in EPF (EPF D_pkt) have the same structure as EPF I_pkt with addition of Producer’s signature. Figure 9 shows the packet formats for EPF.

6.4. System Model

6.4.1. Controller Announcement

When the network starts, the controller announces its presence in the network by broadcasting the C_ID pkt. All the nodes update their LITs after receiving the packet, as shown in Figure 10.
6.4.2. Topology Discovery and Content Placement

At first, all the nodes broadcast \( L_{\text{info}} \) pkt to their neighbor nodes (in one-hop network range) after controllers’ announcement with its ID. After updating LITs, all the nodes then send \( L_{\text{info}} \) pkt to controller having information of their content names, ID, neighbor ID and the remaining energy of the nodes by enabling controller ID. Controller then updates the table entries of its GIT using the information sent by each node in \( L_{\text{info}} \) pkt. After GIT setup, the actual forwarding mechanism starts. This whole process of topology discovery is presented in Figure 11 and Algorithm 1.
Algorithm 1: Topology Discovery and Controller Table Building

Require: Node \( N \), Controller SDN;
SDN.broadcasts(C_ID pkt);
if \( N.already.Bind.To.SDN() \) then
    \( N.drops(C_ID pkt) \);
else
    \( N.BindsTo(SDN) \);
    \( N.Updates(LIT) \);
    \( N.Broadcast(L_Info pkt) \);
    For all \( L_Info pkt \) received from \( N \);
    SDN.Builds(GIT);
end

6.5. EPF Forwarding Model

There are two communication modes in the IoT system discussed in earlier sections: a pull-based communication model that needs an interest request to respond with a data packet and a push-based communication model in which sensing nodes in the IoT system itself sense, collect, process and send data to the consumer without any prior request for that data.

When the consumer node sends an interest packet for content, the interest is sent to a neighbor node for content discovery and retrieval. The node matches its CS against the requested content. If data is found, the producer node sends data back to the consumer node. If there is no entry in the PIT or the CS, then the consumer node itself retransmits the packet to the controller to discover a producer in an efficient way. After receiving the interest packet, the controller checks its GIT for the matching content name and the producer of that content. After matching the content in the GIT, the controller computes the best possible and shortest path between the consumer and producer using the Dijkstra algorithm. In this computation, the controller verifies the residual energy of the potential forwarder with the content type and the active network status. The Dijkstra algorithm is the classical single-source algorithm that is always optimized in computation [32]. After the computation and path establishment, the controller sends the packet back to the consumer node after embedding the path information in the packet. The consumer sends the interest packet to the producer, and every potential forwarder responds to the request after matching the path-naming scheme. All the nodes that do not match the path naming will drop the packet, while other nodes will send data back toward the consumer or forward the packet toward the next hop.

As discussed in detail, the regular and unimportant sensing data is much more present in the IoT environment than high priority and important data. Important information is rarely generated after time in the network, while the regular flow of the traffic in the IoT system can cause problems for the sensor node’s energy. Therefore, the arrival of high-priority information can be met with problems in in-time and efficient delivery. Government-related information, law enforcement agency-related information, high red-alert alarms, emergency situations, road accidents, and theft are scenarios of high priority; chatting, traffic patterns, and temperature are low-priority information. We categorized the five different types of data according to priority type in Table 1.
Table 1. Priority Assignment Table.

| Packet Type     | Priority Type |
|-----------------|---------------|
| Government      | High          |
| Emergency       | High          |
| Police          | High          |
| Regular Sensing | Low           |
| Chat            | Low           |

Potential forwarders check the name embedded in an interest and data packet to differentiate the high and low priority packets. Naming schemes match the name in packet headers to the priority level, as defined in Table 1. An energy threshold is set to achieve reliable transmission of high priority interest and data packets. We set the energy threshold value as 16%. Only high priority packets can forward through nodes beyond defined energy thresholds. GIT will update residual energy when the node’s residual energy reaches the threshold after getting information from \( L_{\text{info}_\text{pkt}} \).

In our scheme, the priority mechanism is performed by using the timer-based rebroadcast mechanism for both interest and data packets. A defer timer value is set locally by each forwarder after evaluating the priority level that is embedded in packet header. The value of the timer is computed in a fully distributed way by each processing node without incurring any signaling overhead. Interest and data packet communication are described in Algorithm 2 and discussed above in EPF forwarding model.

Algorithm 2: Received Interest/Data Packet

Require: Node \( N \), Intermediate \( I \), Controller \( \text{SDN} \);
On Incoming \( EPF\ I\ _{pkt} \);

\( N \) computes priority using Priority.Naming;
if \((\text{Content}!=\text{PIT} \&\& \text{Content}!=\text{CS})\) then

- \( \text{PIT}.\text{add}(EPF\ I\ _{pkt}) \);
- \( \text{EPF}\ I\ _{pkt} \) sends to \( \text{SDN} \) enabling \( C\_ID \);
- \( \text{SDN}.\text{search}(\text{GIT}) \) and Content found in GIT Then ;
- \( \text{SDN} \) calculates shortest path (Dijkstra Algorithm), Keeping energy threshold in calculation;
- \( \text{SDN}.\text{embeds}(\text{shortest path}(EPF\ I\ _{pkt})) \);
- \( \text{SDN}.\text{Sends}(EPF\ I\ _{pkt} \text{ to } N) \);
- \( N.\text{broadcasts}(EPF\ I\ _{pkt}) \);

else

- Content!=GIT;
- Stop \( EPF\ I\ _{pkt} \) transmission;
- Remove \( EPF\ I\ _{pkt} \);
end

\( EPF\ I\ _{pkt} \) matches the forwarding path;
if (forwarding path Mathes) then

- Forward packet towards producer;
else

- Drop(\( EPF\ I\ _{pkt} \));
end

In order to assign priority to Data packets over Interest packets, two adjacent and distinct time windows are used to initialize the defer timers for Interest and Data forwarding. The above said time
windows are Interest Defer Window (IntDW) and Data Defer Window (DataDW). The range of DataDW is \([0, \text{DataDWB}]\) and range of IntDW is \([\text{DataDWB}, \text{IntDWB}]\). DataDWB and IntDWB are window bounds for DataDW and IntDW respectively. The sizes of DataDW and IntDW are set according to the short inter-frame spacing (SIFS) and 802.11 slot time to adapt MAC dynamics quickly.

The aim of proposed forwarding scheme is to give High Priority (HP) Interest and Data packets, priority over Low Priority (LP) Interest and Data packets. For this purpose, we further split DataDW and IntDW into two separate sub-windows. In first sub-window of DataDW, only HP data packets can communicate using high priority timer (HPDT) in range of \([0, \text{HPDB}]\). The transmission of LP data packets will take place in second sub-window by selecting low priority timer (LPDT) in the range of \([\text{HPDB}, \text{DataDWB}]\). In IntDW, we used same mechanism in which only HP Interest packets can transmit using high priority timer (HPIT) in range of \([\text{DataDWB}, \text{HPIB}]\) and LP Interest packet transmission will take place in second sub-window by selecting low priority timer (LPIT) in the range of \([\text{HPIB}, \text{IntDWB}]\). HPDB and HPIB are the bounds for high priority data and interest packets respectively. The complete defer window mechanism described in Figure 12.

This is the responsibility of each node to compute HPDB and HPIB dynamically according to the overheard traffic load for each priority level. To compute HPDB and HPIB, each node used to keep the track of total received data bytes of HP and LP. RHP and RLP are the total number of high priority and low priority bytes received. Any node can easily infer the RHP and RLP from packet-size field in NDN packet’s header. The bound between high priority and low priority (data packets and Interest packets respectively) in DataDW is dynamically computed as:

\[
\text{HPDB} = \frac{\text{RHP}}{\text{RLP}} + \text{RHP}.\text{DataDWB} \\
\text{HPIB} = \frac{\text{RHP}}{\text{RLP}} + \text{RHP}.\text{IntDWB}.
\]

Higher number of HP results in higher HPDB and HPIB. We can compute the LPDB and LPIT analogously. After computation of defer window bounds each node needs to calculate (HPDT, HPIT, LPDT and LPIT for high priority and low priority Data and Interest packets respectively), whenever a node wants to forward the packet. If we use the aforementioned defer timer scheme then we can forward the high priority data and Interest packets earlier than the low priority ones. This mechanism reduces the unnecessary delay in forwarding of high priority packets. Figure 13 shows the Complete Forwarding flow of EPF.

![Figure 12. Priority defer Window.](image-url)
7. Simulations and Results

In this section, we presented a brief discussion about comparative schemes and simulation results. We evaluated the simulation under different performance metrics and simulation parameters. We compare our work in EPF with GIF and traditional flooding in NDN and evaluate the performance of our scheme. Selection of the best suitable threshold value for energy efficient strategy has also been presented in this section. We compared our scheme with an efficient forwarding scheme named Geographic Interest Forwarding (GIF) [19]. In GIF, the support for Push based traffic is provided for NDN-IoT environment. Forwarding strategy in GIF is designed to enhance the network performance in term of energy consumption and scalability of sensor devices in IoT environment. In this scheme, the sink node having more resources than the sensor nodes, but the maximum load on the sink node results in the resource consumption. Whenever a node sensed some data, after Producer table entry in sink node, the push based forwarding consume a lot of energy. Sensor nodes can die very early in GIF because of the regular passing of the interest and data packet. Path loss issue can be there because of the dead sensor nodes. If the sink node will die and consume its energy then there is no criteria to preserve recent entries in the sink node and define another sink node. This paper is the most relevant to our proposed EPF. There are papers that discuss controller-based forwarding, but there is a lack of papers in the IoT domain that handle SDN controller-based communication with NDN-enabled communication. Some papers present the architecture of SDN-enabled communication, but still do not describe a forwarding mechanism. Others provide forwarding schemes related to SDN–IoT but in VANETs related IoT, not for WSNs and IoT. For the comparative study and analysis, we selected the GIF scheme because it was the most relevant to our simulation parameters and provided us with a broader and clearer view of the results of our scheme.

7.1. Simulation Model

The aim of the simulation is to evaluate whether the EPF based on the SDN–NDN–IoT system can provide us with better results in terms of network efficiency, efficient forwarding, and energy consumption of sensor nodes. To that end, the implementation of our code is in ndnSim and evaluation of EPF is with the GIF and native-flooding mechanism of NDN. In our simulation environment, IEEE 802.15.4 MAC protocol used. We deployed sensor nodes in a lattice and grid topology. The grid
size is $10 \times 10$ nodes. The transmission range of each node was 100 m. The transmission network range was 1000 m $\times$ 1000 m. We evaluated our simulation results over an average of 20 independent runs and computed the results after that. For energy related computations, energy model of NDN is used in the simulation scenario. The initial energy of the nodes was 50 joules. Implemented simulation parameters are shown in Table 2.

Table 2. Simulation parameters.

| Simulation Parameters                  | Values          |
|---------------------------------------|-----------------|
| Simulation time (s)                   | 1000 s          |
| Field size (m$^2$)                    | 1000 $\times$ 1000 |
| Number of nodes                       | 100             |
| Number of Consumer/Producer pairs     | 1–5             |
| Communication range (m)               | 100 m           |
| Interest size (byte)                  | 70              |
| Data size (byte)                      | 120             |
| Initial energy (joule)                | 50 J            |
| Energy Model                          | Energy Model    |
| Rx sensitivity                        | $-86$ dbm       |
| Tx Power                              | 18 dbm          |
| Power Consumption (tx)                | 1.74 W          |
| Power Consumption (rx)                | 0.9603 W        |
| Power Consumption (idle)              | 0.6699 W        |
| Simulations run                       | 20              |
| Energy Threshold                      | 16%             |
| Data rate                             | 0.25 Mbps/250 Kbps |
| Technology                            | IEEE 802.15.4   |

7.2. Performance Metrics

To evaluate the simulation, six different performance metrics were used in our results:

- Total number of interests (NI): The total NI packets transmitted in the network from the consumer side.
- Number of high-priority interests (HPI): The total number of high-priority interests successfully communicated in the network.
- Total number of Retransmissions (RTx): RTx shows the total number of retransmitted interest packets.
- Average energy consumption (AEC): The average power consumption of all the sensor nodes in the network.
- Content retrieval time (CRT): The time it takes for the consumer to send an interest and receive data.
- Network lifetime (NLT): The time when the first node dies in the network.

7.3. Threshold Selection

In our scheme for energy efficiency, an energy threshold value that is, 16% has been set to limit access transmission of the unimportant and periodic network packets. In this section, we analyze different threshold values for the sake of selecting a threshold value with the improved results in NI, retransmissions, dead nodes, overall AEC, interest satisfaction ratio, and priority interests. We selected the values 4, 8, 12, 16, 20, 24, and 28 as threshold values to identify the best threshold value. The gap between the values is four so that we can analyze the results better. A smaller gap between threshold values would not permit significant analysis. We analyzed the threshold value with different threshold and a different number of consumer–producer pairs. The main idea to increase the number of
consumer–producer pairs was to increase the traffic rate in the network and the flow of different types of traffic so that we could analyze the threshold results better.

Figure 14a shows an analysis of the total NI in the network. As the threshold value increases, the value of the total NI decreases gradually. Due to the threshold value, we limit the interests with low priorities. So, the low-priority packets less than the threshold value could not pass through the network. The interest rate increased with the increase in consumer–producer pairs. The lowest value of the threshold value was 16%, but as the threshold value increased from 20 due to an increase in retransmission interest, the total NI also increased.

**Figure 14.** Threshold selection against the number of total interests and average energy.

Figure 14b shows an analysis of the AEC in the network. The average consumption of the nodes’ energy was high with a low threshold because of the excessive transmission of low-priority packets. With a threshold value of 16%–20%, AEC was lowest. With increase in threshold value, an increase in AEC occurred because of packet retransmission. The NI packet was smaller, with a threshold value of 16%, which resulted in fewer retransmissions of the packet. These two factors resulted in fewer dead nodes, and the energy consumption was also limited to the threshold selection. After evaluating and analyzing different threshold values with different evaluation parameters, threshold value of 16% gives us better results in almost all the evaluation comparisons. Therefore, we selected the threshold value of 16 for our energy efficiency scheme.

### 7.4. Simulation Results

Figure 15 shows the performance analysis of CRT, NI, AEC, and NLT against one to five consumer–producer pairs. The value of the x-axis is consumer–producer pairs; y-axis values are CRT, NI, AEC, and NLT. This performance graph illustrates that the performance of EPF was better than the performance of GIF and flooding. The value of CRT also increased with more consumer–producer pairs because of the access in traffic load and network congestion. The CRT of EPF was lower than GIF. The NI and AEC in EPF were less than both comparative schemes because of efficient use of the SDN controller and energy-efficient forwarding. Low energy consumption led to lower NLT in EPF against GIF and flooding. The energy efficient mechanism based on threshold that prevents the node energy for high priority packets results in better performance as compare to flooding and GIF. Increased Consumer/Producer pairs results in more traffic in the network. Any potential forwarder with energy below the threshold value will drop the LP packets. Efficient energy consumption will lead to the better NLT. As, only HP packets can travel through the nodes with energy less than threshold value, so so the NLT will also increase. The proposed forwarding mechanism uses controller and energy efficient mechanism, that results in better CRT as compared to other schemes. Controller involvement
makes sure the fast and in-time packet delivery. While the limitation on low priority packets also results in better CRT and NI.

![Graphs showing performance analysis](image)

**Figure 15.** Performance analysis against number of consumer/producer pairs.

Figure 16 shows the evaluation of the performance according to the number of high-priority interests (HPI), number of retransmissions (RTx), AEC, and NLT against the number of nodes. The value of the x-axis is the number of nodes; y-axis values are HPI, RTx, AEC, and NLT. This performance graph illustrates that the performance of the EPF is far better than the performance of GIF and flooding. When the number of nodes increased, the value of the HPI increased in EPF because of the efficient priority forwarding mechanism. The priority interest satisfaction rate in EPF is far better than others, because of energy aware priority forwarding mechanism. Network availability is more reliable in EPF. In our proposed scheme, decreased NI and increased satisfaction ratio results in lower RTx in EPF than GIF because of the limitation on low priority packets and exact location mapping of the content through controller. The AEC in EPF is lower than both comparative schemes because of the efficient use of the SDN controller and energy-efficient forwarding. Low energy consumption led to lower NLT in EPF against GIF and flooding.
8. Conclusions

In this paper, authors proposed an Energy-Efficient Priority-Based forwarding in SDN enabled NDN-IoT. EPF uses the SDN controller to manage the flow control of communication and take record of node’s information that minimize the broadcast storm and helps in fast forwarding. Energy threshold mechanism is used to control the energy consumption. An Energy threshold 16% has been devised to control the low packet transmission. Only high priority packets can go through network. Defer window mechanism reserves the priority slots to give priority to the highly important packets. We compared our scheme with flooding and GIF that is an efficient forwarding mechanism in NDN-IOT. Our scheme outclass both comparative schemes and gives the best performance in terms of No. of Interest, Content Retrieval Time, Network Life Time, No. of High Priority Interests, Retransmissions and Average Energy Consumption because of the use of centralized controller and efficient forwarding strategy. In future, we will take mobility of the nodes into account to enhance the current forwarding scheme.

Author Contributions: Conceptualization, A.T. and R.A.R.; methodology, A.T. and R.A.R.; software, A.T.; validation, A.T., R.A.R. and B.-S.K.; formal analysis, A.T., R.A.R. and B.-S.K.; investigation, A.T., R.A.R. and B.-S.K.; resources, A.T; data curation, A.T. and R.A.R.; writing original-draft preparation, A.T.; writing-review and editing, A.T., R.A.R. and B.-S.K.; visualization, A.T., R.A.R. and B.-S.K.; supervision, R.A.R.; project administration, A.T. and R.A.R.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government. (No. 2018R1A2B6002399).

Conflicts of Interest: The authors have declared no conflict of interest regarding this publication.
References

1. Meulen, R.V.D. Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017 up 31 Percent from 2016. Available online: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016 (accessed on 7 March 2019).

2. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015, 17, 23472376. [CrossRef]

3. Khan, R.; Khan, S.U.; Zaheer, R.; Khan, S. Future Internet: The Internet of Things architecture, possible applications and key challenges. In Proceedings of the 10th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2012; pp. 257–260.

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

5. NDN Project Architecture. Available online: http://www.nameddata.net/project/faq/ (accessed on 10 October 2020).

6. Tan, L.; Wang, N. Future Internet: The Internet of Things. In Proceedings of the 3rd ICACTE, Chengdu, China, 20–22 August 2010; pp. V5-376–V5-380.

7. Nisar, K.; Welch, I.; Hassan, R.; Sodhro, A.H.; Pirbhulal, S. A Survey on the Architecture, Application, and Security of Software Defined Networking. Internet Things 2020, 100289. [CrossRef]

8. Nunes, B.A.; Mendonca, M.; Nguyen, X.N.; Obrazcka, K.; Turletti, T. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Commun. Surv. Tutor. 2014, 16, 16171634. [CrossRef]

9. Tariq, A.; Rehman, R.A.; Kim, B.S. Energy Efficient Priority Aware Forwarding in SDN Enabled Named Data Internet of Things. In Proceedings of the 2020 International Conference on Electronics, Information, and Communication (ICEIC), Barcelona, Spain, 19–22 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4.

10. Caraguay, A.L.V.; Peral, A.B.; Lópezo, L.I.B.; Villalba, L.J.G. SDN: Evolution and opportunities in the development of IoT applications. Int. J. Distrib. Sens. Netw. 2014, 10, 735142. [CrossRef]

11. Qin, Z.; Denker, G.; Giannelli, C.; Bellavista, P.; Venkatasubramanian, N. A software defined networking architecture for the Internet-of-Things. In Proceedings of the IEEE network operations and management symposium (NOMS), Krakow, Poland, 5–9 May 2014; pp. 19.

12. Aubry, E.; Silverston, T.; Chrisment, I. SRSC: SDN-based routing scheme for CCN. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.

13. Torres, J.; Alvarenga, I.; Boutaba, R. et al. Evaluating CRoS-NDN: a comparative performance analysis of a controller-based routing scheme for named-data networking. J. Internet Serv. Appl. 2019, 10, 20. [CrossRef]

14. Syrivelis, D.; Parisis, G.; Trossen, D.; Flegkas, P.; Sourlas, V.; Korakis, T.; Tassiulas, L. Pursuing a software defined information-centric network. In Proceedings of the 2012 European Workshop on Software Defined Networking (EWSDN), Darmstadt Germany, 25–26 October 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 103–108.

15. Ansari, F.; Rehman, R.A.; Kim, B.S. CSF: Controller based Selective Forwarding in Software Defined Named Data based MANETs. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.

16. Arsalan, A.; Rehman, R.A. Distance-based scheme for broadcast storm mitigation in named software defined vehicular networks (NSDVN). In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.

17. Alowish, M.; Shiraishi, Y.; Takano, Y.; Mohri, M.; Morii, M. A novel software-defined networking controlled vehicular named-data networking for trustworthy emergency data dissemination and content retrieval assisted by evolved interest packet. Int. J. Distrib. Sens. Netw. 2020, 16, 15501472090280. [CrossRef]

18. Banerjee, A.; Hussain, D.M.A. SD-EAR: Energy Aware Routing in Software Defined Wireless Sensor Networks. Appl. Sci. 2018, 8, 1013. [CrossRef]

19. Aboud, A.; Touati, H.; Hnich, B. Efficient forwarding strategy in a NDN-based internet of things. Clust. Comput. 2018, 1–14. [CrossRef]
20. Gao, S.; Zhang, H.; Zhang, B. Energy efficient interest forwarding in NDN-based wireless sensor networks. *Mob. Inf. Syst.* **2016**, *2016*, 3127029. [CrossRef]
21. Tariq, A.; Rehman, R.A.; Kim, B.S. Forwarding Strategies in NDN based Wireless Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *2019*, 22, 68–95. [CrossRef]
22. Amadeo, M.; Campolo, C.; Iera, A.; Molinaro, A. Named data networking for IoT: An architectural perspective. In Proceedings of the 2014 European Conference on Networks and Communications (EuCNC), Bologna, Italy, 23–26 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–5.
23. Meddeb, M.; Dhraief, A.; Belghith, A.; Monteil, T.; Drira, K.; Gannouni, S. AFIRM: Adaptive forwarding based link recovery for mobility support in NDN/IoT networks. *Future Gener. Comput. Syst.* **2018**, *87*, 351–363. [CrossRef]
24. Zhang, Q.Y.; Wang, X.W.; Huang, M.; Li, K.Q.; Das, S.K. Software defined networking meets information centric networking: A survey. *IEEE Access* **2018**, *6*, 39547–39563. [CrossRef]
25. Assefa, B.G.; Ozkasap, O. State-of-the-art energy efficiency approaches in software defined networking. *Proc. ICN* **2015**, 268.
26. Ioannou, A.; Weber, S. A survey of caching policies and forwarding mechanisms in information-centric networking. *IEEE Commun. Surv. Tutor.* **2016**, *2016*, 18, 2847–2886. [CrossRef]
27. Quevedo, J.; Antunes, M.; Corujo, D.; Gomes, D.; Aguiar, R.L. On the application of contextual iot service discovery in information centric networks. *Comput. Commun.* **2016**, *89*, 117–127. [CrossRef]
28. Stankovic, J.A. Research directions for the Internet of Things. *IEEE Internet Things J.* **2014**, *1*, 3–9. [CrossRef]
29. Muralidharan, S.; Roy, A.; Saxena, N. MDP-IoT: MDP based interest forwarding for heterogeneous traffic in IoT-NDN environment. *Future Gener. Comput. Syst.* **2018**, *79*, 892–908. [CrossRef]
30. Benedetti, P.; Ventrella, A.V.; Piro, G.; Grieco, L.A. An SDN-aided Information Centric Networking Approach to Publish-Subscribe with Mobile Consumers. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 130–137.
31. Shah, S.H.; Yaqoob, I. A survey: Internet of Things (IOT) technologies, applications and challenges. In Proceedings of the 2016 IEEE Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 21–24 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 381–385.
32. Zhan, F.B. Three fastest shortest path algorithms on real road networks: Data structures and procedures. *J. Geogr. Inf. Decis. Anal.* **1997**, *1*, 69–82.

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).