(IJIEST) 2020, Vol. No. 6, Jan-Dec **e-ISSN: 2454-9584; p-ISSN: 2454-8111**

FEASIBILITY ANALYSIS OF FOUNDING SMART CITY BY LEVERAGING THE NAMED DATA NETWORKING OF THINGS (NDNOT) TECHNOLOGY

Pranshul Pahwa

ABSTRACT

Objectives: The paper shows the model for Smart City based on Named Data Network (NDN). In country like India an effective model based on NDN like ours will be very useful for managing traffics. Methods/Analysis: NDN is an upcoming technology, which is believed that it will replace Internet. The Internet Protocol version 6 (IPv6) will exhaust with the growing rate of connected devices. The paper proposes a complete model based on Named Data Networking of Things (NDNoT) expanding the idea of Internet of Things (IoT) for not only managing the traffic but also for helping accidental cases. The model uses sensors to convey information to the cloud server from where it can be distributed to concerned authorities and person using NDN as the infrastructure. The paper covers the whole algorithm for designing the model. Findings: Issues like Security, Congestion, Routing, Reliability and Speed are improved to an extend in NDNoT as compared to the models based on IoT due to its principles of allocations depending on Hierarchal names with signatures. Improvement: The model can be practically implemented using NDN.

1. INTRODUCTION

In today’s world, there are innumerable ways for communication, with certain protocols. There is a host-to-host communication. We try to get connected to certain servers on Internet by just typing the URL or its equivalent IP address in our web browser. Whenever the connection gets established, transfer of information starts from the server to our web browser and then the web browser converts it into displayable and user understandable format. In short, if everyone wants that same set of data in their browser then they must communicate with that host which is having that data.

Server gets busy in doing same process number of times resulting in poor transmission. Every server has a limited capability to handle number of threads. If number of requests coming on that server is greater than the handling capacity the server may even crash. This is what exactly happened with “Flipkart” on its so-called “Big-Billion Day” some years ago.

Now we have a replacement for this and that was named as “Cloud”. Undoubtedly cloud computing sorted out such kind of issues by providing scalability of resources. But what’s’ next? How to improve the technology further? All the domains of Computer science have been improved, but we left one portion and that is “Networking”. Networking is the most important aspect for the solutions of such issues. IPv4 is the protocol, which is most common for Internet connections. It uses a 32-bit addressing scheme resulting in total of 232 addresses1. As Internet is growing with fast pace it is expected that soon the unused addresses would be exhausted. A new addressing system IPv6 is used to have more addresses, which allows 2128 addresses1.

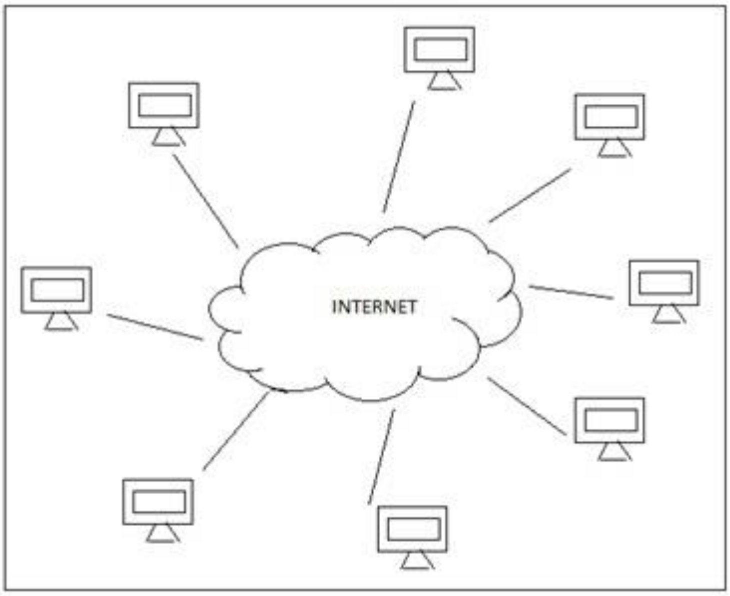

Figure 1. Network

But this kind of transformation in technology still compels us for host-to-host communication. Again we have to face the traffics over a network. Named Data Network is a replacement for the above issues2. NDN addresses the issues by replacing addresses by Names based on the content2. The proposed model uses NDN to solve the issues of traffic, accidents, driving rules violation etc.

A lot of work has been done to resolve the congestion in network by reducing redundant transfers for which many new Internet and Internet-scale architectures have been proposed3. The concept of name-centric instead of address-centric is the new paradigm. In addition to recently proposed data-centric network architectures, the HTTP protocol and its globally deployed infrastructure have rightly been identified as a data-centric system4,5.

NDN proposes touse the concept of name centric system. The search and request would be on based of name not address and request of same name would be checked and resolved by using some tables called Content Table, PIT(Pending Interest Table) etc. An efficient PIT is designed, named DiPIT in which Bloom filters are used as the core structure which splits the PIT into several sub-tables and implements them on each face of NDN node6.

Ad hoc Vehicular Network, Vehicular Information Network, Internet of Things is being widely used now. Intelligent Vehicles communicating with different devices are the future and the concern is the network traffic due to so many communicating devices. Raptor codes7are applied at the content sources and have very low computational cost. They reduce content retrieval times by 83% compared to original NDN7,8.

NDN work on the principle of longest prefix matching for searching and caching the results, which increases the power consumptions. A power consumption model of multi-core software NDN router has been developed9. According to it caching can reduce power consumption of an NDN network ifthe routers consume power in proportion to their loads and proportion of caching and forwarding is also similar.

2. NETWORKING

One can think the present TCP/IP as a four-layer model, with each layer responsible for performing some particular task. Protocols are there for each layer, which decides the services of the layer.

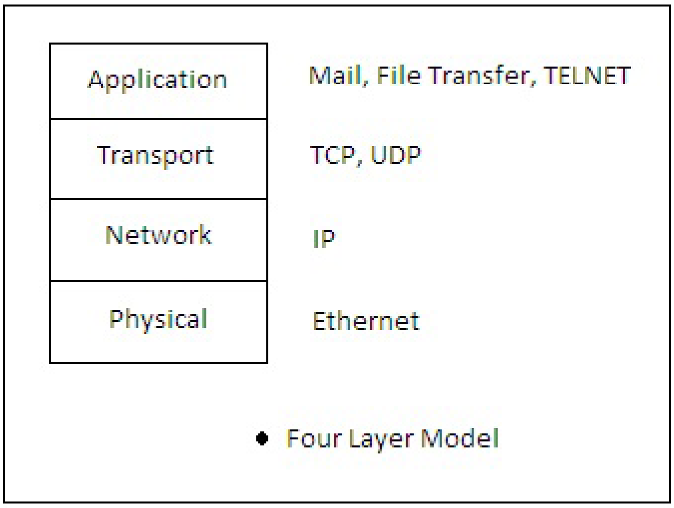

Figure 2. TCP Model

Application: • This is the layers which is responsible for running the user application. Several protocols on it are FTP, SMTP, POP.

Transport: • This layer is responsible for transmission of data from one node to another. There are two basic protocols UDP and TCP.

Network: • This layer is responsible for getting data packets from one network to another. IT takes care of routing, congestion. Main protocol is IP.

Physical: • It is the lowermost layer, which deals with the physical media of communication.

3. NAMED-DATA-NETWORKING

Changing technology of networking from IPv4 to IPv6 only increases the number of hosts that can communicate over the Internet. A particular IP address gets routed to another IP address (for e.g. 173.194.116.223 or google.co.in) by an algorithm known a "Longest Prefix Match Algorithm"9. After a long research over it, it was found that requesting could get faster if naming conventions are used instead of an IP address to get data.

Figure 3 depicts the host-to-host communication. The web browser of client via Interface-1 generates a request for a page. After applying the longest prefix match, the request gets routed to the correct interface (in this case it's Interface-4). After sending the desired data, the connection gets closed. But is it necessary to communicate to that farthest system always? Sometimes or we can say most of the time, the data that one system is requesting for is present in the buffer of a nearby interface. So, instead of establishing a connection to the farthest interface for a data, one can get that data from the sideway systems if that system is having that requested data.

Figure 4 depicts that the Interface-2 already has that requested data. So, a communication path is developed between the Interface-1 and Interface-2 instead of Interface-1 and Interface-4. The major problem with host-to-host communication is it is dependent on addresses.

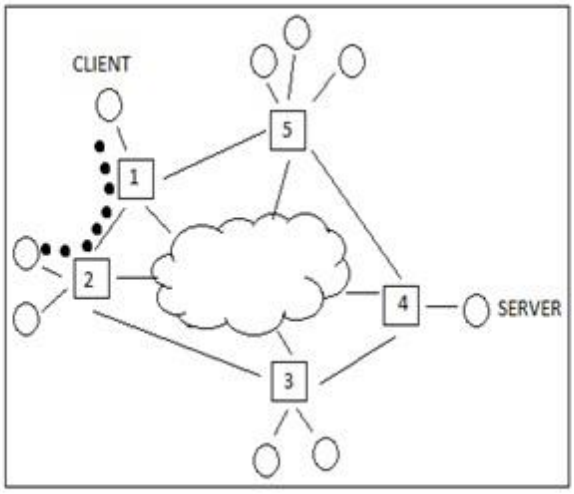

Figure 3. Networking based on IP address

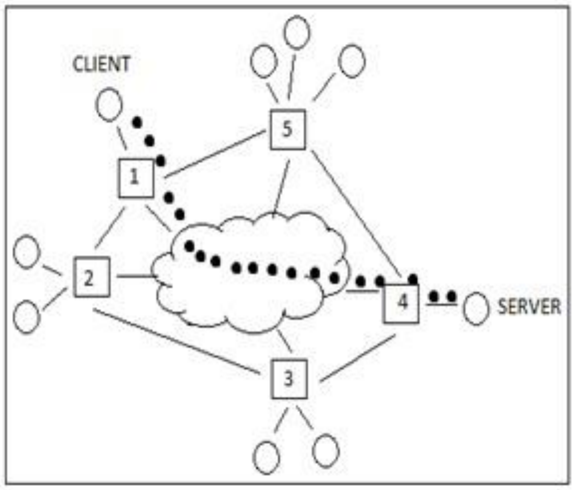

Figure 4. Named-Data-Networking

Named-Data-Networking came up with an ideology that communication should be based on named data request instead of the addresses (telephony model) i.e. NDN is a simpler general purpose protocol which searches the data depending on the name rather than addresses. People might be thinking that, is this network orggoing to replace the present Internet? Well, it is partially true actually NDN works on the top of Internet protocols not actually replacing them. So, the Internet is supposed to work as it is working right now with a much less traffic.

4. NDN OVERVIEW

The main focus/goal of NDN is to make content available to a requester of data irrespective of the location of that content hosting server. Basically two types of packets are related with NDN: Interest Packets and Data Packets10. These two packets together remove the notion of addresses.

Interest packet possesses name, order preference &nonce (present location) whereas Data packet contains name, meta-data (content-type), content and signatures (for security authentication). Requester sends Interest packet from its gateway. Whichever node comes along the path of the interest packet; three questions arises:

1. Is data present with you?

2. Is a request already pending?

3. Which next hop may lead to the source?

Upon these three questions NDN interest forwarding takes place. Each NDN router maintains three data structures within it11.

Content Store:• The CS stores the packets in the cache (as same as Cache Memory in Memory Hierarchy), to fulfil future requests for the content that has already been fetched. For name lookups, it uses exact matching.

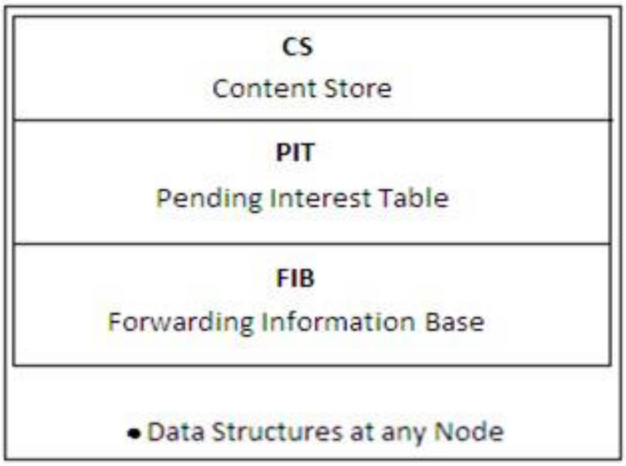

Figure 5. NDN packet structure

Table:• It keeps the interest requests that are created by any requester until or unless they are satisfied.

Forwarding Information Base:• It contains the information about the next hops that can lead the interest packet to the data source.

If the desired packet is available at CS request is satisfied and data is send to the requester else using the FIB interest is sent further till it is matched and data is sent back whenever the interest is matched. When the data is sent to the receiver, the publisher binds name to the data, which is verified by the receiver. The Data packet back traces the path of the Interest Packet in reverse order to reach the data requester. Retracing an address is not a big issue but the primary focus is on, how the forwarding mechanism will take place.

5. FORWARDING OF INTEREST PACKET IN NDN

NDN uses name based routing and forwarding, it resolves the problem faced in address based IP routing like addresses exhaustion, NAT traversal, mobility of devices and scalable address management. Each node (router or end node) in NDN can forward the interest packet to more than one interface (multipath routing) listed in FIB, which increases the efficiency. As the same path is taken in reverse by Data problem of looping and duplicity is also addressed. An advanced "Controller-based Routing Scheme (CRoS)" is also discussed in12. CRoS introduces special controllers, which have two main functions: 1. acquire topology and calculate routes, and 2. store named data locations.

Consider this scenario.

Device-1 is emitting an Interest Packet asking for a • piece of data at domain "gkv.ac.in/? page_id=2664" hop.

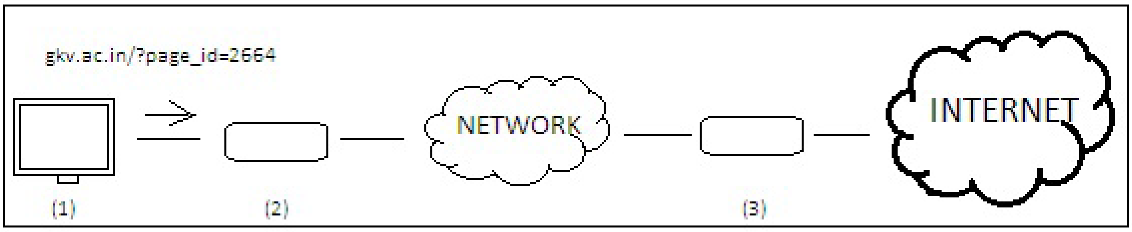

Figure 6. Forwarding stage 1

First Forwarding element (Device-2) receives the • Interest packet. It first searches that requested data in its buffer. If it is not available, then it looks whether a pending request is in flight or not. At last if all situations fail then the interest is forwarded to the next hop and added to the Pending Interest Table (PIT). In this case, Device-2 doesn't have the required data. So, it forwarded the Interest Packet to next hop.

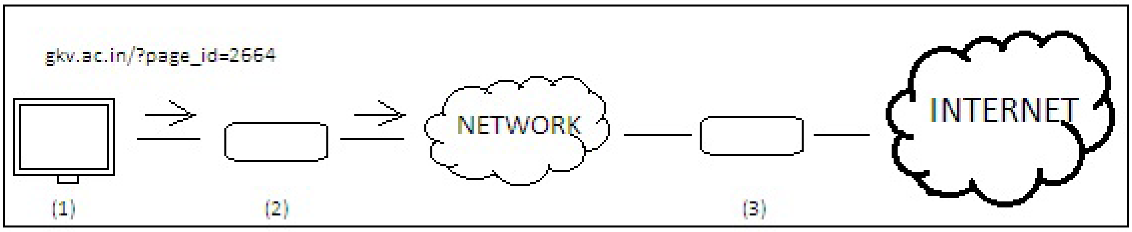

Figure 7. Forwarding stage 2

•Same thing happened with Device-3 and the controller of Device-3 forwarded the Interest packet to the next hop and this request gets added to the PIT.

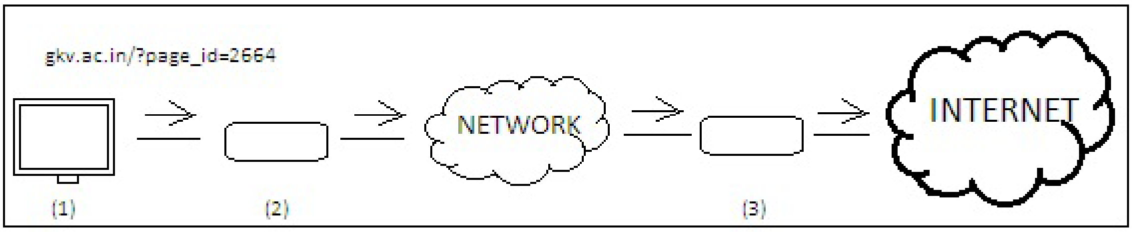

Figure 8. Forwarding stage 3

• Now suppose a situation where a Device-4 is expecting the same data from domain "gkv.ac.in/?page_id=2664". Device-4 will generate the Interest Packet and the controller of Device-4 will route it as per its Forwarding table. Device-3 will receive the Interest Packet and look in to its buffer. This time, Device-3 is having the requested data. So, it will return the Data Packet to Device-4. Some people might thing that what's the benefit of such kind of forwarding or if this forwarding is beneficial then up to which extent?

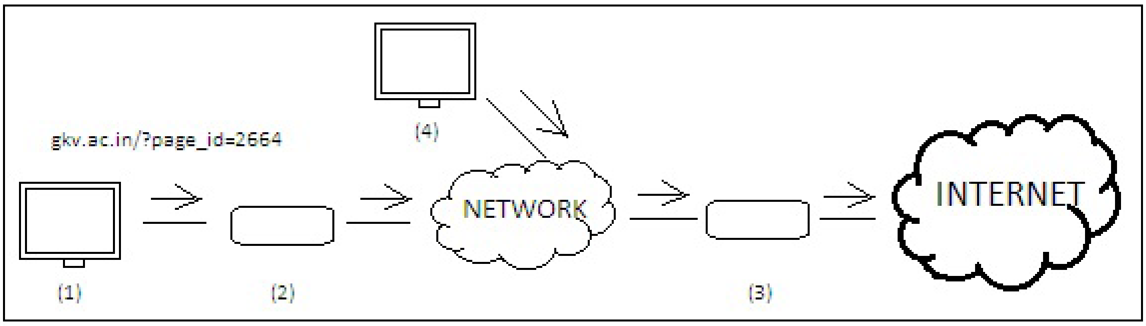

Figure 9. Forwarding stage 4

In simple terms:

Minimum time to fetch data = time taken to communicate to the next node

Maximum time to fetch data = time taken in present (host-to-host) communication mode 6. Live Streaming of Cricket Match Using NDN

Let us consider a case of Cricket match, live streaming using NDN. Present generation is a big fan of Cricket. Everyone loves to watch cricket matches. People are so addicted to watching cricket that they don't even miss it during their office hours. No one can stick to television all the time for live telecast. So what people prefer is they watch the live streaming via Internet.

Video of cricket match is recorded and uploaded to the server with a very high-speed connection. Then the end-user obtains the frames of the match from the server on their devices (phones, tablets, laptops, etc).

But what happens maximum times are we suffer from buffering. Even a delay of 6 seconds to 5 minutes can occur. All the end users try to communicate to the main server to get the required frames of the video. Then it becomes a very typical job to schedule all the threads. When a thread gets the CPU, it sends the required frame of video from the server to the client. Till that time, client has to wait. A huge traffic it will create in the network. This is the thing what exactly happens with all of us.

Now think it in NDN's scenario. Some nodes will get the required frames directly from the server. When other nodes try to fetch the same frames it can obtain it from other nodes where their controllers will route their Interest Packet. All nodes need not to communicate to the host. Definitely it will result in less time to fetch the content that is required. In other words, the node that is providing the content to the requester serves the purpose of the host and act as if it is the host to the requester.

The above example shows how NDN can be used for various applications involving high network traffic.

7. NDNOT

NDNoT stands for 'Named-Data-Networking of Things'. The scenario before a decade was, people used to say that IoT (Internet of things) is the future, it is the one on which our future relies. But the fact is, if the present host-to-host communication (or we say the Telephony Model) is changed then the protocols used for transmitting or receiving signals by the different sensors in IoT must be changed. The benefits of using NDN over the telephony model are clearly visible. So, if NDN is the future of Internet then the NDNoT is the upgraded version of IoT and then IoT will get replaced. Many papers13-17 discusses use of IoT at various fields but soon NDNoT will be

the ruler.

IoT exchanges data over IP-centric Internet which has vulnerabilities related to security, privacy, and mobility. NDN has built-in support for the end users as well as the service providers. NDN provides data security instead of channel security, which is presently being provided by the Internet.

There are several sectors where this NDNoT can be implemented. Actually, in almost all sectors it can be implemented. But for initial level, our prime concern is only for those sectors, which require sudden attention to the consumers or end users or the service providers depending upon the situations. For example, m-health services or traffic controlling systems.

8. PROPOSED MODEL FOR "TRAFFIC-CONTROL APPLICATION"

One of the most important sectors, which need to be upgraded, is the Traffic-Controlling process. Application of such projects hardly affects the countries with less population, but this project is designed mainly considering countries with large (or at least acceptable) population such as India, China, etc. Controlling traffic in some scenario can even get worse.

Imagine a smart city where there are sensors on roads for sensing the traffic. Sophisticated chips and sensors are embedded in the physical things that surround us each transmitting valuable data. Data that lets us better understand how these things work and work together. But how exactly do all these devices share such a large data and we put that data into work?

A component called the Diagnostic Bus gathers all the data from all the sensors of a vehicle then passes it to a gateway into a car. The gateway integrates and sorts the data from the sensors. This way, only the relevant data will be transmitted to the cloud platform.

But before sending the organized data, gateway and platform must register with each other and confirm a secure communication. After registering, all devices can share the data with the platform.

Cloud/server has records each and every device connected to it – device id, current status of the device depending up on which it sends the messages on the cell phones so that people can get rid from such traffic.

After receiving signals from various sensors of traffic cameras and the gateway of cars, analysis takes place. As per the location of the devices that are connected to that cloud server near the point of accident get a message on their smartphones "take diversion from 'XYZ' to avoid traffic".

The main backbone of the project is that for all communications data will be converted into packet forms of NDN and the infrastructure would be modelled as per NDN requirements. Instead of searching and contacting the cloud server for each communication, NDN principle will be used. This will help in reducing the time delay and will make the system robust and responsive.

9. STEPS FOR IMPLEMENTATION

- Registering all the vehicles and cameras to the cloud server.
- Generate a unique Device-ID and Group-ID (Group-ID will be generated depending upon the location of a vehicle at a particular instance of time and it will keep on changing as per the geographic location).
- When an accident takes place, the sensors of the vehicle after collecting and sorting the data transmits to the cloud server.
- Cloud server will signal the cameras to check whether the accident has actually taken place or not.
- Sensors of camera will impart signal of the status of accident.
- After filtering all the data and passing it from various constraints required actions will take place. For example, informing nearby police stations and the public nearby the point of accident.

At some moment of time an accident takes place. This will definitely create chaos on that road. Traffic will increase due to this accident. Devices start securely communicating to a common platform. A cloud platform integrates data from many devices and applies analytics to derive most valuable data. The sensors will impart signals regarding the status to the cloud server.

10. PROPOSED ALGORITHM

Step 1: Register all cameras

Camera camera = new Camera ();

Camera.camera_id = Server.getId (camera);

Step 2: Register all vehicles

Vehicle vehicle = new Vehicle ();

Vehicle.vehicle_id = Server.getId (vehicle);

Ve hicle.group_id = Vehicle.generateGroupId (vehicle_id, vehicle.getLocation ());

Step 3: Add vehicle in location list

List [vehicle.group_id]. add (vehicle);

Step 4: If location changes then

List [vehicle.group_id]. delete(vehicle.vehicle_id);

Ve hicle.group_id = Vehicle.generateGroupId (vehicle_id, vehicle.getLocation ());

List [vehicle.group_id]. add (vehicle);

Step 5: For imparting a particular message to a number of vehicles of particular location

Server.sendMessage (List [location_number], message);

11. CONCLUSIONS AND FUTURE WORK

From all aspects it is clear that NDN has much better capabilities that the host-to-host communication and hence it is the future of the Internet. Working over such technology will definitely worth it. Rate of progress in this field signifies that, within half of a decade, life will get smarter by the use of Internet of things in collaboration with NDN that is, NDNoT. All sorts of communications will be overruled by NDN.

This paper work is just a proposal of, what kind applications are possible from NDN. We provided one application of NDN as “Traffic-Control Application”. We have designed the entire prototype for this application. But still, the actual implementation is yet to be done. Future works regarding this application is to implement it using “Sensors”. Sensors will be used to sense the status of a vehicle and to transmit information to the cloud platform. An efficient NDN based cloud platform can be used so as to derive useful information from the recorded data and perform various actions depending upon the situation.