Scientific Research Publishing

Advances in Internet of Things, 2017, 7, 11-24
http://www.scirp.org/journal/ait
ISSN Online: 2161-6825
ISSN Print: 2161-6817

Proposed Caching Scheme for Optimizing Trade-off between Freshness and Energy Consumption in Name Data Networking Based IoT

Rahul Shrimali[1], Hemal Shah[1], Riya Chauhan[2]

[1]Ganpat University, Mehsana, Gujarat, India
[2]PG School, GTU, Ahmedabad, Gujarat, India
Email: rahulshrimali2526@gmail.com, hemal.shah@ganpatuniversity.ac.in, riyachauhan@gmail.com

How to cite this paper: Shrimali, R., Shah, H. and Chauhan, R. (2017) Proposed Caching Scheme for Optimizing Trade-off between Freshness and Energy Consumption in Name Data Networking Based IoT. *Advances in Internet of Things*, **7**, 11-24.
https://doi.org/10.4236/ait.2017.72002

Received: March 10, 2017
Accepted: April 17, 2017
Published: April 20, 2017

Abstract

Over the last few years, the Internet of Things (IoT) has become an omnipresent term. The IoT expands the existing common concepts, anytime and anyplace to the connectivity for anything. The proliferation in IoT offers opportunities but may also bear risks. A hitherto neglected aspect is the possible increase in power consumption as smart devices in IoT applications are expected to be reachable by other devices at all times. This implies that the device is consuming electrical energy even when it is not in use for its primary function. Many researchers' communities have started addressing storage ability like cache memory of smart devices using the concept called—Named Data Networking (NDN) to achieve better energy efficient communication model. In NDN, memory or buffer overflow is the common challenge especially when internal memory of node exceeds its limit and data with highest degree of freshness may not be accommodated and entire scenarios behaves like a traditional network. In such case, Data Caching is not performed by intermediate nodes to guarantee highest degree of freshness. On the periodical updates sent from data producers, it is exceedingly demanded that data consumers must get up to date information at cost of lease energy. Consequently, there is challenge in maintaining tradeoff between freshness energy consumption during Publisher-Subscriber interaction. In our work, we proposed the architecture to overcome cache strategy issue by Smart Caching Algorithm for improvement in memory management and data freshness. The smart caching strategy updates the data at precise interval by keeping garbage data into consideration. It is also observed from experiment that data redundancy can be easily obtained by ignoring/dropping data packets for the information which is not of interest by other participating nodes in network, ultimately leading to optimizing tradeoff between freshness and energy required.

Keywords

Internet of Things (IoT), Named Data Networking, Smart Caching Table, Pending Interest, Forwarding Information Base, Content Store, Content Centric Networking, Information Centric Networking, Data & Interest Packets, SCTSmart Caching

1. Introduction

With the fast growing technology the communication is no more confined between devices and humans. We have reached to an era where the IoT devices are designed and expected to communicate to each other using Machine-to Machine paradigm. It is said that IoT is collection of 6A that are Any Time, Any Where, Any One, Any Thing, Any Service, Any Path. IoT devices are controlled devices having limited resources comprising of a memory, CPU and battery. The embedded devices used in IoT are usually expected to work in low power wireless frequencies. These devices is IP enabled, can work using IPv6 over Low Power Wireless Personal Area Network (LoWPAN *i.e.* IEEE 802.4.15) and they make up network of Internet of Things. The optimal energy usage plays vital role in deciding the throughput of network, its efficiency and life time, if the devices are self-powered. However, today's Internet's timepiece architecture focuses mainly on how smart devices get connected worldwide with each other using set of network layer functionalities. Providing freedom to upper and lower layer protocols and technologies to innovate and grow independently turns into the hotheaded growth without considering more constraints [1].

The NDN architecture withholds the same chronometer form, and transmogrifies the compatibility of data as an alternative of its place. NDN makes the original Internet design elegant and powerful by changing the network communication rules morphologically and can change the entire delivery process of IP packets where retrieval of data is possible by indentifying Contents *i.e.* name instead of destination address exclusively [2] as shown in **Figure 1**.

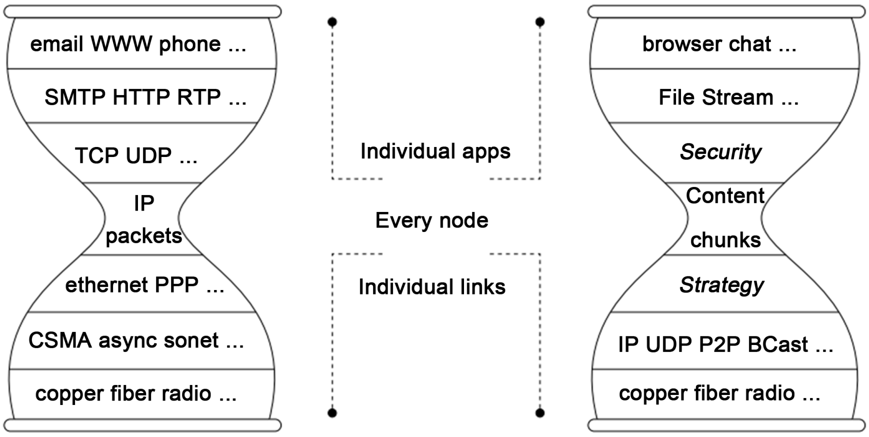


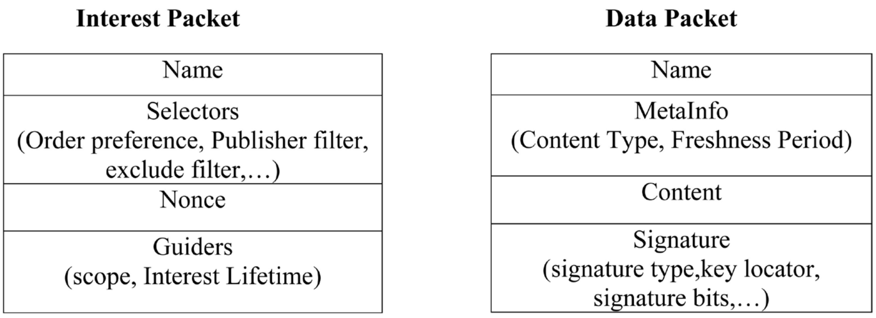
Figure 1. Internet and NDN hourglass architecture [1].

NDN architecture is a suitable architecture for IoT technology as three layers like thing Layer, NDN layer and application layer fitting best for Machine to Machine Communication. NDN layer is mainly consisting of Management control plane and Data plane. Management control plane includes security of transmitted packets, routing of interest and data packets, configuration of node, etc. Data plane includes caching, security of cached data, naming conventional, data caching strategy etc. According to the NDN communication model, data subscriber or consumer is the first party who initiates the entire process for data exchange through mainly two types of packets: Interest and Data as shown in **Figure 2**. Single data packet comprises both types of packets: interests cart a name and data carry a short of information for which consumer is interested. For receiving desirable data, first consumer sticks the name of desired data into an Interest packet and sends it across the network. Now the job of routers in NDN is to take routing decisions or forwarding decision according to the interest received from either neighboring router or host to data consumers. Suppose, interest packet "pi" reaches to Node "X" subsequently which contains the information seek by consumer node 'Y' then node "X" will combine this information available with it using the name and the content in resultant Data packet "Pd" jointly with a signature by the data providers as per **Figure 2**. Data packet "Pd" obeys has to be transmitted exactly reverse path of the Pi packets to reach back to the data subscriber who has demanded data.

The NDN router maintains three tables: Content Store (CS) maintains cached data received from neighbor node. Pending Interest Table (PIT) is responsible to maintain list of interests not satisfied from publishers. PIT table records incoming and outgoing interest & their name also. While forwarding to the next node, PIT takes due care for list of upstream consumers if it receives several Pi with the same name from downstream consumers, it passes to only the first in upstream toward the data publishers during the forwarding process at NDN node [3].

Forwarding Information Base (FIB) maintains list of records, forwarded to next neighbor by name-prefix. It does not rely much more on the specific knowledge about origin and destiny of NDN data packets. More obviously, it simply caches NDN data packets to assure and entertain future Interests. On the arrival of each interest packet let's say Pi, NDN router will follow the below procedure:

```
 For every $P_i$:
          If {matching $P_d$ exist in CS}
          then return $P_d$ on interface I from which interest originated;
      else if {matching $P_d$ exist in PIT} then
store $I_n$: Interest's incoming interface in the PIT
      else {matching $P_d$ exist in FIB}
          then: forward the $P_i$ toward the $D_{p:}$ Data Producer(s)
```



Interest Packet

Name

Selectors
(Order preference, Publisher filter, exclude filter,…)

Nonce

Guiders
(scope, Interest Lifetime)

Data Packet

Name

MetaInfo
(Content Type, Freshness Period)

Content

Signature
(signature type,key locator, signature bits,…)

Figure 2. Types of packets in NDN [1].

On the other side, on the arrival of Data packet (Pd), an NDN aware router will follow the below pseudo code:

```
For every $P_d$:
        If {$P_d$ matches in PIT}
        then initiate forwarding of $P_d$ on $LD_i$: list of all downstream interfaces;
        return: Delete($P_d$) from PIT and Cache $P_d$ in CS

    else repeat step 1.

    else{matching $P_d$ is exist in FIB}
        then: forward the $P_i$ toward the $D_p$: Data Producer(s)
```

To balance the flow of Pi and Pd, it is assumed that the flow of Pd must take exactly the quash path of Interests, and per each Pi there should be only one resultant Pd on each channel. Here we can note that no host address and interface addresses would be carried by Pi and Pd; Data producers are made responsible to receive Pi forwarded toward them according to the names concede in Pi, while Data Consumers are responsible to receive Pd based on the Pending Interest Table entries set by Interests at each station.

The Content Store can be compared with buffer memory in traditional IP datagram routers, but limitation of traditional IP datagram Router is that it's not storing any data packets once forwarded to the next node, while NDN routers have capability to reuse the references of cached data for future use. NDN guarantees optimal data delivery of both static and dynamic file. More specifically, against loss of dynamic file contents, caching is more beneficial to save efforts of multicast and retransmission policy. More unrelenting & gigantic in network storage—called Repository is available in NDN which facilitates application specific storage management, on other side CS provides opportunistic caching to get optimum performance [1].

Names are considered as a unique entity in NDN *i.e.* router may not have the idea of meaning of a name. It is also possible that entire naming scheme is independent of boundaries between routers. This separation allows varieties of application to choose varieties of naming scheme according to need of each applications [2]. Naming rules in NDN are assumed to be hierarchical structured e.g., an audio produced by Ganpat University PhD Centre may be namedv/gnuphdschool/wimc/rahul/track.mp3, where "/" describes different elements using textual illustrations same as URL structure. This URL type structure represents relationships between data elements in the application. Naming conventions indicate versions and segmentation and is application specific. The retrieval of data from producer side depends on the capability of consumer to build and generate the name for interest Pi. To do so, centralized name generation algorithm must be adapted by producer and consumer to have common derivative naming scheme. The rest of the paper is organized as below: Section II shows the related work and literature review done to address the issue, Section III includes our proposed approach of smart caching technique, implementation of proposed work, Section IV includes results and analysis. Section V shows future scope in our work done so far.

2. Related Work

Caching in NDN for the Wireless IoT scenario, Mohamed Ahmed Hai *et al.* [4] presented a probabilistic distributed approach called pCASTING for caching in which freshness of data can be optimized even in energy and space constrained environment. Through various scenario based test and simulations in ndnSIM, authors could achieve improvement in energy efficiency and data retrieval time with respect to traditional NDN caching mechanism assuming constrained environment. They developed pCASTING scheme is based on the probability distribution by considering parameters like node health level, cache possession status, and the lasting freshness data packets. To describe caching probability, authors in their work proposed Caching Utility Function (CUF) Fu [4] that takes the content attributes like Energy Level (EN), Cache Occupancy (OC) and freshness (FR) in account:

$$Fu = \sum_{i=1}^{Np} Wig\left(xi\right)$$

where, $N_p = 3^1$, $0 \le W_i \le 1$ and Power Function can be presented as:

$$g\left(x\right) = xi^n \text{ with } n \ge 1$$

The Utility Function is therefore expressed by the weighted sum of power functions:

$$Fu = w_1 * EN^n + w_2 * (1 - OC)^n + w_3 * FR^n$$

Every time on the arrival of Pd, host calculates the function value Fu to obtain the caching probability of that Pd. The technique is capable to reduce delay and energy consumed by every node at same time [4].

Jose Quevedo *et al.* [5] discussed in their work about caching in Information Centric Networking using novel cache replacement and decision policy. Authors presented their proposed work on Content Centric Network (CCN) scenario. The existing CCN technique was inspected in brief and new consumer driven freshness approach was narrated in their work. The proposed work can be extended by real time deployment of sensors in field by designing more complex algorithm with freshness agreement policy.

Xiaohu Chen *et al.* [3] presented caching strategy in Information-Centric Networking with the perspective of content delivery, called Least Unified Value (LUV)-Path. In proposed work, unique value is assigned to demonstrate the contents which are cached collectively and cooperatively by routers and entire delivery path. This value is known as LUV which then gets combined with distance of router from content provider to imitate its importance. Authors has implemented the concept of LUV Path under different topologies and proved that customer delay, network traffic and alleviating provider pressure is reduced in case of FIFO, LRU, LUV algorithms using LUV Path strategy. Extending their future work; authors introduced the scope of content popularity distribution which gives the judgment to host to propagate the contents in the network at the cost of less redundancy and retrieval delay.

Xiaoyan Hu *et al.* [6] in their work proposed novel routing approach that provides name based strategy and caching during routing. They focused on knowledge based cache systems in which self awareness in its control plane for the caching can be guaranteed using Cache-Aware Name Based Routing (CANR) method. In this method, router has to learn what kind of contents needs to be cached and what exactly router should advertise to rest of the network hence in this work, The Import Cache Filter Model (ICFM) is intended for the calculating and estimating the caching need of every router. In addition to that, to provide system scaling, Export Cache Routing (ECR) achieves advertisement of caching routing information in the network. Author has also suggested design of distributed cooperative caching method by various nodes as future scope.

Kyi Thar *et al.* [7] proposed consistent hashing based cooperative caching and forwarding method in content centric network. They concluded that Leave Copy Everywhere scheme for caching is not up to the mark in case of maintaining cache redundancy and storage capacity of the CCN network. The entire method is targeted for virtualization of router where multiple groups of routers forward Interest packets and collectively cache the data. Authors proposed the solution of scalability problem by modulo hashing but, content popularity prediction for cache decision and cache replacement is still open challenge found in their work.

Tao Liu *et al.* [8] introduced advanced content routing scheme based on neighbor's caching information obtained and routing decision are taken accordingly. This short cut policy combines caching algorithms and ACO algorithms. They also carried out the performance measurement of shortcut routing. In the context of power aware caching scheme, Dariusz Gasior and Maciej Drwal proposed an enhanced Power-Aware Multi-level (PAM) cache policy [9] which is capable enough to control energy consumption in Software Defined Network. This policy could optimize and conserve the power utilization of repository offering high performance and bandwidth [9]. Proper Identification and selection of dirty blocks are done from upper layer cache and the same are supposed to remove from storage device to provide less retrieval time and less power consumption in this scheme. In their research work, they propose a novel multilevel cache algorithm, which can sharply decrease the energy consumption via extending the period of standby mode of data disks.

In IoT networks & Applications, we often require sending and receiving a lot of data which may lead to more time and energy consumption as all latest or fresh information will be fetched from intermediate hosts or routers as an when required by IoT nodes attached to service interface of an IoT. Hence, Router play vital role in providing fresh information but may result into frequent contacts and REQUEST/REPLY messages as per PUBLISH/SUBSCRIBER architecture. To get improvement in freshness of data we can get latest updated data from intermediate node but other side we compromise energy requirements at every node which pose tradeoff between freshness of data and energy consumption in NDN. In our work, we propose efficient caching of data using NDN architecture where we make intermediate nodes or middle level nodes responsible to cache. Our main objectives are to achieve more freshness in update cached data in certain time interval and to analyze energy consumption while updating caching data freshness and maintaining cache memory to optimize the tradeoff as discussed earlier.

3. Proposed Approach and Implementation

In our work, we have introduced maintenance of additional information in specialized table called as SCTSmart Caching table as shown in Table 1 apart from CS, FIB, PIT tables as discussed earlier in introduction section. The SCT stores all entry for data caching and also manage the latest data interested data with their freshness. SCTSmart caching is designed for removing old and not interested data from CS table.

In SCTSmart Caching table, each caching entry maintains with faces and nonce. Faces are the total number of interest got by node. Nonce is the time of latest interest of cache. We update SCT for every Max Update Time. Max Update Time is time for updating smart caching table.

As per our objectives stated above, we shall set Max Update Time. We also declare Max Face Value which is total number of Interest. As per upcoming interest of cached data, we update SCT for maintaining freshness of cached data. As per how much fresh data is maintained we optimize ratio of freshness and energy consumption in Name Data Networking. Figure 3 shows flow chart for

Table 1. Proposed SCTSmart caching table.

Name	Face	Time

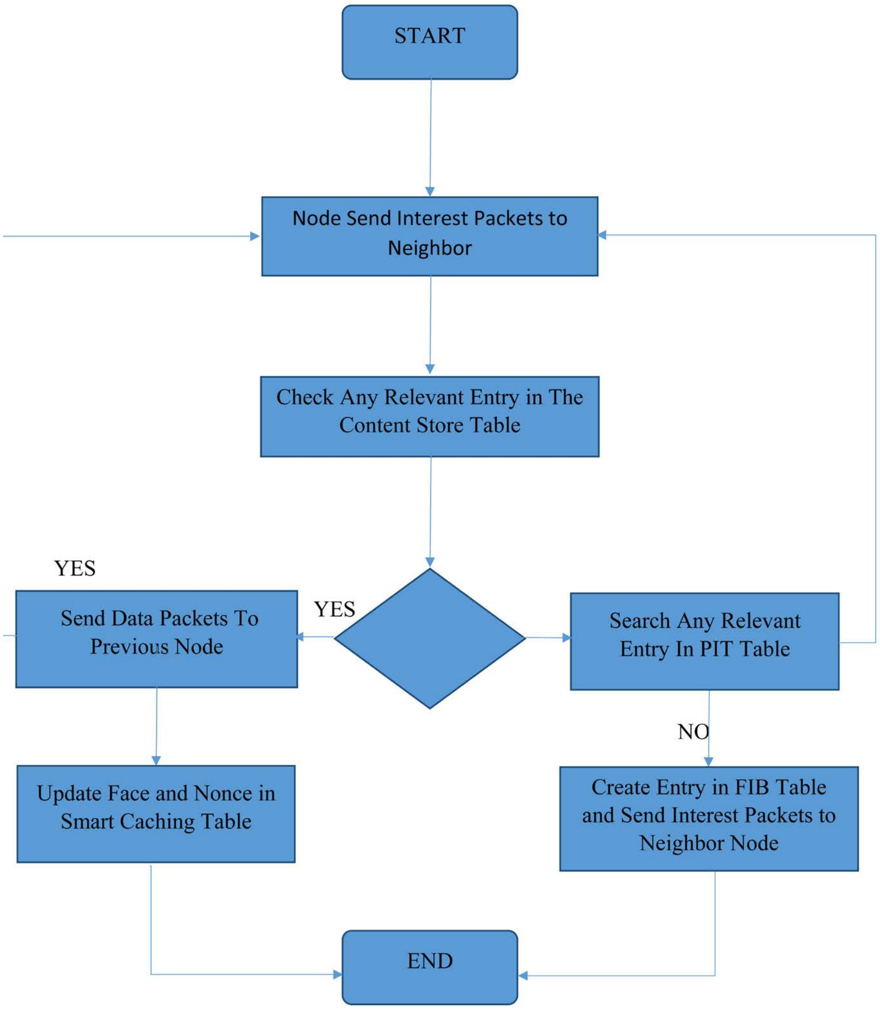

START
Node Send Interest Packets to Neighbor
Check Any Relevant Entry in The Content Store Table
YES
Send Data Packets To Previous Node
YES
Search Any Relevant Entry In PIT Table
NO
Update Face and Nonce in Smart Caching Table
Create Entry in FIB Table and Send Interest Packets to Neighbor Node
END

Figure 3. Transaction of interest and data packets.

transaction of mainly two types of Packets Pi and Pd in NDN development. At the initial level, a very first time when node request or we can say that send interest packet to its neighbor, neighbor check relevant entry in Content Store table.

If requested packet is found in CS then neighbor node send data packet to node and update Smart Caching Table. If it is not found there then neighbor node will check in PIT table. If neighbor node finds relevant entry in PIT table then it will wait for incoming data packet. If not then one entry is created in FIB table and send Interest Packet to neighbor node.

SCTSmart Caching strategy, we have assumed time as a nonce in the table and Face_Value is a value of counter which describes Interest of data. As shown in Figure 4, first we check any relevant entry is there in smart caching table. If not found then entry must be created. If any relevant entry is there then the Face_Value will be incremented by 1 with the condition (Face_Value) > (Max_Face_Value) and (Current_Time –nonce) <= (Max_Update_Time). If it does not satisfy the condition then said entry would be removed from CS.

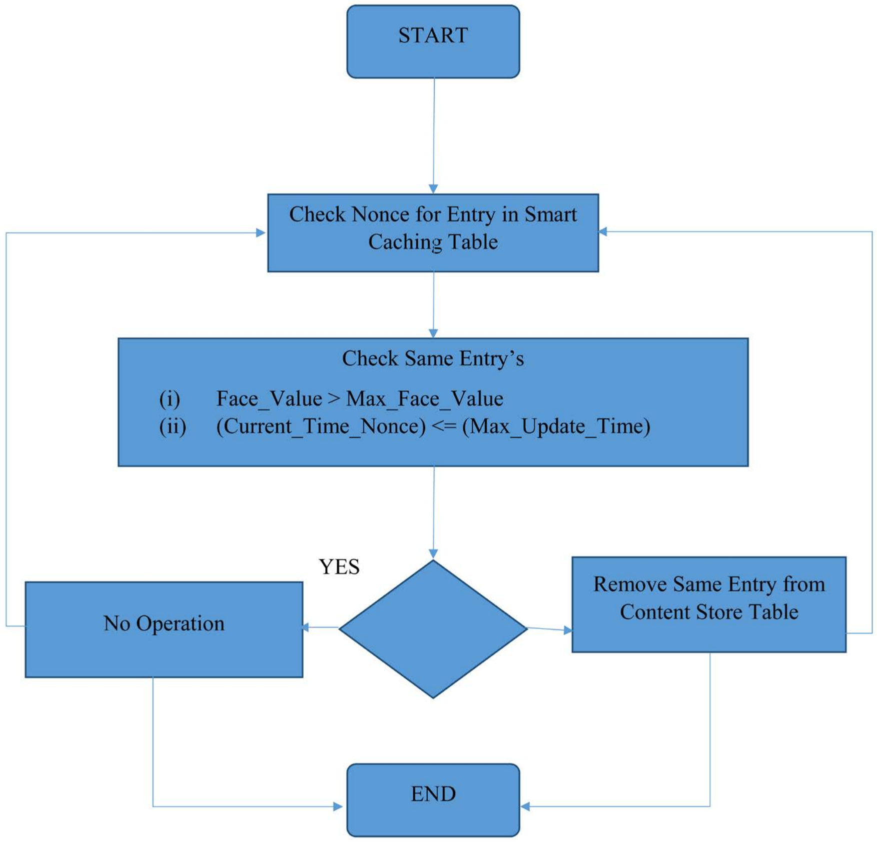

START
Check Nonce for Entry in Smart Caching Table
Check Same Entry's
(i) Face_Value > Max_Face_Value
(ii) (Current_Time_Nonce) <= (Max_Update_Time)
YES
No Operation
Remove Same Entry from Content Store Table
END

Figure 4. Smart caching strategy flow chart.

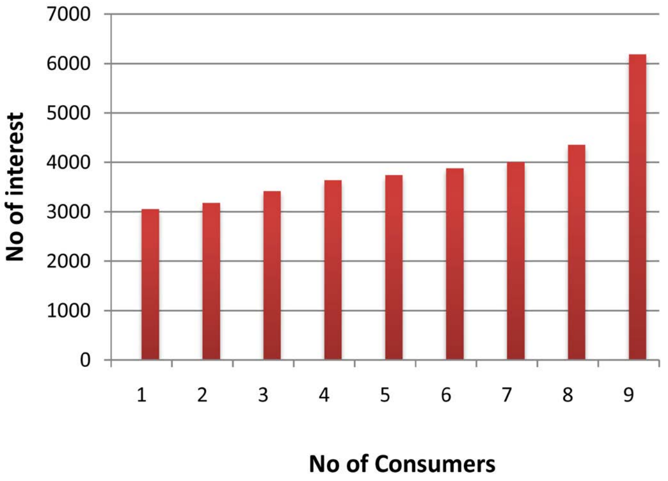
For implementing NDN, NFD is very useful network forwarder tool available with ndnSim that work as instrument and elaborated jointly with the NDN protocol. The major design goal of NFD is to forward Pi and Pd in addition to that to support different strategic environments for forwarding process like best rout strategy, broadcast strategy, client control strategy etc. in NDN. Forwarding Pi and Pd packets, NDN deploys lower-level network transport mechanisms to process packets into NDN Faces which maintains collection of fundamental data structures like CS, PIT, FIB etc that we already discussed. Apart from basic packet forwarding and processing, NFD also provides multiple forwarding techniques [10].

In simulation, we have assumed different simulation parameters as listed in Table 2.

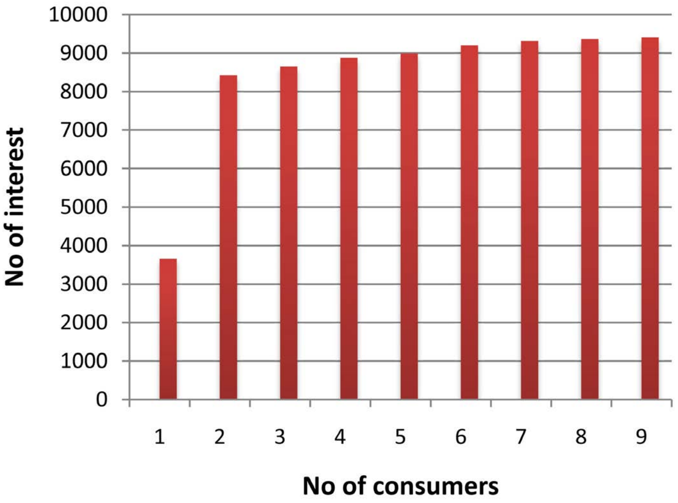
4. Results and Analysis

After implementing our proposed scheme, we have done comparative analysis of energy consumption and freshness achieved by our smart caching scheme and No Caching, Caching Everything schemes by varying no of consumers, refresh rate, and content store capacity in lossy channel environment for different transactions.

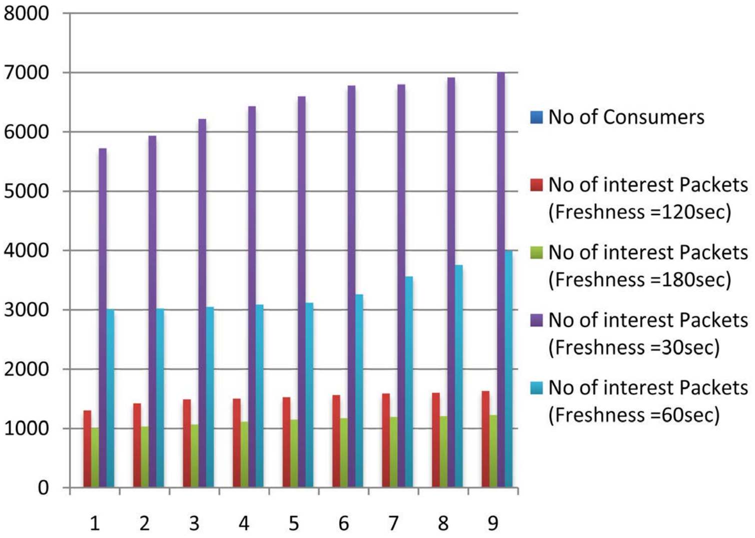
Here, from above results analyzed from Graphs 1-4, we can estimate the rate of interest packets, *i.e.* amount of Interest packets to be generated by varying of



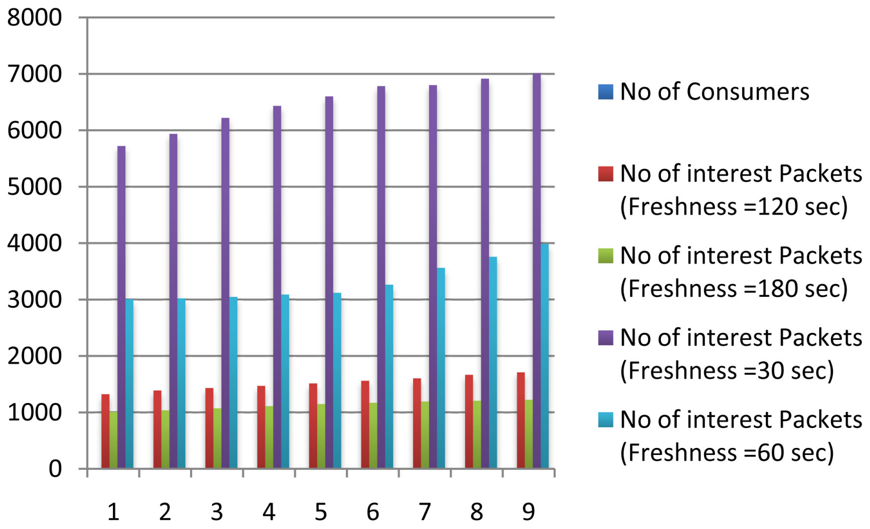
Graph 1. No of consumers Vs No of Interest with no caching.



Graph 2. No of consumers Vs no of interest with caching.



Graph 3. No of consumers Vs no of Interest with smart caching scheme with respect to different refresh rate & content store size = 10 KB.



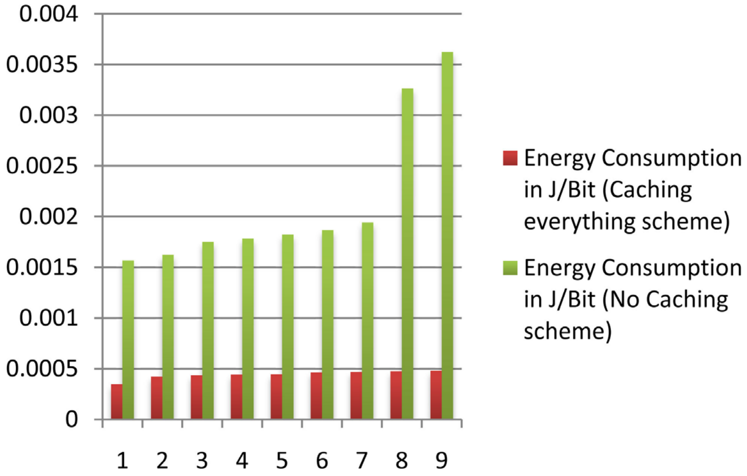
Graph 4. No of consumers Vs no of Interest with smart caching scheme with respect to different refresh rate & content store size = 20 KB.

Table 2. Simulation parameters.

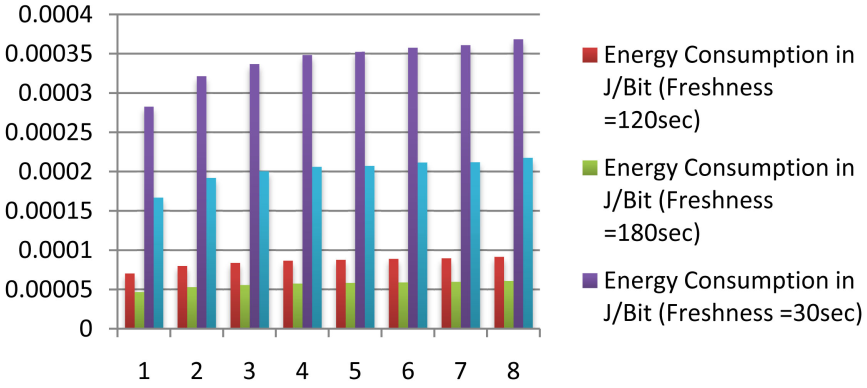
No of Nodes	60
No of Subscribers/Consumers	1 to 8
Mobility	Random Walk Mobility model
Network	802.11a
Refresh Period	30Sec, 60 Sec, 120 Sec, 180 Sec
Caching Strategy	a. No Caching, b. Caching Everything c. Proposed Smart Caching
Content Store Capacity	10 KB, 20 KB, 30 KB
Max_Update_time	Refresh Period
Max_Face_value	10
Simulation Time	10 Minutes
Channel	Lossy
Initial Energy of Node	1 Jule
Energy Required for Transaction (Etran)	50 nJ/bit
Energy Required for Data Caching (EDC)	10 nJ/bit
Energy Required for Amplifier (Eamp)	10 pJ/bit/m^2

consumers and keeping content store size = 10 and 20 KB respectively. For different periodic intervals like *i.e.* 30 Sec to 120 Sec, less amount of interest packets are required to remain present in NDN in case of our proposed Smart Caching scheme as compared No caching schemes. Our approach achieves better result even at lower refreshing rate due to adding time as nonce additionally. This dynamics adds demand based data retrieval with maximum updates.

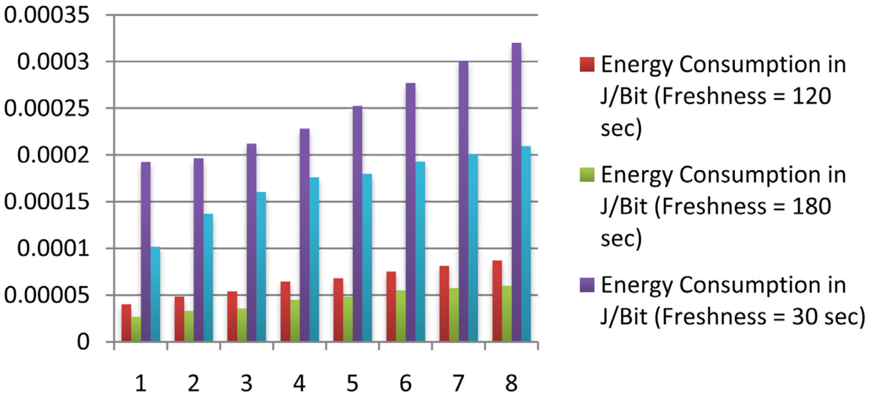
From Graphs 5-7 it can be clearly observed that in case of No Caching and Caching everything schemes, Consumer Energy requirement is gradually in-



Graph 5. Comparison of energy consumption for no caching and caching everything schemes.



Graph 6. Energy consumption in smart caching strategy with content store size = 10 KB.



Graph 7. Energy consumption in smart caching strategy with content store size = 20 KB.

creased due to amount of attempt for maintaining freshness of data seems to be power consuming per bits, this requirement get degraded in case of our proposed scheme.

It is obvious that the rates of Interest packets are higher using No Caching scheme as compare to Caching and our proposed smart caching scheme at NDN router as intermediate NDN hosts is not trained to store data packets during the transaction for future use. Our approach achieves reduction in no of interest packets at regular time interval; In our implementation the cache gets refreshed to optimize the no of total available interest packets present in NDN at the freshness rate of 180 sec, 120 sec, 60 sec and 30 sec . No of Interest packets are increased with respect to freshness of data, on other side, no of interest packets are reduced as size of content store is increased gradually which establish that Freshness of data is always directly proportional to communication cost leading to energy consumption.

5. Conclusion and Future Scope

Using energy efficient smart caching strategy we would be able to achieve network lifetime .We can say that freshness is in proportional to the cost. In our work, energy consumption in No Caching strategy is five times higher than Caching Everything strategy. While energy consumption in Caching Everything strategy is eight times greater than Smart Caching strategy which is increased as No of consumers and freshness of data are increased. The future work will comply of the real world deployment of the algorithm that is proposed in our work. Enhancing energy and freshness by deploying other variants of caching schemes can be open research area.

References

[1] Jacobson, V., Burke, J., Zhang, L., Zhang, B., Claffy, K., Krioukov, D., Papadopoulos, C., Abdelzaher, T., Wang, L., Yeh, E. and Crowley, P. (2014) Named Data Networking (NDN) Project 2013-2014 Report. Palo Alto Research Centre.

[2] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T.C. and Whlisch, M. (2014) Information Centric Networking in the IoT: Experiments with NDN in the Wild. *Proceedings of ACM ICN*, Paris, 24-26 September 2014, 77-86.

[3] Chen, X., Fan, Q. and Yin, H. (2013) Caching in Information-Centric Networking: From a Content Delivery Path Perspective Innovations. 9*th International Conference on Information Technology*, Abu Dhabi, 17-19 March 2013, 48-53.

[4] Hail, M.A., Amadeot, M., Molinarot, A. and Fischer, S. (2015) Caching in Named Data Networking for the Wireless Internet of Things. *IEEE International Conference on Recent Advances in Internet of Things*, Singapore, 7-9 April 2015, 1-6. https://doi.org/10.1109/riot.2015.7104902

[5] Quevedo, J., Corujo, D. and Aguiar, R. (2014) Consumer Driven Information Freshness Approach for Content Centric Networking. *IEEE Conference on Computer Communications Workshops*, Toronto, 27 April-2 May 2014, 482-487.

[6] Chen, X., Fan, Q. and Yin, H. (2014) CANR: Cache-Aware Name-Based Routing. *IEEE* 3*rd International Conference on Cloud Computing and Intelligence Systems*, Shenzhen, 27-29 November 2014, 212-217.

[7] Thar, K., Ullah, S. and Hong, C.S. (2014) Consistent Hashing Based Cooperative Caching and Forwarding in Content Centric Network. *Network*, *Operations and Management Symposium*, Hsinchu, 17-19 September 2014, 1-4. https://doi.org/10.1109/apnoms.2014.6996598

[8] Liu, T., Tian, M. and Cheng, D. (2013) Content Activity Based Short-Cut Routing in

Content Centric Networks. 4*th IEEE International Conference on Software Engineering and Service Science*, Beijing, 23-25 May 2013, 479-482.

[9] Meng, X., Zheng, L., Li, L. and Li, J. (2015) PAM: An Efficient Power-Aware Multi-Level Cache Policy to Reduce Energy Consumption of Software Defined Network. *International Conference on Industrial Networks and Intelligent Systems*, Tokyo, 2-4 March 2015, 18-23.

[10] Afanasyev, A., Moiseenko, I. and Zhang, L. (2015) NDN SIM: NDN Simulator for NS-3. Technical Report NDN-0028.