An Information-Centric Networks- A Survey on Applications & Cache Management

Monika Pawar¹, Aishwarya Mishra²

1,2Department of Computer Science & Engineering, I.E.S. College of Technology, Bhopal, India

Abstract— Information centric network is a special form of computer network that is created for selected or registered nodes only. The information is stored on a central node and all nodes are using it. But the information is available for registered or specific nodes that are attached with it. Information-Centric Networking (ICN) is a current trend in research and development. As it uses centralized information storage concept hence accessing should be fast enough to satisfy all connected nodes requirements or demands. Due to that efficient caching scheme is essential. In this paper we are presenting some techniques offered by various researchers in this field.

Keywords— Cache Management, Information Centric Network, Automatic Cache Management, Distributed Optimization.

I. INTRODUCTION

Information-centric networking or ICN is a special kind of computer networking in which users are allowed to their needy data apart from referenced data or physical location. This network also commonly known as with its another names also like data-oriented networking, content-based networking or content-centric networking. ICN is emerging networking environment for vast majority of Internet activities that are related to specific information access and delivery. Such types of networking are developed for better and efficient use of internet. For these specific applications were developed like mission critical systems, interactive applications for business and social interactions. The purpose of these applications are to archive some advantages like protection, dependability, quality of service and resilience [1].

The most benefit of Internet is communication that enabling the delivery of data among pairs of end hosts. Unfortunately end-point centric model is no longer fulfilling the current communications needs. Advanced users focused on the desired specific information while the underlying network focuses on the end-to-end communication among end-hosts. As far as current management techniques in Information-centric networking are concerned, self-management intelligence, dynamic caching, flexible and adaptive to changing conditions and feedback based closed-loop control solutions. These benefits make information centric network efficient and strong. Multicache is one of the schemes that enhance the fast accessing of data among end users.

MultiCache acquires advantage of information-awareness to get better network utilization through resource sharing. To use this scheme network should deploy and control proxy superimpose routers to facilitate the joint provision of multicast and caching to pursue both synchronous and asynchronous requests [2].

In this scheme end user interact through simply providing flat location independent identifiers for the desired content without appealing in the process of locating an end-host’s data. MultiCache aims to establish an information-centric model of communication that better reflects current Internet usage patterns, in order to facilitate the deployment of resource sharing mechanisms. The introduction of information-awareness into the network enables the network to identify pieces of information [2], another scheme known as universal caching also used to enhance the Information-centric networking. It is straightforward that the network delay and publisher load can be reduced by having content requests satisfied at network edge as much as possible. This scheme is capable to reduction both network delay and publisher load [3].

As far as specific solution provided by ICN is concerned, it supports following functionality to serve its services [4].

Content-centric request/reply model for data distribution: In this category users are interested to store their data that is propagated to numerous network nodes according to their choice and liking made by serving nodes. According to the interest of demanding node data is transferred from serving node to demanded node.

Route-by-name operations: Every data item has a unique name along with routing of all related information is Performed according to this unique name. This permits the interaction among interested users in known content and the closest copy of the desired data item.

In-network caching: Routing information of end-to-end path among nodes are stored by user in local cash content. After reception of interested data, router confirms about proper storage of requested piece of content in the local cache. If the interested content is not propagated to the end user after requesting by demanded user, it may be due to inefficient availability of content or cache.
Security embedded in the content: It is most important to provide security among end users that are using same application. ICN security works on both security among user along with the protection of data content also. Fake version of data content should be eliminated and actual version of data content should be preserved.

ICN network leads to the design problems of efficient caching strategies, such as optimal dimensioning of caches and intelligent content placement. The rest of paper is organized as follows. In Section II describes about background details of ICN. Section III describes about propose related work offered by various researchers. Section IV describes about conclusion.

II. BACKGROUND

The ICN network is inherently created and managed as a multi-level caching system, with the caches physically distributed across the network and the user requests originated in geographically dispersed nodes. When an object is requested from an intermediate node then if the cache has a copy of demanded content, the response is returned locally. Generally it is compulsory to scattered contents hierarchically in the network like with most popular contents at the edge, less popular contents near the edge and least popular contents even further. According to this mechanism the aggregated cache hit rate of the network was maximized. Due to this the network delay and publisher load should be reduced. When delay is concern, it is clear that the closer a node with a content copy is to a user, the less time the user requires to get the content. It must be observable that caching strategies can utilize storage capacity to absorb latency by replicating the most popular contents.

III. RELATED WORK

This section describes some related work to cache management in information centric network.

Distributed cache management architecture for ICN that dynamically reassigns information items to caches was recommended by Sourlas at al [1]. Depending on the observed item request patterns distributed managers make information item replacement decisions like popularity and locality. This will help to minimize the overall network traffic cost imposed by user’s requests. They derived four distributed online intra-domain cache management algorithms. They also categorized them according to the level of cooperation needed and compare them in terms of message overhead, complexity, performance and convergence time. as per the results this cache management algorithms perform closely to the derived lower bound. Autonomic self-management envisages systems that can manage themselves given high-level objectives by administrators.

Extending autonomic management from individual devices to the collective self-management of networks of such devices results in autonomic networking. Current approaches applied to Content Distribution Networks (CDNs) follow static off-line approaches with algorithms that decide the optimal location of caches and the assignment of information items and their replicas to those caches based on predictions of content requests by users [1].

In this respect the deployment of an intelligent substrate architecture that enables the assignment of information items to caches to take place in real-time, adapting to the ever changing user demand patterns. Distributed Cache Managers (CMs) decide the items every cache stores by forming a substrate that can be organized either in a hierarchical manner for scalability reasons or in a peer-to-peer organizational structure. The required information such as request rates, popularity/locality of information items and current cache configurations, is exchanged between the distributed cache managers through the intelligent substrate functionality. Every cache manager decides in a coordinated manner with other managers whether to cache an item. This may require the replacement of an already stored item. The replacements are performed towards maximizing a network wide utility function [1].

A MultiCache based architecture of information centric network was suggested by Katsaros et al [2]. Particularly they proposed the architecture that takes advantage of information-awareness to improve both the utilization of network resources and the end-user experience. They also compare a MultiCache-based content distribution application with the popular BitTorrent application. As per demonstrated result a significant reduction of traffic load, in conjunction with considerably lower download times. This architecture is based on overlay network, based on the deployment of additional infrastructure inside access networks. By deploying MultiCache inside access networks, content is cached close to the clients, facilitating the discovery of caches in the clients’ networking vicinity and therefore enabling the localization of traffic. Similarly in contrast with typical end host deployment scenarios dedicated to the maintenance overhead of the routing substrate and mitigating servers is expected to provide lower churn rates [2].

They designed MultiCache-based content distribution application that can be directly compared with BitTorrent. In this application, a content provider employs content fragmentation to create multiple trees for the delivery of a single file. Fragment identifiers are retrieved by end-hosts out-of-band or MultiCache-torrent file.
Content distribution via BitTorrent is an obvious case of information-centric communication, forced by the current, information agnostic, Internet to be controlled by end-hosts. They considered different bandwidth allocations for the uplink and downlink directions of access links, as current access technologies. The utilization of network resources is measured in terms of the egress inter-domain traffic (EIT) and intra-domain traffic load (ITL) incurred for the delivery of the content. The EIT metric reflects the total number of bytes of egress traffic at each stub AS, thus expressing the amount of traffic that network operators must pay for in order to reach a transit domain. The ITL metric measures the aggregate number of block transmissions over all links of an AS, allowing us to assess the load imposed within each administrative domain [2].

Age-based cooperative cache scheme was suggested by Ming et al [3] for information centric network. This scheme focuses on reducing network delay and publisher load of ICN. They achieved this by providing lightweight cooperative mechanism under age-based control. This scheme spreads popular contents to the network edge while at the same time eliminated unnecessary content replication at intermediate network nodes. They designed two light-weight algorithms to realize this scheme. They evaluated the impact of the algorithms under real trace and realistic network topology. The age-based cooperative scheme dynamically configure content’s age in an implicit cooperation manner among ICN routers. This scheme increases the aggregated in-network cache hit probability, thus reduce network delay and publisher load simultaneously [3].

A light-weight cooperative algorithm by dynamically adjusting content’s age at different network nodes were also designed. These algorithms ensure that a router keeps data longer when it serves as a leaf router in one multicast group, and holds data shorter if it serves as an intermediate router in other groups. This algorithms focus on the cache management of a single content. In practice, contents are often split into a sequence of chunks. To a large extent, this avoids the problem that a replica is cached in an intermediate router but is never used. The age is determined for each content object using some factors like Distance to the server and Popularity of the Content. The results showed that this scheme can achieved significant gains to enhance network performance [3].

Veltri et al [4] focused on an ICN framework named COntent NETwork (CONET), and on a specific implementation based on this framework known as coCONET. In the CONET framework each piece of content or chunk is characterized by a unique name. When a user is interested in a given piece of content, its terminal generates an interest message and this message is forwarded by network nodes towards the origin node.

This forwarding process is based on the name of the content that is included in the interest message. When content send back to interested node that time intermediate node cache it. In the specific coCONET solution forwarding mechanism of interest messages is called “lookup-and-cache”. If a node doesn’t know the next-hop node to forward an interest message, send a query to Name Routing System. Name Routing System (NRS) is providing answers about necessary routing information. The decoupling of switching/forwarding function from the routing and control functions is managed by Software Defined Network (SDN) paradigm [4].

Information-centric networks can enable fully functional packet-level caches and use them as general network components. Arianfar et al [5] look at some of the inefficiencies that can be managed with packet caches, specifically from the transport layer point of view. The router cache contains two main components: a packet store and an index table to access the store. Packets are assigned to different cells in the packet store. The index table contains information to find the packet in the packet store. The packet store is large and can be kept in the router’s DRAM The size of index table is smaller, thus it can be kept in SRAM. A packet store (DRAM) cell address can be pre-defined based on the index position or it can be calculated based on available free space at the time of insertion. In this model almost all the packets entering the router are indexed and cached. They further extended this model with random fractional caching, where each packet has a pre-set probability of getting indexed and cached [5].

Muscariello et al [6] focused on the performance evaluation of ICN networks and develop an analytical model of bandwidth and storage sharing under limited resources. Each user is supposed to implement a receiver-driven flow control protocol yielding fair and efficient bandwidth utilization along the path to the content, while content storage is managed by least-recently used (LRU) per-chunk replacement policy. Under these assumptions, a closed-form characterization of the average content delivery time is provided that captures the tradeoff among user performance and limited network resources. An interesting application of this model is the optimal dimensioning and localization of storage resources in a given bandwidth capacity setting [6].

Sourlas et al [7] proposed the deployment of an intelligent substrate architecture that will enable the assignment of information items to caches to take place in real-time, based on changing user demand patterns. Distributed Cache Managers decide the items every cache stores by forming a substrate that can be organized either in a hierarchical manner for scalability reasons or in a peer-to-peer organizational structure.
Communication of information related to request rates, popularity/locality of information items and current cache configurations, takes place among the distributed cache managers through the intelligent substrate functionality. In the emerging ICN proposals, information is replicated almost ubiquitously throughout the network with subsequent optimal content delivery to the requesting users. Thus, efficient placement and replication of information items to caches installed in network nodes is key to delivering on this promise. Current approaches applied to Content Distribution Networks follow static off-line approaches with algorithms that decide the optimal location of caches and the assignment of information items and their replicas to those caches based on predictions of content requests by users [7].

Other approaches can also be realized in which manager’s base their decisions on a local view of the user demand for specific items but coordinate to maximize the overall network gain, as well as solutions where managers act selfishly aiming at maximizing their own local utility. Since all the above decisions are made in a distributed manner, uncoordinated decisions could lead to suboptimal and inconsistent configurations. Coordinated decision making of a distributed cache management solution can be achieved through the substrate mechanisms, by ensuring that managers change the configuration of each cache in an iterative manner i.e. one at a time and not autonomously in a potentially conflicting manner. All these approaches are investigated in this work by proposing and comparing different distributed cache management algorithms and evaluating their performance with respect to their autonomicity. Distributed on-line cache management Algorithms presented three distributed, gradient descent type, on-line cache management algorithms that capture the particularities of the volatile environment under consideration. All of them are adaptive to popularity and locality changes of the user demands. To achieve this each manager may update the contents of its corresponding cache, by fetching new items and replacing existing ones. Nevertheless this mechanism differs in the amount of information that needs to be communicated through the substrate, the required level of coordination among the cache managers, and the performance objective [7].

Psaras et al [8] concerned with cache management operations that have to be adjusted to fit in a completely decentralized and uncoordinated environment. They focused on the allocation of the available cache capacity along a path of caching entities among different content flows. As a starting point we intuitively observe that caching each content in every cache-enabled device along the delivery route, inherently causes huge caching redundancy.

They tried to reduce caching redundancy and make more efficient use of available cache resources, in order to reduce overall network utilization and potentially increase user-perceived quality. In this scheme, problem of content placement within a system of caches from the path caching capability point of view. They proposed ProbCache, an algorithm that approximates the capability of paths to cache contents, based on path lengths, and multiplexes content flows accordingly. The ultimate goal of ProbCache is to utilize resources efficiently, reduce caching redundancy and in turn, network traffic redundancy. Their results suggest that there is a lot of space for resource management optimization of in-network caching policies [8].

A study on Information-Centric Networking was presented by Bengt Ahlgren et al [9]. They compared and discussed some of the features and design choices of the 4WARD Networking of Information architecture (NetInf), PARC’s Content Centric Networking (CCN), the Publish-Subscribe Internet Routing Paradigm (PSIRP), and the Data Oriented Network Architecture (DONA). All four projects take an information-centric approach to designing a future network architecture, where the information objects themselves are the primary focus rather than the network nodes. In CCN the term content-centric is used with essentially the same meaning. The Information-Centric Networking (ICN) paradigm is addressing a set of issues of the current Internet and content distribution architectures. In general, the need for scalable and efficient content distribution has fueled a proliferation of overlay networks such as Peer-to-Peer (P2P) networks and Content Distribution Networks (CDNs). This has introduced information access models where endpoints essentially try to access named content, without actually requiring a transport layer session that is bound to a specific host [9].

Rodriguez et al [10] derived models to calculate the latency experienced by the clients, the bandwidth usage, the disk space requirements, and the load generated by each cache cooperating scheme. They developed analytical models to study and compare the performance of both hierarchical and distributed caching. Analytical models can explore the different trade-offs of the different cache cooperating schemes and simulate different scenarios. The caching document copies at the access points of intermediate Internet Service Providers (ISPs) reduces the connection time compared to the case where there is no support from intermediate caches in the network and there are only edge caches. They also find that distributed caching has lower transmission times than hierarchical caching since most of the traffic flows through the less congested lower network levels. The hierarchical caching has lower bandwidth usage than distributed caching, since hierarchical caching uses intermediate caches in the network.
However, distributed caching distributes the traffic better, using more bandwidth in lower network levels. Further analysis of hierarchical and distributed caching showed that the disk requirements for distributed caching are much smaller than for hierarchical caching [10].

Borst et al [11] developed a cache management algorithm aimed at maximizing the traffic volume served from the caches and minimizing the bandwidth cost. They focus on a cluster of distributed caches (Content Distribution Networks), either connected directly or via a parent node, and formulate the content placement problem as a linear program in order to benchmark the globally optimal performance. Fan et al [12] addressed the issue of scalable protocols for wide-area Web cache sharing. They compared network traffic and CPU overhead of proxies using ProWGen with proxies that are not using ICP. The results show that even when the number of cooperating proxies is as low as four, ICP increases the interproxy traffic by a factor of 70 to 90, the number of network packets received by each proxy by 13% and higher, and the CPU overhead by over 15%. After that they proposed a new cache sharing protocol called “summary cache.” Under this protocol, each proxy keeps a compact summary of the cache directory of every other proxy. When a cache miss occurs, a proxy first probes all the summaries to see if the request might be a cache hit in other proxies, and sends a query messages only to those proxies whose summaries show promising results. If the request is a cache hit when the summary indicates otherwise the penalty is a higher miss ratio. The summary cache enhanced ICP protocol and implements a prototype within the Squid proxy [12].

In a single-level proxy cache is Web proxy caching performance to certain workload characteristics. Busari et al [13] developed a synthetic Web proxy workload generator that offers the required flexibility. The workload generation tool is then used in a sensitivity study of proxy cache replacement policies. The goal is to examine the sensitivity of proxy caching policies to certain workload characteristics. Therefore, ProWGen incorporates only those characteristics deemed relevant to caching. In addition, ProWGen models only the aggregate client workloads seen by a typical proxy server, rather than individual clients. The trace-based approach uses an empirical trace and either samples it or permutes the orderings of the requests to generate a new workload. This approach is straightforward to implement but has limited flexibility, since the workload is inherently tied to a known system. The analytical approach uses mathematical models for the workload characteristics of interest, and uses random number generation to produce workloads that statistically conform to these models. This approach, though challenging, offers a lot of flexibility.

The workload generator incorporates five selected workload characteristics, which are deemed relevant to caching performance. These workload characteristics are: one-time referencing, file popularity, file size distribution, correlation between file size and popularity, and temporal locality. In ProWGen, the user specifies the percentage of one-timers desired in the synthetic workload. Once the value of this parameter is specified, the number of one-timers in the workload is determined, and their reference counts are fixed at one. The results showed that policies are relatively insensitive to the percentage of one-timer documents in the workload, and to the Pareto tail index of the heavy-tailed document size distribution [13].

IV. CONCLUSION

The information-centric approach to the network of the future is currently being explored by a number of researchers. The development of Information-Centric Networking (ICN) concepts is one of the significant results of different international Future Internet research activities. In a network based on the information-centric paradigm, the information objects are first-class citizens with unique identity independent of the device they are stored on. This approach enables efficient and application-independent information caching by the network infrastructure, and thus allows large-scale information distribution without violating basic assumptions or resorting to special tweaks or add-ons. In this paper we have shown some caching techniques.

Acknowledgment

I would like to thank Prof. ………………………………, for accepting me to work under his valuable guidance. He closely supervises the work over the past few months and advised many innovative ideas, helpful suggestion, valuable advice and support.

REFERENCES


