Modelling Queuing Delays in Content-Centric Networks

Y Tofis†, I Psaras ‡, G Pavlou‡

Department of Electronic and Electrical Engineering,

University College London, Torrington Place, WC1E 7JE. England, UK

Email: uceeyto@ee.ucl.ac.uk

Abstract: Router's reactive caching or packet caching on the network level is an element of paramount importance as far as the performance of the Content-Centric Networks are concerned. This paper aims to provide an analytical model for the total end to end average queuing delay of the Content-Centric Networks based on the cache replacement policy.

1. Introduction.

Content distribution and retrieval is expected to be the key characteristic of next generation/future internet. Content Centric Networking (CCN) proposes a new structure in the computer networks based on the content that the user needs rather than location that the content is stored; an approach that current communications are rely on. In this new approach, content would be routed according to its name (named data or content) rather than an IP address that indicates host's location (named host) [1].

This new approach of internet architecture will have some substantial benefits such as increased scalability, security (implemented at the network and data link layers) and performance at the same time, achieved from the simpler configuration of network devices and content caching [2,3]. At this paper we have focus on the latter aspect, examining the improved average total response time (namely the average time from browser's request of a content-packet until its reception) and reduced congestion achieved by employing this caching mechanism to each router of the Content Centric Network.

2. Delay analysis for the total end to end queuing delay of the network.

Router caching is a mechanism that satisfies content packet requests, namely packet requests for web objects, media files, software, documents and other downloadable objects, applications, real time media streams as well as queries about DNS, routes, and databases, on behalf of an origin Content Server Source or Origin (the site that maintains the current, authoritative version of the requested content packet) [1]. Every router in the CCN structure is equipped with a caching storage in which stores recently requested packet copies, depending on the adopted replacement policy. Each packet request originated from the access network is firstly forwarded to the access router's cache storage and a check is performed in order to deduce if a copy of the requested packet is cached (cache hit). In the case of a cache hit, the request is satisfied and the cached content is sent back to the client. In the case that the cache doesn't contain a copy of the requested packet (cache miss) then the request is forwarded to the next router towards the path to the Content Server Source. The procedure is repeated until the request reaches the Content Server Source, where it is eventually satisfied.

After a copy of the requested packet has been located, either at an intermediate router cache memory or at the Content Server Source, the packet is sent back to the client following the reverse path of the requesting packet. During its route to the client, the requested packet is received and cached by the routers of the path under consideration.

In this paper we will study the queuing delay, that consists a quite accountable component of the nodal total delay. Below we characterise the queuing delay using the metric of the total end to end average queuing delay. For the purposes of the developed model, we use the following nomenclature:

M: the total number of the packets transmitted in the community network under consideration

λ_i: average request rate for packet i

$\beta = \sum_{i=1}^{M} \lambda_i$: average rate at which the content packets arrive at the access router's queue (in units packets/sec

L: average size of content packets (in bits)

$L\beta$: average rate at which bits arrive at the queue (bits/sec)

R: the bit rate or throughput or transmission rate (in bits/sec)

By defining the aforementioned quantities we can now define the traffic intensity, an average rate of which bits arrive at the router's queue to the rate that bits are transmitted, namely are pushed out of the queue:

$L\beta/R$: traffic intensity

Assuming that the router's queue can handle an infinite number of content packets, we examine the case that the traffic intensity is less than one ($L\beta/R<1$), otherwise the queue will enlarge in terms of time resulting in an infinite queue delay [4].

r: cache hit rate. It is the fraction of the requests that are satisfied by the cache

$m=1-r$: cache miss rate

N: the size of the cache in terms of packet places

We consider that the content requests are negligibly small, thus they don't contribute to the traffic intensity. Taking into account that the average length of the queue at the *M/M/1* queuing systems is given by $\frac{\beta\Delta}{1-\Delta\beta}$ [5], where Δ is the average time required to send a packet over the access link and β is the arrival rate of packets to the access link (in consistency with our notation Δ is equal to the $\frac{L}{R}$) then the total average waiting time at the queue of the access router would be $\frac{\Delta}{1-\Delta\beta}$.

According to the Content Centric Networks (CCN) structure, the intermediate routers act both as a client and a server at the same time[1,2]. When a router receives requests from and sends packets to the client, it acts as a server. In the case that it sends (forwards) packet requests to and receives packets from adjacent routers or the Content Server Source, it acts as a client. This fact enables us to use a modified form of the aforementioned formula in order to the estimate the total average response time.

Taking into account the fact that the cache rate is r we can claim that the traffic intensity of each router can be reduced by a factor of $(1-r_c)$, where $c=1...N$, the number of intermediate routers from the end user to the Content Server Source.

Thus, after $N-1$ hops (N routers) the **T**otal **A**verage network **Q**ueuing **D**elay would be equal to:

$$TAQD(N)=\sum_{c=1}^{N}\frac{\Delta}{1-\Delta\beta\cdot\prod_{j=1}^{c}(1-r_j)}=\sum_{c=1}^{N}\frac{L/R}{1-\frac{L\beta\cdot\prod_{j=1}^{c}(1-r_j)}{R}} \quad (2.1)$$

We should have in mind that $R=min\{R_{j,j+1},\ 1<j<N-1\}$. Usually the bottleneck is located at the access point and so R is equal to the access link capacity.

The cache hit rate (r_c) can be be modelled according to the replacement policy adopted at the cache and assuming Poisson arrivals for for each packet i.

- In the case of Time to Live (TTL) cache replacement policy [3] then the hit rate (the supplementary of miss rate) is given by the relation:

$$r_c=1-m_c=1-P(\textit{having zero arrivals of packet } i \text{ in TTL time space})=$$
$$1-P_0(\tau=TTL)=1-e^{-\lambda_i\cdot\prod_{j=1}^{c}(1-r_j)\cdot TTL} \quad (2.2)$$

- In the case of Least Recently Used (LRU) and First In First Out (FIFO) cache replacement policy [3] we start our observations at time t=0 where the cache memory is empty and we examine the Poisson arrivals for each subsequent time space $\frac{N}{\beta}$ which is the average time that the size of the cache memory gets full if it is empty at $t=0$. The miss rate is given by the probability of the following event of "having zero request arrivals for i in the observation time space and to have at the same time arrivals (from the set of $M-1$ remaining packets, i' from now on) of at least N different packets. We say at least N because in that case the cache memory will become full of packets from the i' set, meaning a cache miss at the subsequent "observation slot" for the packet i.

  We say N different because in the case that two or more requests arrive (at the same "observation slot") for the same packets of the i' set, then these requests will reserve only one place in the memory, since

they are related to the same (only one) packet". The probability of the previously described event can be quantified with the following expression:

$$r_c = 1 - m_c = 1 - P\left(\text{to have zero request arrivals for packet } i \text{ for the time space } \frac{N}{\beta} \right) \cdot \sum_{n=N}^{M} \left(\frac{M!}{(M-n)!} \cdot P\left(\begin{array}{c} \text{to have one request arrival of each of the } n \text{ (at least } N) \\ \text{mutuallydifferent packets from } i' \text{ set} \end{array} \right) \cdot P\left(\text{to have zero request arrivals for the } M-n-1 \text{ remaining objects of } i' \right) \right) =$$

$$1 - e^{-\lambda_i \cdot \prod_{j=1}^{c} (1-r_j) \cdot \frac{N}{\beta}} \cdot \sum_{n=N}^{M} \frac{(M-1)!}{(M-n-1)!} \cdot \prod_{i=1}^{n} \left(\frac{\left(\prod_{j=1}^{c} (1-r_j) \lambda_i \cdot \frac{N}{\beta} \right)^n}{n!} \cdot e^{-\lambda_i \cdot \prod_{j=1}^{c} (1-r_j) \cdot \frac{N}{\beta}} \right) \cdot \prod_{i=1}^{M-n-1} \left(e^{-\lambda_i \cdot \prod_{j=1}^{c} (1-r_j) \cdot \frac{N}{\beta}} \right) \quad (2.3)$$

where $\frac{(M-1)!}{(M-n-1)!}$ are the permutations (order matters since the arrival of the same packets from i' in a different order is another way of arrival arrangement and should be counted separately) of the choosing *M-n-1* packets from the whole number *M-1* of *i'* elements.

3. Modelling the cache state using Random Walk Theory.

Each state of the random walk model illustrates the router (or node) where the cached packet of interest *i* is located. We note that the packet *i* can possibly be cached at the same time in the cache memory of consecutive routers. In our analysis we consider as "state" the closest router from the end user that has in its cache memory the packet of interest *i*. The initial state $\mathbf{S_N}$ is the Content Server Source. One Random Walk chain corresponds to each access network (and thus to each **A**ccess **N**ode (**AN)** or Remote Node or First Node or Access Node or Aggregation Node). We assume here that there is only one path towards the content source, the optimal one). We symbolise the state of the **AN** as $\mathbf{S_1}$.

The probability of reaching a state of the random walk depends on the number of loops that have been taken place from the $\mathbf{S_N}$ towards the $\mathbf{S_1}$. One loop means going one state back and then one ahead.

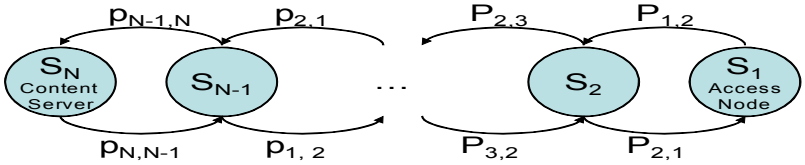

Figure 1: Router's cache states described by random walk model

The following expression is given without mathematical proof due to the limited space of the paper.

$$P\left(\begin{array}{c} \textit{to go} \text{ to the state } S_1, \text{ with all possible ways,} \\ \text{namely the sum of all possible loop compinations} \end{array} \right) = \sum_{m=0}^{\infty} P(\textit{to} \text{ go to the state } S_1 \text{ after m loops}) =$$

$$\sum_{m=0}^{\infty} \left(\begin{array}{c} P_{N,N-1} \cdot P_{N-1,N} \cdot P_{N,N-1} \cdot P_{N-1,N-2} \cdots \cdot P_{2,1} \cdot \\ \sum_{i_1=2}^{N-1} \sum_{i_2=2}^{N-1} \cdots \sum_{i_m=2}^{N-1} P_{i_1,i_1+1} \cdot P_{i_1+1,i_1} \cdot P_{i_2,i_2+1} \cdot P_{i_2+1,i_2} \cdots \cdot P_{i_m,i_m+1} \cdot P_{i_m+1,i_m} \end{array} \right) \quad (3.1)$$

where the probability $P_{i,i+1}$ is expressed assuming Poisson arrivals. In this case, we should define a time space at which the packet requests are considered. This time is defined as the average time that the cache memory becomes full if we consider it empty initially. This time is $\frac{N}{\beta}$. Thus, the probability of changing state, $P_{n,n+1}$ is the probability that the request arrivals to the state *n+1* for the time space $\frac{N}{\beta}$ are less than the ones arriving at

the state *n,* assuming in both cases Poisson arrivals. We have supposed that the arrivals at the state *n+1* are independent from the arrivals at the state *n*. Given that, $P_{n,n+1}$ is given by:

$$P_{n+1,n} = 1 - P_{n,n+1} = P\left(\begin{array}{c} k_n > k_{n+1},\ \text{where } k_n \text{ and } k_{n+1} \text{ are the requests for the packet } i \\ \text{at the states } n \text{ and } n+1 \text{ respectively for } \dfrac{N}{\beta} \text{ time space} \end{array} \right) =$$

$$\sum_{k_n=1}^{\infty\ (\text{or the number of request arrivals of packet } i)} \ \sum_{k_{n+1},\, k_{n+1}<k_n}^{k_n-1} \left(\frac{\left(\prod_{j=1}^{n} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)^{k_n} \cdot e^{-\left(\prod_{j=1}^{n} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)}}{(k_n)!} \cdot \frac{\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)^{k_{n+1}} \cdot e^{-\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)}}{(k_{n+1})!} \right) = \tag{3.2}$$

$$\left[1 - P(k_n = 0)\right] \cdot \sum_{k_n=1}^{\infty} \ \sum_{k_{n+1},\, k_{n+1}<k_n}^{k_n-1} \frac{\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)^{k_{n+1}} \cdot e^{-\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)}}{(k_{n+1})!} =$$

$$\left(1 - e^{-\left(\prod_{j=1}^{n} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)} \right) \left(1 - \sum_{k_{n=1}}^{\infty} \ \sum_{k_{n+1},\, k_{n+1}<k_n}^{k_n-1} \frac{\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)^{k_{n+1}} \cdot e^{-\left(\prod_{j=1}^{n+1} (1-r_j)\lambda_i \cdot \frac{N}{\beta} \right)}}{(k_{n+1})!} \right)$$

4. Conclusions.

We have derived an analytical model that determines the average total end to end delay in the framework of Content-Centric Networks. The proposed model can be used as an implicit QoS mechanism since through the Random Walk model, we can specify the probability of the requested content being cached at a specific node (State), while we can derive the average end to end latency related with that State. Namely we can derive analytical results of a specific average total end to end delay with specific probability.

Thus, the probability of having a specific network delay *ATQD(n), 1<n<N* for the packet of interest *i* is given by the expression (3.1).

Furthermore, we can provide an upper bound of the *ATQD* that is valid for long periods of time (e.g. for one year).

References.

[1] Van Jacobson, D. K. Smetters, James D. Thornton, Michael Plass, Nick Briggs, Rebecca L. Braynard, "Networking Named Content", Palo Alto Research Center (PARC), Palo Alto, CA, USA, 2009.

[2] Elisha J. Rosensweig, Jim Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks", Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, 2009.

[3] Jaeyeon Jung, Arthur W. Berger and Hari Balakrishnan, "Modeling TTL-based Internet Caches", MIT Laboratory for Computer Science, Cambridge, USA, 2003.

[4] James F. Kurose, Keith W. Ross, "Computer networking : a top-down approach", 5th edition, Pearson, Boston, 2010.

[5] Gross, D., J. Shortle, J. Thompson, C. Harris, "Fundamentals of Queueing Theory", 4th edition. Wiley, Hoboken, 2008.