Analysis of content size based routing schemes in hybrid satellite / terrestrial networks

[1]Élie BOUTTIER,[2]Riadh DHAOU,[3]Fabrice ARNAL,[3]Cédric BAUDOIN,[4]Emmanuel DUBOIS,[2]André-Luc BEYLOT

[1]elie.bouttier@enseeiht.fr, TéSA, Université de Toulouse; F-31071 Toulouse, France
[2]{dhaou,beylot}@enseeiht.fr, IRIT, Université de Toulouse; F-31071 Toulouse, France
[3]{fabrice.arnal,cedric.baudoin}@thalesaleniaspace.com, Thales Alenia Space
[4]emmanuel.dubois@cnes.fr, Centre National d'Études Spatiales

***Abstract*—Satellite networks are easy-to-deploy solutions to connect rural un-served and underserved areas. But satellite latency has a significant negative impact on performance. Hybrid networks, combining high-throughput long-delay links (e.g. GEO satellites) and short-delay low-throughput links (e.g. poor ADSL), can improve user experience by the use of intelligent routing. Emerging solutions, such as MultiPath TCP (MPTCP), already optimize the throughput in these hybrid networks. However, this kind of solutions does not take into account QoE requirements by the lack of relevant flows information, leading to sub-optimal path selection.**

This paper proposes an architecture able to retrieve the content size through interconnection with Content Delivery Networks (CDNs). Then, we conduct an analytical study of a probabilistic and a size threshold based routing schemes with the Mean Value Analysis (MVA) method. This shows the great benefit brought by size information in terms of QoE. To solve the limitations due to the threshold configuration, we propose a third algorithm that takes into account the path delay and capacity. Finally, we develop a testbed in order to validate our model and to compare this third scheme to the previous ones. We obtain results equivalent to the size threshold scheme, without its disadvantages.

***Index Terms*—heterogeneous multipath networks, CDN interconnections**

I. INTRODUCTION

As Internet usage is growing, there is a need for new access network technologies able to handle huge amount of traffic (FTTH, ...). Although deploying these new networks is quickly profitable in urban areas, the cost is too high to serve rural areas. Consequently, these areas stay un-served or underserved regarding these new needs.

Satellite networks are high bandwidth solutions, easy to deploy and at a reduced cost. Thereby, they are increasingly requested to connect these un-served and underserved areas. However, satellite networks also have a very large propagation delay. This has bad consequences on users experience, particularly for interactive services. Almost all web traffic uses TCP, a protocol highly affected by round-trip time (RTT) due to the use of an handshake and the slow-start algorithm. Encrypted traffic use the Transport Layer Security (TLS) protocol, which also adds a handshake. All these observations show how much the user experience is constrained by networks delays.

Nonetheless, underserved areas are connected to low throughput networks but with a short delay like long-line ADSL. Alone, these networks are not suitable for new bandwidth requirements, but it is interesting to combine them with a satellite connection in order to improve the end-user experience by taking the best of both of them.

The heterogeneous context introduces a lot of performance issues for multipath technologies. Several mechanisms have been proposed to overcome these limitations. At network layer, many proposals focus on TCP performance limitations due to packet reordering caused by the large difference between path delays. For example, the Earliest Delivery Path First (EDPF) [1] scheduler estimates packet arrival time in order to ensure correct delivery order. At transport level, the MPTCP [2] protocol allows to efficiently aggregate throughput over paths with different capacities thanks to the use of several congestion windows. Several application level solutions have been proposed for video delivery scenarios such as [3], often based on intelligent chunks retrieval.

However, these optimizations are limited by the use of restricted information gathered from the layer at which they operate. Other approaches such as [4] [5] propose to improve MPTCP by modifying the routing algorithm in order to take QoE requirements into account. [4] proposes to introduce size-aware routing, which can significantly improve the experienced latency in the context of heterogeneous networks, but the objects size is determined on the fly leading to inappropriate routing decisions and the parameterization is arbitrary.

In this paper, we propose to benefit from information provided by Content Delivery Networks (CDNs) infrastructures, which are key components of Internet nowadays, to improve path selection algorithms. We present an hybrid architecture able to retrieve objects size thanks to CDN Interconnection (CDNI). Then, we study three routing algorithms and show with an analytical model how the objects size knowledge can improve the user experience. Finally, we developed and implemented a testbed in order to validate our model and compare the performance of each routing scheme. Our results show the great improvement brought by size-aware routing.

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

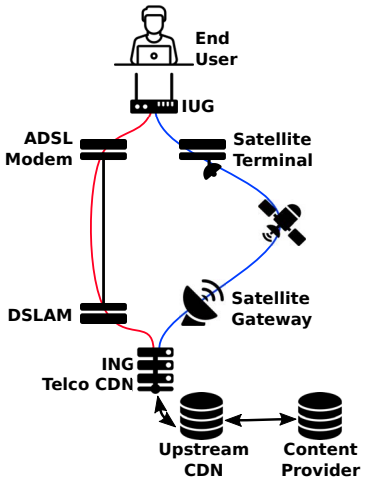

Fig. 1. Intelligent routing hybrid architecture.

II. Intelligent routing architecture

A. Hybrid architecture

In the hybrid architecture represented on figure 1, the end-user is not directly connected to any of terrestrial or satellite networks but instead is connected to a virtual ISP through a proxy called Intelligent User Gateway (IUG). The IUG is itself virtually connected through both access networks to its counterpart in virtual ISP networks, the Intelligent Network Gateway (ING).

By splitting the end-to-end connections, the IUG and ING can use any transport protocol on each path, such as TCP or SCTP [6]. Thus, the Intelligent Gateways operate in a transparent way for the end-user and the destination, while the data can be spread over the multiple underlaying paths.

B. CDN Interconnection (CDNI)

We propose to retrieve content informations, and especially content size, from Content Delivery Networks (CDNs). Indeed, CDNs know a lot of information about hosted contents such as file size but also MIME type, video bitrate, etc. On the other hand, more and more contents are subject to caching nowadays, increasing the scope of this approach. To retrieve these information, we propose to use CDN Interconnection (CDNI) interfaces.

CDNs have been introduced as a solution to solve new delivery issues due to the increasing bandwidth usage [7]. They are composed of several cache servers called edge servers deployed in strategic places. Content requests from end-users are redirected to the closest edge server, reducing experienced delay, network usage and the load of origin content servers. Content providers delegate content delivery to CDN operators through service agreements.

Edge servers are usually located at POPs in the Internet, allowing CDN operators to peer directly with Internet Service Providers (ISP) avoiding transit delay and cost. However, POPs are generally still far away from end users. There is a trend to get edge servers closer to the user by placing them in access network. This is sometime done by CDN operators providing cache servers to ISP to add them to their network. However, this approach is limited: these cache servers only enable to serve content from one CDN operator. Furthermore, they are still managed by CDN operators whereas ISPs have a better knowledge of their network and possible optimizations.

For these reasons, ISPs are interested in deploying their own CDN inside their network (Telco CDN) in order to directly handle content requests from their users. This is formalized through a content delivery agreement between a CDN operator and a Telco CDN. Thenceforth, content requests originating from the ISP are redirected by the CDN operator to the Telco CDN.

Such an agreement comes with the need to communicate several pieces of information between both CDNs: source IP address of content requests to redirect, redirection end-point, authentication, bandwidth limitations, ... The CDNI Working Group at IETF is currently working on standardized interfaces to allow CDNs to communicate such information and encourage democratization of CDN interconnections. Four JSON over HTTP interfaces have been defined including among others the Request Routing interface (RR) and the Metadata Interface (MI) [8]. Thereby, the information that we need for routing can be easily recovered through the MI interface.

In our case, we suppose the virtual ISP operates his own CDN interconnected with several upstream CDNs.

III. Algorithms

We propose to study three different routing algorithms. The first one is naive whereas the two others are size-aware. We aim to minimize the necessary time for a given object to be requested and fully received.

All of them are flow based, i.e. each object is forwarded entirely throught the same path. This choice avoids out-of-order delivery which causes the experienced latency being the highest of both path, thereby annihilating the best advantage of the terrestrial path.

A. Probabilistic routing scheme

This routing scheme aims to share the load between the two networks on a random basis. Flows are routed through the terrestrial network with a probability p_{ter} and through the satellite network with a probability p_{sat}. In this analysis, we choose routing probabilities equal to the ratio of the paths' capacity c_i:

$$p_{\text{ter}} = \frac{c_{\text{ter}}}{c_{\text{ter}} + c_{\text{sat}}} \qquad p_{\text{sat}} = \frac{c_{\text{ter}}}{c_{\text{ter}} + c_{\text{sat}}} \tag{1}$$

Note that object sizes are not used in this routing decision algorithm. Under heavy load, these probablities should achieve best performance but we do not expect especially good result in general from this method. We use it as a reference in order to show the advantages of size-aware routing schemes and to validate our model and implementation.

B. Size threshold based routing scheme

This second routing scheme is based on the assumption that long-objects benefit mostly from high-bandwidth paths, reducing download time and unloading low-delay paths. Furthermore, short objects are on average more delay-sensitive (e.g. HTML content or AJAX requests) than long objects (e.g. videos). Thus, we consider a fixed threshold; flows whose object size is smaller than the threshold are routed through the terrestrial network otherwise they are routed through the satellite network.

C. Minimum latency routing scheme

This scheme aims to minimize the overall experienced latency. Upon each new flow arrival, we compute for each path i the expected latency e_i with the following formula:

$$e_i = d_i + (n_i + 1) \cdot \frac{s}{c_i} \tag{2}$$

where

- d_i is the delay of the path i;
- $n_i + 1$ is the current flow count assigned to the path i plus the potentially newly assigned flow;
- s is the size of the considered object;
- c_i is the capacity of the path i.

This estimation supposes the path capacity is fairly shared among the n_i flows affected to the considered path. In fact, the latency is overestimated as some flows will complete before the newly affected one, increasing available per-flow throughput. However, future flows must also be taken into account. Therby, we assume ending flows are compensated by the new arrivals, so an approximate prediction could be to consider n_i as constant during the flow lifetime.

The simple fact to affect one flow to a path has the side effect to deteriorate the performance of the flows already affected to the same path. We take this in account by adding to the expected latency the additionnal delay sustained by existing flows, and the new flow is routed through the path p that is minimizing this quantity:

$$p = \underset{i}{\operatorname{argmin}} \left\{ d_i + (n_i + 1) \cdot \frac{s}{c_i} + \sum_{j=0}^{n_i} \frac{\min(r_j, s)}{c_i} \right\} \tag{3}$$

with r_j the remaining bytes of flow j to be transmitted, the flow j being affected to path i.

This method is interesting because it only needs the delay and capacity of the considered paths instead of an arbitrary threshold. Furthermore, it could be applied to a network with three or more paths without any changes.

IV. Modeling

We created a model to assess probabilistic and content size threshold routing strategies. It uses queuing network and Mean-Value Analysis to compute theoretical average time to retrieve an object through the hybrid network.

A. Benchmarking scenario

In order to evaluate the performance of the proposed routing schemes, we consider a hybrid network with two paths, such as in figure 1. The activity of a home network is modeled by N processes, each one being idle during a random time, waking-up, sending a request for a specific object and then going back to its idle state after the reception of the full content. As a single end user can make multiple concurrent connections, N do not really model the number of users but rather the overall network activity.

The measured performance metric is the Flow Completion Time (FCT), defined as the elapsed time between the emission of the query and the reception of the last part of the requested object.

B. Queueing network modeling

Routing strategies operate in a closed network of six queues as shown on figure 2. They model user idle time, request sending time and content receiving time.

- The A queue models the user idle times. It has an infinite number of servers (delay queue) – each user directly starts to think about their next request when arriving in this queue – and an exponential service law.
- The B queue is also a delay queue, modeling the content request propagation times – requests being smalls, the transmission times is neglected. As requests are always routed through the terrestrial network, the service time is constant and equal to the terrestrial network delay.

We neglect the time needed by the ING to acquire the content from the upstream server. Consequently, the next step is the routing process in order to send the content on the appropriate network.

Remaining queues model content transmission and propagation times. C_{ter} and D_{ter} queues stand for the terrestrial network whereas C_{sat} and D_{sat} queues stand for the satellite network.

- C_i queues model flows transmission time. The path being fairly shared between every flow affected to a specific network, we model it with a unique server under Processor Sharing (PS) service discipline.
  Service time is proportionnal to the average flow size and inversely proportionnal to the path capacity. While path

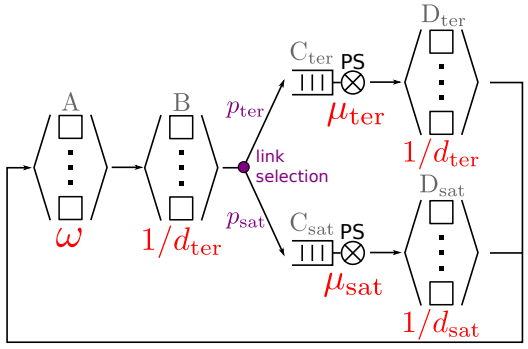

Fig. 2. Queueing network diagram.

capacity is fixed, note that perceived flow size distribution depends on the routing strategy in case of size-based routing scheme.

- D_i queues model content propagation time from ING to IUG. As B, they are also delay queues with a constant service law, respectively equals to the terrestrial and satellite network delay.

C. Mean-Value Analysis (MVA)

The mean-value algorithm presented in [9] allows to compute average response time, queue length and throughput at each queue in a closed queueing network. It is based on the arrival theorem and the Little's law and proceeds recursively on the number of clients in the network. This method is less powerful than the BCMP [10] theorem as it does not give queue length distribution. Nevertheless, only average service time is needed and it applies more generally.

The algorithm starts with an empty system, queue lengths being initialized to zero. At each iteration, the number of clients m is incremented until the required number of clients is reached and the following steps are computed:

1) The average response time $R_k(m)$ at each queue k as average queuing time in a system with $m-1$ clients plus average service time of the m-th client:
$$R_k(m) = \frac{L_k(m-1)+1}{\mu_k} \tag{4}$$
with $L_k(m-1)$ the average length of the k-queue.
In the case of delay queue, there is no queuing time so average response time is directly equal to $1/\mu_k$.
2) The system throughput using Little's law:
$$\lambda(m) = \frac{m}{\sum_{k=1}^{K} R_k(m) v_k} \tag{5}$$
with v_k being the visit ratio of the k-queue, i.e. the ratio of users going through the k-queue during a loop in the system.
3) The mean queue length $L_k(m)$ at each queue using Little's law:
$$L_k(m) = v_k \lambda(m) R_k(m) \tag{6}$$

D. Parameters

1) Service rate: The A queue service rate is equal to the parameter ω of the exponential law followed by the thinking time. For the B and D_i delay queues, the service rate is trivially $1/d_i$. C_i queues service rate μ_i is equal to $\overline{s}/c_i$. The average flow size $\overline{s}$ directly depends on the routing scheme, especially with size-based strategies, and will be studied specifically later.

2) Visit ratio: All clients passing through A and B queues, the visit ratios are 1 for these queues. C_{ter} and D_{ter} queues visit ratios are equals to the probability of routing through the terrestrial path p_{ter} whereas C_{sat} and D_{sat} queues visit ratios are equal to its counterpart p_{sat}. These probabilities also depends on the routing strategy.

It should be noted that these probabilities must be independant of queue states. Thereby, our model does not apply to the last routing algorithm, which is based on traffic condition.

V. Testbed architecture

We implemented a testbed in order to benchmark previously presented routing schemes, in particular the third one which can not be modeled. The testbed emulates the network activity generated by a home network, connected to the Internet through a virtual hybrid-enabled ISP.

As we focus on the path-selection, our testbed does not use any congestion control algorithm in order to avoid side effects and obtain results easier to interpret. We also assume a perfect knowledge of the delay and the capacity of underlaying networks, although in reality these information must be estimated, using for exemple management information (e.g. MIB) or TCP-stack variables (RTT, cwnd, ...).

A. Architecture

The testbed includes four components: a content server, two proxies (corresponding to the ING and the IUG) and a test client. All components run on the same physical machine but in two different Linux network namespaces, the test client and IUG in a first one and the ING and content server in the second one. The two namespaces are interconnected by two virtual ethernet interface pairs, corresponding to the terrestrial and satellite networks. The constant delay d_{ter} and d_{sat} of these networks are emulated with Linux `netem`. It should be noticed that there is no bandwidth limitation at the link level: this is emulated by software. This ensures that there are no wire-losses as we did not implement error detection and retransmission algorithms.

The test client communicates with the IUG using TCP, as the ING and the content server. However, the IUG and ING use a simple prototyped UDP-based protocol to communicate over both path.

B. Test client

The test client is composed of N independent threads. Each thread sleeps during a random time, then wakes-up, connects to the IUG and sends a request for a specific object. Threads return to their sleeping state after the reception of the full content from IUG and the elapsed time is recorded.

C. Intelligent Gateways

The IUG is a SOCKS5 proxy listening for connection requests from the test client. On connection demand, the IUG forwards it through the fastest path to the ING on high-priority QoS queues, the latter establishing a connection to the content server. From there, the test client and the content server can communicate through the Intelligent Gateways.

The UDP datagrams exchanged by the IUG and ING contain a header with a connection ID and a sequence number. These informations are used to correctly reconstruct corresponding streams, datagrams being reordered if needed.

Links are shared between flows according to a fair queueing scheduler, ensuring the same amount of bandwidth to every flows. It is a slightly modified version of Linux FQ scheduler in order to take flows instead of packets as input as we operate at the application layer. FQ attempts to emulate the

fairness of bitwise round-robin which is equivalent to the Processor Sharing (PS) scheduler used in the model. Thereby, FQ is asymptotically equivalent to PS, so the model and the testbed should give identical results . In addition, the use of a bandwidth-fair scheduler allows to distinguish path-selection and scheduling effects, scheduling having an important impact on performance [11].

VI. Results

A. Experimental parameters

For our experiments, we consider two heterogeneous paths. The first one has a delay of 25 ms and a download capacity of 1 Mbps whereas the second one has a delay of 350 ms and a download capacity of 5 Mbps. Upload capacities are unlimited. The UDP datagrams exchanged by the ING and the IUG carry a payload of 1450 bytes, with a headers of 50 bytes (due to the sum of underlaying headers). The clients idle time follows an exponential law with an average value of one second. We also suppose that the objects' size follows an exponential law with an average size of 11600 bytes (equivalent to 8 UDP messages). This assumption facilitates the calculation of the average flow sizes (including headers) but the model still applies with a general law.

B. Probabilistic routing scheme

As we choose routing probabilities equal to the ratio of path capacities, we obtain $p_{\text{ter}} = \frac{1}{6}$ and $p_{\text{sat}} = \frac{5}{6}$.

For the analytical results, we need to know the average flows size in order to compute the service rates of C_{ter} and C_{sat} queues. With this routing strategy, they are equals on both networks. Although we previously supposed object sizes to follow an exponential law, flow sizes are longer due to the headers introduced by the packetization process. However, the average flow sizes $\overline{s}$ can still be computed:

$$\overline{s} = \sum_{i=1}^{\infty} \int_{(i-1)\cdot p}^{i\cdot p} (i \cdot h + x) \cdot e^{-\frac{x}{s}} \mathrm{d}x \tag{7}$$

with s the average object sizes, h the header size and p the size of the message payloads.

The integral has a long but formal expression, and the sum can be numerically computed with a very small shift after few iterations. Finally, service rates are, for both queues, equal to $\mu_i = c_i/\overline{s}$.

Results obtained from both analytical analysis and experiments are represented on figure 3. Experienced delay is compared for a varying number of clients in the system. Results are very close, validating the model and the testbed implementation.

C. Object size threshold routing scheme

Routing probabilities can be computed from objects size cumulative distribution function (CDF). As objects size follow an exponential law, these probabilities have a very simple expression:

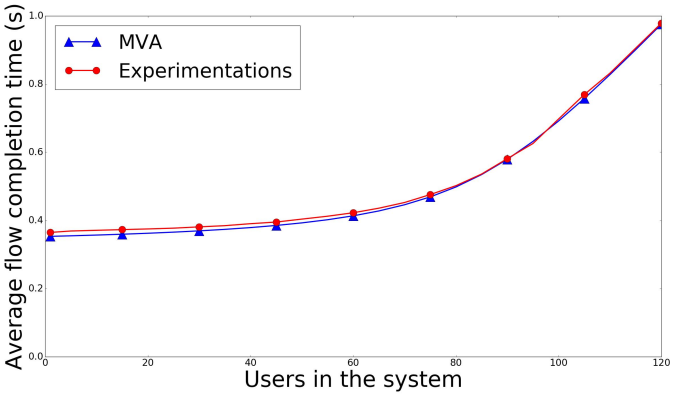

Fig. 3. Comparison of theoretical and analytical results for a probabilistic routing strategy.

$$p_{\text{ter}} = CDF_s(t) = 1 - e^{-\frac{t}{s}} \qquad p_{\text{sat}} = 1 - CDF_s(t) = e^{-\frac{t}{s}}$$

Like before, we need average flow sizes on each path to set our model parameters. In contrary to the previous routing strategy, these averages are different for each considered path. First, we determine the object size distribution on each path. This is done by normalising exponential distribution probability density function on, t being the threshold, $[0;t]$ for terrestrial network and $]t;\infty[$ for satellite network. Thus, we obtain:

$$s_{\text{ter}}(x) = \frac{\lambda e^{-\lambda x}}{1 - e^{-\lambda t}} \qquad s_{\text{sat}}(x) = \lambda e^{-\lambda(x-t)} \tag{8}$$

As previously, average flow sizes should take headers into account:

$$\begin{aligned} \overline{s_{\text{ter}}} &= \sum_{i=1}^{\lceil t/p \rceil} \int_{(i-1)\cdot p}^{\min(i\cdot p, t)} (i \cdot h + x) \cdot s_{\text{ter}}(x) \mathrm{d}x \\ \overline{s_{\text{sat}}} &= \sum_{i=\lceil t/p \rceil}^{\infty} \int_{max((i-1)\cdot p, t)}^{i\cdot p} (i \cdot h + x) \cdot s_{\text{sat}}(x) \mathrm{d}x \end{aligned} \tag{9}$$

Figure 4 represents size threshold routing scheme results for a constant number of clients (40) in the system but varying the threshold. On the left side of the graph, the threshold is very small causing the majority of flows to be routed through the satellite network. Contrariwise, on the right side the threshold is more important, causing flows to be routed through the terrestrial network. In this right part, as the threshold decreases, average Flow Completion Time decreases due to short requests being routed through the low-delay network. However, when the threshold becomes too high, there is a quick increase of average FCT due to congestion on the terrestrial network.

We can determine from this concave curve the best threshold for a given set of load parameters (number of clients in the system, object size, arrival law, ...). Figure 5 shows a comparison of theoretical and analytical results for a threshold value of 9298 bytes which is the threshold achieving the best

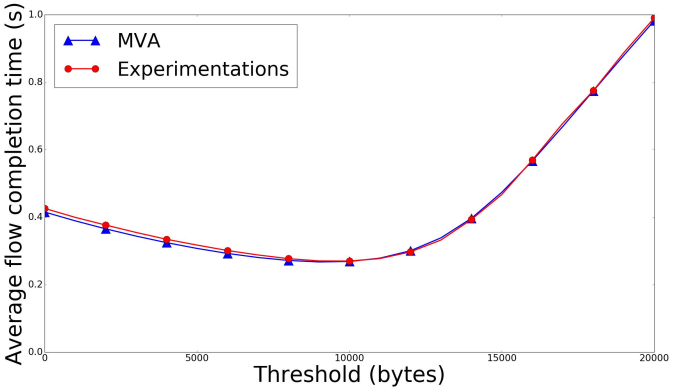

Fig. 4. Comparison of theoretical and analytical results for the size threshold routing scheme varying threshold value, with a fixed number of 40 clients.

performance for a system with 40 clients according to MVA. However, this value gives bad performance for a higher or a lower number of clients in the system, showing the importance of this parameter. We also represent on the same figure the theoretical and experimental performance valued with the best threshold for each value of N. These optimal thresholds are computed from theoretical results.

D. Minimum latency routing scheme

Figure 6 compares experimental results of the three proposed routing algorithms. We observe that the minimum latency routing scheme provides results very close to those obtained with the best size threshold one. These results have the advantage to be obtained without the need of any parameters from traffic paterns.

Conclusion

In this paper, we propose an hybrid architecture allowing efficient routing decision from both throughput and QoE point of view in a heterogeneous environment. Contrary to other approaches such as current MPTCP algorithms where the main objective is to increase the aggregated throughput, our approach also considers user requirements thanks to the CDNI standardized interface that enables to retrieve flows information. We demonstrate through both theoretical studies

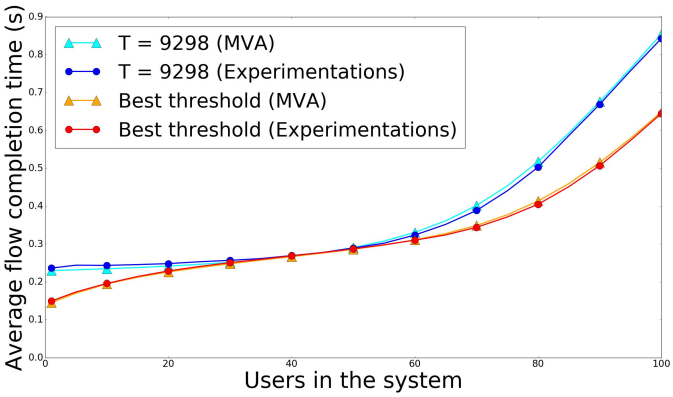

Fig. 5. Comparison of theoretical and analytical results for the size threshold routing scheme varying number of clients.

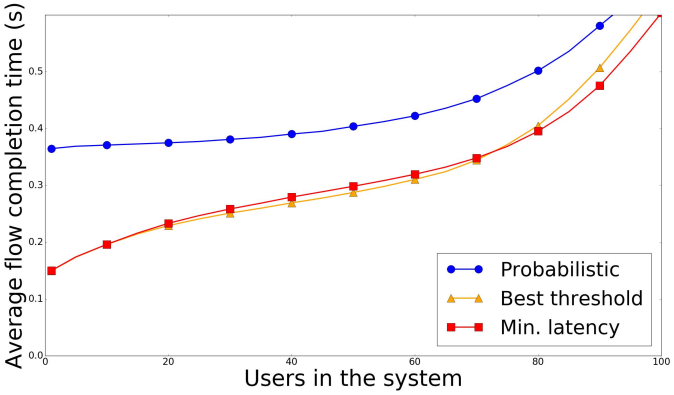

Fig. 6. Performance comparison between the three routing algorithms.

and experimentations that the knowledge of the content size can improve experienced latency. We proposed an algorithm achieving close to optimal performance (from a MVA perspective). In addition, this method is adaptive and does not require to configure a size threshold.

As these first results demonstrate the improvement brought by size-aware routing scheme, we plan to improve the proposed algorithm to take benefit from other CDN information (e.g. type of content, codecs ...). In addition, we will refine the QoE benefits evaluation with other metrics such as jitter, packet error rate or application based performance criteria. We finally plan to test our proposed solution in a more realistic environment. This implies notably to introduce real transport protocols such as QUIC or TCP-PEP between the Intelligent Gateways.

References

[1] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 388–403, April 2006.

[2] C. Paasch, S. Barre *et al.*, "Multipath tcp in the linux kernel." [Online]. Available: http://www.multipath-tcp.org

[3] P. Sharma, S. j. Lee, J. Brassil, and K. G. Shin, "Aggregating bandwidth for multihomed mobile collaborative communities," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 280–296, March 2007.

[4] "D3.3.2 multi access networking architecture," BATS Project, 2014. [Online]. Available: http://batsproject.eu

[5] X. Corbillon, R. Paricio-Pardo, N. Kuhn, G. Texier, and G. Simon, "Cross-Layer Scheduler for Video Streaming over MPTCP," in *ACM Multimedia Systems 2016 Conference*, ser. MMSys 2016. ACM, 2016. [Online]. Available: http://dash.ipv6.enstb.fr/mmsys2016/

[6] R. Stewart, "Stream Control Transmission Protocol," RFC 4960 (Proposed Standard), Internet Engineering Task Force, Sep. 2007.

[7] G. Peng, "Cdn: Content distribution network," 2004.

[8] B. Niven-Jenkins, R. Murray, M. Caulfield, and K. J. Ma, "CDN Interconnection Metadata," Internet Engineering Task Force, Internet-Draft draft-ietf-cdni-metadata-13, Mar. 2016, work in Progress.

[9] M. Reiser and S. Lavenberg, "Mean-value analysis of closed multichain queuing networks," *J. ACM*, vol. 27, no. 2, pp. 313–322, Apr. 1980.

[10] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, Apr. 1975.

[11] D. Lu, H. Sheng, and P. Dinda, "Size-based scheduling policies with inaccurate scheduling information," in *Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings. The IEEE Computer Society's 12th Annual International Symposium on*, Oct 2004, pp. 31–38.