Improving content delivery with size-aware routing in hybrid satellite / terrestrial networks

Elie Bouttier, Riadh Dhaou, Fabrice Arnal, Cédric Baudoin, Emmanuel Dubois, André-Luc Beylot

Improving content delivery with size-aware routing 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*—As Internet usages expand quickly, access networks are modernized with new technologies like fiber-optic communications. However, upgrade costs are prohibitive in sparsely populated areas, the latter turning notably towards satellite connection. Indeed, this technology allows deploying a high-throughput Internet access quickly in these regions. Nevertheless, GEO satellites induce a long delay, not experienced on terrestrial infrastructures despite their low throughput.**

In this paper, we consider a heterogeneous network with both a satellite and a terrestrial path. This kind of architecture is known to be difficult to operate because of the important differences between used technologies. The emerging Multipath TCP (MP-TCP) transport protocol, whose design enables to aggregate disparate paths properly, brought new hopes for heterogeneous networks. However, it does not take user Quality of Experience (QoE) into account as it focuses on maximizing the links occupancy.

This paper proposes an intelligent path selector using the content size to maximize users QoE in heterogeneous networks. Before detailing this method, we describe the architecture able to retrieve the size of delivered contents thanks to Content Delivery Network Interconnection (CDNI). Finally, we implement a testbed to evaluate the behavior of the proposed routing method. The results show a significant improvement of the delivery performance, outperforming MP-TCP.

I. INTRODUCTION

The massive increase of Internet usage leads to a rise of networks requirements. Legacy Asymmetric Digital Subscriber Line (ADSL) access networks are not able anymore to adequately satisfy users needs. Therefore, they are gradually upgraded mainly in favor of fiber-optic communications. However, for economic reasons, operators target densely populated areas while the countryside stays poorly served.

As a result, many users in these regions turn towards satellite Internet access which offers a high bandwidth broadband access without the need for very long and expensive works. However, the propagation delay to the geostationary orbit has a substantial negative impact on the user experience. Indeed, many services are highly sensitive to the delay, especially interactive services like VoIP or video games but also web browsing which requires numerous connections to retrieve a large number of contents. Moreover, HTTP relies on the Transmission Control Protocol (TCP) which is constrained by the delay, notably due to the initial handshake and the slow-start period. The common use of the Transport Layer Security (TLS) protocol adds another extra handshake.

From these observations, the short delay of long lines ADSL connections appears as a notorious advantage despite the limited throughput. Then, it becomes interesting to combine both networks with an intelligent routing to make use of the special benefit – capacity or short delay – of each network.

Multipath networks have been subject to numerous studies. However, heterogeneous systems, combining paths with different characteristics, bring some particular issues. Hybridization techniques mostly operate at layer 2 for local networks and layer 3 for global networks. But the widespread TCP protocol operates poorly in the presence of packet reordering, introduced by the difference of delay between the various disparate paths. Therefore, aggregation at these levels is excluded and load-balancing is limited to the flow decision scale.

Multiple load-balancing algorithms exist, such as classic Round-Robin (RR) or hash-based schedulers [1], and their implementation have been standardized through several protocols such as Link Aggregation Control Protocol (LACP), Shortest Path Bridging (SPB) or Enhanced Interior Gateway Routing Protocol (EIGRP). The new transport protocol MP-TCP [2] stands out and can aggregate heterogeneous paths by re-ordering the data at the receiver. The protocol knew significant improvement with the minRTT scheduler and the Opportunistic Linked-Increases Algorithm (OLIA) for congestion control which is inspired by [3] and brings pareto optimality. It is still evolving with new propositions such as the Blocking Estimation-based (BLEST) scheduler [4] or Weighted Vegas (wVegas) [5] and Balanced Linked Adaptation (BALIA) [6] congestion control algorithms.

However, neither routing algorithms nor MP-TCP takes into account application-level flow information as they focus on network metrics and resource utilization. Though, they can be used to improve the end-user experience significantly as proposed in the BATS Project [7] or sketch in [8] in a simplified unrealistic case using the UDP transport protocol.

In this paper, we propose a multipath routing algorithm, MinFCT, to enhance the performance of hybrid satellite / terrestrial networks. Thanks to CDNI, requested contents size

can be retrieved. This information is then used by MinFCT to estimate the Flow Completion Time (FCT) and take the best possible routing decision, achieving an outstanding performance increase.

We start the paper by describing the proposed hybrid architecture. In part II, the MinFCT routing strategy and its Flow Completion Time estimation method are described. Afterward, we present our implemented experimental testbed and then the obtained results. Finally, we conclude on the large benefit of the MinFCT algorithm.

II. Architecture

This section describes a multipath architecture enabling a home network to access the Internet through a terrestrial and a satellite path.

A. Hybrid architecture

In the hybrid architecture represented in figure 1, the end-user is not directly connected to any of terrestrial or satellite networks but instead is attached to a virtual Internet Service Provider (ISP) through a proxy called Intelligent User Gateway (IUG) reusing the terminology introduced in BATS [7]. 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 [9]. 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 underlying paths.

B. CDN Interconnection (CDNI)

We propose to retrieve content information, 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 content 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 this information, we propose to use CDN Interconnection (CDNI) interfaces.

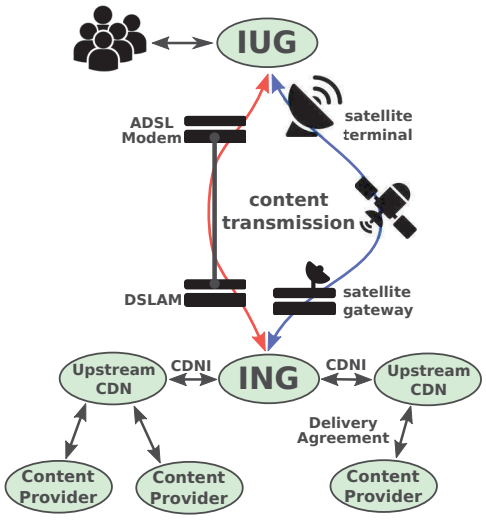

Fig. 1: Hybrid architecture

CDNs have been introduced as a solution to solve new delivery issues due to the increase in bandwidth usage [10]. 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 the experienced delay, the network usage, and the load of the origin content servers. Content providers delegate content delivery to CDN operators through service agreements.

Edge servers are usually located at the Points-of-Presence (POPs) on the Internet, allowing CDN operators to peer directly with ISPs avoiding transit delay and cost. However, POPs are still far away from end users. There is a trend to get edge servers closer to the user by placing them in access network. Sometimes, CDN operators provide 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 CDN inside their network (Telco CDN) to directly handle content requests from their users. These deployments are formalized through content delivery agreements between CDN operators and Telco CDNs. 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 interfaces (using JSON over HTTP) have been defined including among others the Request Routing interface (RR) and the Metadata Interface (MI) [11]. Thereby, the information that we need for routing can be easily recovered through the MI interface.

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

III. The MinFCT routing strategy

The proposed routing strategy is called MinFCT which stands for Minimum Flow Completion Time. This strategy is in charge of the load balancing between the terrestrial and the satellite paths. It operates at the flow-level which means that all packets belonging to the same flow are sent on the same path. Indeed, aggregation of the same flow on different path turns out to be counterproductive, especially by increasing terrestrial congestion [12]. Therefore, the routing decision is taken only once for each content.

The routing scheme aims to minimize the Flow Completion Time (FCT). Indeed, it is a homogeneous criterion to evaluate the performance of our multipath network, relevant for both delay sensitive flows and elastic traffic. For this purpose, the

FCT is estimated on each path, the one that minimizes it being selected. The estimation method is based on the flow size and various path information: capacity, delay, loss rate and the number of flows sharing the path.

A. Estimation of path characteristics

1) Delay: The path delay is merely estimated from TCP handshakes. This value could be temporally averaged and smoothed but our experiments did not show the need for it.

2) Capacity: On the satellite path, the capacity is supposed known. Indeed, satellite systems are very integrated, with a lot of interaction between different components. For instance, Performance-Enhancing Proxies (PEPs) could be coupled with resource allocators to obtain available capacity for the given user. We supposed this kind of cooperation with our proxy.

Of course, this method can not be implemented on the terrestrial path. A capacity estimation from TCP stack information has been considered, but this information turns to be highly unreliable. Indeed, as the traffic often presents a lot of short flows, the TCP congestion control is not reached or does not have time to converge.

Instead, the capacity is estimated from TCP acknowledgment (Ack) packets passively observed on the path near the gateway. Each TCP flow is tracked, the last acknowledged segment being recorded and updated when an Ack is observed. The distance between the previous and the new values indicates the number of bytes successfully received by the terminal. Finally, the total amount of acknowledged data is averaged on a sliding temporal window.

The estimated throughput is reasonably accurate but delayed by the path delay and recovery period after losses. Indeed, when a loss occurs, the following segments are acknowledged with the sequence number of the lost segment, no longer allowing estimating received data. The SACK option introduced in RFC 2018 [13] adds thinner per-block acknowledgment and could be used to reduce the delay of the estimation introduced by losses.

3) Loss rate: The loss rate is estimated from duplicates TCP acknowledgment (DupAck) detected during the passive capacity estimation. When a DupAck is observed for a given flow, a loss is recorded, and the flow is marked in recovery. Subsequent DupAcks are ignored until a new segment is properly acknowledged, the recovery mark being removed. The loss rate is deducted from the recorded losses and the number of packets sent by the gateway to the terminal. As for capacity, the loss rate is averaged on a sliding temporal window.

This method somewhat underestimates the loss rate as multiple losses can occur in the same flows, but only one is recorded per recovery time. Precision could be improved by decoding the TCP SACK option field but at the cost of a substantial increase in the implementation complexity. Nonetheless, the estimated loss rate is quite accurate and completely sufficient for the operation of MinFCT.

4) Concurrent flows count: In order to find out the individual available capacity for a given flow, the overall path capacity is divided by the number of flows competing for a given path. The proxy easily knows the number of flows affected to a given path, but it turns out to be an overestimation of active flows. Indeed, some flows are waiting for their initial handshake to complete and others, although having no more data to send, are still waiting for the acknowledgment of in-fly data. Therefore, only flows with a completed handshake and pending data, in applicative or TCP buffers, are taken into account. On congested links, defined here by a loss rate higher than 1%, flows with unacknowledged in-flight data are also taken into account as probably retransmitting.

B. Flow Completion Time (FCT) estimation

The literature proposing TCP performance models is based on an idea first published in [14]. But they focus on the long-term behavior and rely on the loss rate which varies greatly and depends on the activity of several concurrent flows. Therefore, we propose our own FCT estimation method applicable to the MinFCT routing decision scheme.

The Flow Completion Time is estimated mainly from the content size, the path delay and the individual capacity available for the considered flow. It is computed as the sum of the times of the multiple steps of the TCP connection and other factors influencing the completion time:

- the initial handshake
- the transmission time and the propagation time
- the additional time due to the slow start
- the additional time suffered by competing flows
- the retransmission time due to losses

These different times are computed as follow:

1) Initial handshake: naturally three times the path delay.

2) Transmission time: computed with the following formula:

$$t_{\mathrm{e}} = (n_i + 1) \cdot \frac{\mathrm{s}}{c_i} \tag{1}$$

with n_i the number of competing flows on the path i, s the size of the content and c_i the overall capacity of path i.

3) Propagation time: once the path delay.

4) Slow start: Especially on paths with a large bandwidth-delay product, for instance satellite paths, the TCP connection can stay in the slow-start phase for a long time before reaching the congestion control. The slow-start phase is characterized by several round-trip times (RTTs) that increase the FCT by that much. To take the additional time into account, we compute the maximal congestion window (cwnd max) before entering the congestion control and estimate the number m of required RTTs to reach it. The cwnd max, expressed in segments, is proportional to the per-flow bandwidth-delay product:

$$\mathrm{cwnd}_{\max} = \frac{2d_i}{\mathrm{mss}} \cdot \frac{c_i}{n_i + 1} \tag{2}$$

with usually a maximum segment size (MSS) of 1448 bytes per segment. The number of required RTTs is thus:

$$m_{\mathrm{RTT}}(\mathrm{cwnd}_{\max}) = \mathrm{ceil}\left\{\log_2\left(\max\left(1, \frac{\mathrm{cwnd}_{\max}}{\mathrm{initcwnd}}\right)\right)\right\} \quad (3)$$

with an initial congestion window (initcwnd) of 10 segments as defined in RFC 6928 [15]. However, the maximum cwnd is not necessarily reached for short flows. The maximum number of RTT regarding the flow length is:

$$m_{\mathrm{RTT}}(\mathrm{end}) = \mathrm{floor}\left\{\log_2\left(1 + \frac{s/\mathrm{mss}}{\mathrm{initcwnd}}\right)\right\} \quad (4)$$

Finally, the additional time introduced by the slow start is:

$$t_{\mathrm{ss}} = 2 \cdot d_i \cdot \min\left\{m_{\mathrm{RTT}}(\mathrm{cwnd}_{\max}), m_{\mathrm{RTT}}(\mathrm{end})\right\} \quad (5)$$

5) Additional time suffered by competing flows: Assigning a new flow to a path results in a decrease of individual available capacity of competing flows, thus increasing their completion time. This additional time can be estimated with the following formula:

$$t_{\mathrm{comp}} = \sum_{j=1}^{n_i} \frac{\min(r_j, s)}{c_i} \quad (6)$$

with j ranging all flows affected to the same path i and r_j being the remaining amount of data waiting to be sent for the flow j.

6) Additional times due to losses: Each sent packet can be lost and must then be retransmitted, several times if necessary. The number of transmission of a packet has a expected value of $1 + \tau_i + \tau_i^2 + \ldots = \frac{1}{1-\tau_i} = \mathrm{T}$ with τ_i the loss rate of the considered path. Therefore, a flow of length s will require on average to transmit $s \cdot T$ bytes before a successful reception. Thus, all sizes , s or r_j, present in formulas (1), (4) or (6) are multiplied by the quantity T.

Besides, retransmissions add extra round-trip times, equals on average to $t_{\mathrm{l}} = 2 \cdot d_i \cdot \mathrm{T}$.

Finally, the selected path p is the one which minimizes the estimated Flow Completion Time:

$$p = \underset{i}{\mathrm{argmin}}\left\{4 \cdot d_i + t_{\mathrm{e}} + t_{\mathrm{ss}} + t_{\mathrm{comp}} + t_{\mathrm{l}}\right\} \quad (7)$$

IV. Testbed architecture

A. Architecture

The performance of MinFCT is compared to MP-TCP experimentally on a testbed schematized on figure 2. A test client requests contents from a content server through the Intelligent User Gateway using the SOCKS5 protocol. The gateways are interconnected with two links; the latency is emulated with the Network Emulator (netem) Traffic Control (tc) queueing discipline (qdisc) and the capacity with the Token Bucket Filter (tbf) qdisc.

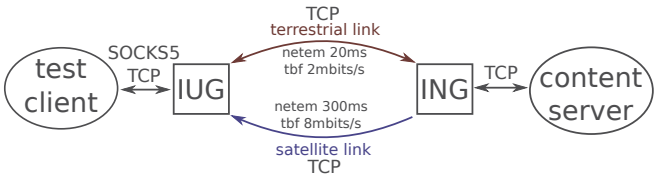

Fig. 2: Experimental testbed architecture.

B. Test client

The test client is composed of n threads imitating the activity of n users. Each thread follows a cycle of requesting an object, waiting for its reception, and finally waiting during a random period modeling the processing of the object.

C. Passive Throughput and Loss rate Reckoning (PTLR)

The throughput and loss rate estimations are done in the same tool. This standalone program runs multiple times, once per path, each instance estimating the parameters of its path.

Packets are captured with `libpcap`. Decoded events (data sending, data acknowledgment or losses) are timestamped and saved at the end of a circular resizable buffer. Periodically, e.g. every 100ms, path characteristics are recomputed and shared with the gateway proxy through shared memory. Before each recalculation, the beginning of the buffer is cleared from outdated values, older than the averaging window. For improved performance, the metrics are computed from aggregated values, like total recorded losses, which are refreshed on every buffer update.

D. Intelligent Gateways

Proxies are implemented in C with asynchronous programming using `libevent2`. The path RTT is obtained with the `tcpi_rtt` member of the Linux `tcp_info` structure which contains a lot of informations about the TCP stack internal state. The structure is retrieved with a simple `getsockopt` function call for the `TCP_INFO` option at the `SOL_TCP` level. The value is retrieved each new connection and smoothed like the SRTT in RFC 6298 [16].

V. Results

A. Experimental parameters

We consider, for our experiments, a terrestrial path characterized by a delay of 25 ms and a download capacity of 2 Mbps, and a satellite path characterized by a delay of 300 ms and a download capacity of 8 Mbps. We do not limit upload capacity as this direction is not part of our study. The `tbf` qdisc is configured with a peak rate set to 150% of the capacity, a buffer also known as burst, set to the amount of data transmitted during 10ms at the maximum capacity rate and a queueing capacity limit set to 40ms for the terrestrial path and 160ms for the satellite path.

Single-path experiments are conducted with TCP cubic between both gateways, the connections are not end-to-end and initiated by the terminal. The MP-TCP version used is `0.92.1` under Linux `4.4.83` with the OLIA congestion control algorithm (wVegas and BALIA give similar results) and the default ”lowest RTT” scheduler. The fullmesh path manager is used, with two IP addresses on the gateway but only one locally visible on the terminal. MinFCT also relies

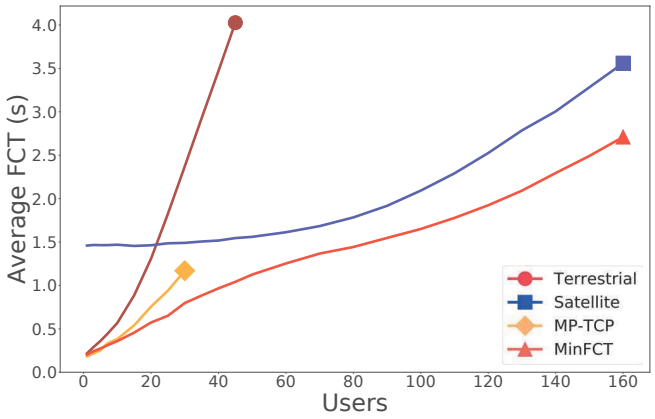

Fig. 3: Comparison of average Flow Completion Time with MinFCT, MP-TCP, TCP over the terrestrial path only and TCP over the satellite path only.

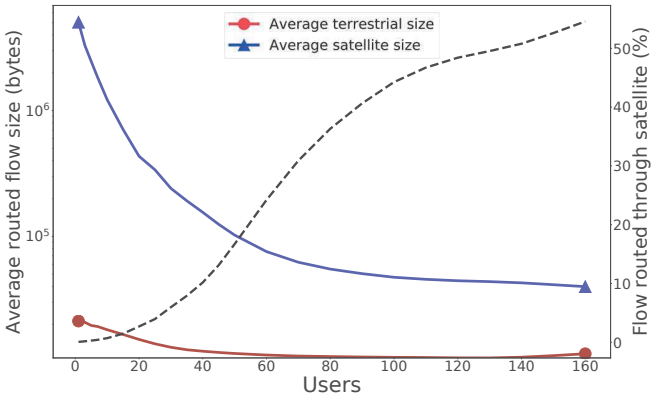

Fig. 4: Average flow size on each path with the MinFCT scheme.

on TCP; two connections are initialized in parallel, one on each path, the request being always sent by the IUG on the terrestrial path and the answer being sent on the path selected by ING with the MinFCT algorithm.

Contents size is a multiple of the MSS (1448 bytes) and the number of packets follows a Zipf law with a parameter α equal to 1.8 and shifted to a minimum of 8 packets. Between two requests, each user observes an exponential time with an average of 1 second.

B. Comparison of MinFCT with different strategies

1) Average Flow Completion Time: Figure 3 compares the average Flow Completion Time between MinFCT and MP-TCP hybridization methods, as well as standalone terrestrial and satellite paths using regular TCP, and this for different numbers of users in the system.

On the terrestrial path, we observe a very small average FCT for few users, thanks to the short path delay. However, there is a quick increase of the FCT as the number of users increases, the path being quickly saturated due to its small capacity. The gradient of the increasing FCT is related to the average flow length and the capacity of the path.

Oppositely, we observe an important average FCT for few users on the satellite path. Indeed, the FCT is bounded by the long path delay. Nonetheless, thanks to the high capacity, the satellite link can adequately fulfill users' requests for a relatively large amount of users. The link becomes saturated only when reaching 100 users. Also, the slope of the saturated portion of the curve is lesser than on the terrestrial path due to the higher path capacity.

The MP-TCP transport protocol shows its ability to improve the terrestrial path performance by aggregating the flows on the satellite path. However, only largest flows are concerned due to the "lowest RTT" MP-TCP scheduler. This scheduler sends a content on the path with the minimum Round-Trip Time, e.g. the terrestrial path. When the TCP buffer of the sub-flow is full, the scheduler starts sending the rest of the content on the satellite path. This operation has the major property of avoiding sending any parts of very short flows on the satellite path, which would have drastically increased the average FCT. Indeed, it would then be necessary to wait data flows in-flight on the satellite path.

Nonetheless, MP-TCP performance deteriorates with the load increase, not so much, but similarly to the use of the single terrestrial path. Indeed, the terrestrial network is heavily congested as all flows try to use it. As a result, the terrestrial path becomes unreliable, increasing the time necessary for connections setup and due to retransmissions.In addition, MP-TCP has specific mechanisms (cookies, losses handling, fallback to regular TCP...) which translates, with the congestion, into broken connections. Experiments with troubled connections are not represented as long flows are more likely to be subject to reset, distorting the results.

The MinFCT hybridization method dramatically outperforms all other schemes. For 30 users, the FCT is reduced by 32% compared to MP-TCP, 46% compared to the satellite network and 66% compared to the terrestrial network. This good performance is explained by several reasons. First, MinFCT avoids sending short flows on the satellite path when the transmission time is lower than the satellite delay. Then, as the load balancing is achieved at the flow-level, the terrestrial path is fully dedicated to flows that are not subject to the satellite delay. This method also asserts a sustainable congestion level on the terrestrial path, preventing its counterproductive use. By sending smallest contents on the terrestrial path, MinFCT increases the number of flows that benefit from the short delay, which improves the average performance.

With only one user, terrestrial, MP-TCP and MinFCT results are very close. Although hardly observable, MinFCT slightly beat the terrestrial network by sending large flows on the satellite path, and is slightly beaten by MP-TCP which aggregate large flows. Under congestion, we observe an asymptotic improvement of MinFCT against the satellite network about 20%, corresponding to the ratio of path capacities.

2) Average flows size: Figure 4 shows the average flows size on the terrestrial and satellite path.

With a single user in the system, there is no congestion, and best routing decision is always taken. Therefore, short and medium flows are routed through the terrestrial path while only longest flows are routed through the satellite path. As the number of users increases, as the congestion on the terrestrial

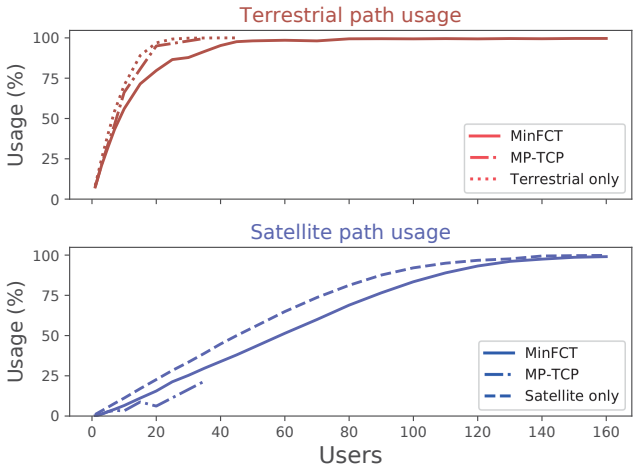

Fig. 5: Occupancy rate for terrestrial and satellite path, considering routing through the terrestrial path only, satellite path only, with MP-TCP or with the MinFCT algorithm.

path, medium flows and even part of the short flows become routed through the satellite path. As a result, we observe a decrease of the average flows size on terrestrial and satellite paths.

When the number of users exceeds 140, the satellite path becomes also congested. Medium flows become partly routed through the terrestrial path again causing an small increase of the average flows size on the terrestrial path. As the satellite path is still used for many medium and short flows, over-represented due to the Zipf law, we observe no visible increase of the average flows size on this path.

3) Paths occupancy rate: On figure 5, we observe the path occupancy ratio, defined as the average transmitted rates divided by the path capacity. We compare one path strategies with the MP-TCP and MinFCT strategies.

With the MinFCT scheme, the terrestrial path is congested from about 40 users whereas, when using the terrestrial path only, it is congested starting at 20 users. On the satellite path, we observe a reduced use of the available capacity, many short flows being sent on the terrestrial path. We observe with MP-TCP a consequent congestion of the terrestrial path while the satellite path is underused. This behavior has already been pointed out as the main reason for MP-TCP bad performance.

Conclusion

In this paper, we presented an architecture and a routing method allowing to take advantage of a heterogeneous network, composed of a terrestrial path and a satellite path. Contrary to other approaches, this method takes into account flows size to minimize the Flow Completion Time. For that, we presented an architecture able to retrieve flows size through Content Delivery Network Interconnection. We outlined a method to estimate the path delay, capacity and loss rate and finally an algorithm to reckon the Flow Completion Time considering the estimated path characteristics and the flow size. Finally, we have implemented a testbed and have evaluated our proposed routing method by comparing its performance to that of single path routing strategies and of the MP-TCP transport protocol.

Our results show the significant improvement brought by our size-aware routing compared to the MP-TCP approach. Though, many improvements are considered, in particular improving path metrics gathering, especially at low loads, or fine-tuning the estimation algorithm to take better decisions. The interconnection protocol of the Intelligent Gateways is subject to many enhancements, with the use of alternative transport protocols like SCTP or QUIC, the use of TCP Fast-Open or even the implementation of a reusable connection pool. Also, we plan to evaluate the performance of the method with different traffic laws and parameters, as well as different path characteristics. Finally, the method could be extended to non-CDN context with a different source of information for the size knowledge or its estimation.

References

[1] J. Qadir, A. Ali, K. L. A. Yau, A. Sathiaseelan, and J. Crowcroft, "Exploiting the power of multiplicity: A holistic survey of network-layer multipath," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2176–2213, Fourthquarter 2015.

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

[3] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 2, pp. 5–12, Apr. 2005.

[4] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, "Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2016, pp. 431–439.

[5] M. Xu, Y. Cao, and E. Dong, "Delay-based congestion control for mptcp," Working Draft, IETF Secretariat, Internet-Draft draft-xu-mptcp-congestion-control-05, January 2017.

[6] A. Walid, Q. Peng, J. Hwang, and S. H. Low, "Balanced linked adaptation congestion control algorithm for mptcp," Working Draft, IETF Secretariat, Internet-Draft draft-walid-mptcp-congestion-control-04, January 2016.

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

[8] E. Bouttier, R. Dhaou, F. Arnal, C. Baudoin, E. Dubois, and A. L. Beylot, "Analysis of content size based routing schemes in hybrid satellite / terrestrial networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[9] R. Stewart, "Stream control transmission protocol," Internet Requests for Comments, RFC Editor, RFC 4960, September 2007.

[10] G. Peng, "CDN: Content Distribution Network," *eprint arXiv:cs/0411069*, Nov. 2004.

[11] B. Niven-Jenkins, R. Murray, M. Caulfield, and K. Ma, "Content delivery network interconnection (cdni) metadata," Internet Requests for Comments, RFC Editor, RFC 8006, December 2016.

[12] E. Bouttier, R. Dhaou, F. Arnal, C. Baudoin, E. Dubois, and A. L. Beylot, "Heterogeneous multipath networks: Flow vs packet based routing in a size-aware context," in *2017 IEEE Global Communications Conference (GLOBECOM)*, Dec 2017, pp. 1–6.

[13] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "Tcp selective acknowledgment options," Internet Requests for Comments, RFC Editor, RFC 2018, October 1996.

[14] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82, Jul. 1997.

[15] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, "Increasing tcp's initial window," Internet Requests for Comments, RFC Editor, RFC 6928, April 2013.

[16] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Computing tcp's retransmission timer," Internet Requests for Comments, RFC Editor, RFC 6298, June 2011.