# Evaluating Web Traffic Performance over DVB-RCS2

Raffaello Secchi, Arjuna Sathiaseelan*, and Gorry Fairhurst

Electronics Research Laboratory (ERG),
University of Aberdeen, AB24 5UE, Aberdeen, UK
{raffaello,arjuna,gorry}@erg.abdn.ac.uk

**Abstract.** The web has undergone a radical change over time. Changes not only in the volume of data transferred, but also the way content is delivered to the user. Current web server architectures are often highly distributed and adapted for user interaction, with transactions characterised by multiple connections to multiple servers. This paper discusses the implication of this new web on next generation two-way satellite systems. It seeks to answer the question of whether classical resource provisioning remains suitable for this traffic. It first presents a more representative simulation model that captures the key features of modern web traffic. It then uses simulation to evaluate the performance over the second generation of DVB-RCS, assessing the impact on performance for a range of bandwidth on demand methods. This paper may be used to formulate recommendations for how to support web traffic in DVB-RCS2.

**Keywords:** Web Traffic, HTTP modelling, DVB-RCS capacity categories.

## 1 Introduction

In the last decade the Word Wide Web (WWW) has radically changed both in terms of complexity and usability. In May 2010 Google published statistics [1] about the size and composition of web pages. This showed many current web pages are significantly more complex compared to when version 1.1 of the Hypertext Transfer protocol was specified HTTP/1.1 [3]. This increase in complexity has been accompanied by a substantial increase in HTTP traffic, with a current average web page size at least ten times larger than a decade ago [3]. Since web is a key service carried by satellite, it is important for the satellite community to understand the implications of this new web, especially the implications on emerging next generation systems, such as the new DVB-RCS2 (Digital Video Broadcasting  Return Channel via Satellite ver. 2) standard[4]. This requires a

* Raffaello Secchi was supported by Astrium UK, subsidiary of the European Aeronautic Defence and Space (EADS) company. Arjuna was supported by the Satellite Internet for Rural Access (SIRA) project funded by the RCUK Digital Economy Programme.

P. Pillai, R. Shorey, and E. Ferro (Eds.): PSATS 2012, LNICST 52, pp. 148–155, 2013.

detailed study of web characteristics, development of a representative simulation model and evaluation of the interactions with the satellite system.

Past work has explored the interactions of TCP and satellite bandwidth allocation methods [2,13,14,15]. These papers have considered server workload models to explore the effects of web over the first generation DVB-RCS system. The interaction of HTTP over DVB-RCS was studied in detail in [15]. Although these papers presented detailed web models, they may not be able to capture the essential nature of the present web. They lack: (i) Support for a distributed server architecture, employed by the majority of popular web services, (ii) Modelling of web browser configurations (e.g. number of simultaneous connections per server and proxy-based approaches), (iii) Support for the network-layer IP signalling required to download a web page, such as the domain name service resolution (DNS). This raises concern about the effectiveness of the models for evaluating HTTP performance and use to explore design decisions for future satellite systems.

This paper is a first step to understand the implications of current web on satellite networking. We define a web model that supports multiple connection per server to fetch the objects that we suggest closely reflects the current parameters of web traffic. Then we use the model to simulate a web transfer across a satellite network with the classical Bandwidth on Demand (BoD) allocation methods. Our results show interesting properties of dynamic allocation to web flows.

The remainder of the paper is organised as follows: Section 2 describes the key characteristics of the web and provides a brief overview of our proposed simulation model. Section 3 discusses the performance of web over DVB-RCS2, followed by conclusions.

### 1.1 The World Wide Web

HTTP [8] is the de facto protocol for delivery of web pages. In HTTP, a web browser acts as a client, while a server application hosts the web site. The client submits a HTTP request message to the server. The server returns a response message to the client, which contains completion status information and may contain content requested by the client. A web transfer usually consists of a sequence of these request-response transactions, possibly to multiple servers, in order to transfer the set of objects forming a web page.

HTTP was initially standardised as HTTP/1.0 [8] and revised in HTTP/1.1[9]. While HTTP/1.0 used a separate connection to the same server for every request-response transaction, HTTP/1.1 can reuse the underlying transport connection multiple times, (process known as *HTTP pipelining*), to deliver the objects at the same server for a given web page. Since establishment of new TCP connections incur considerable delay, web pages transferred using HTTP/1.1 experience less latency. HTTP/1.1 also introduced the *chunked transfer* method to allow an object to be sent as a continuous stream and the *byte serving* method, allowing a server to transmit any portion of a resource. Currently these methods are used extensively in web transfers.

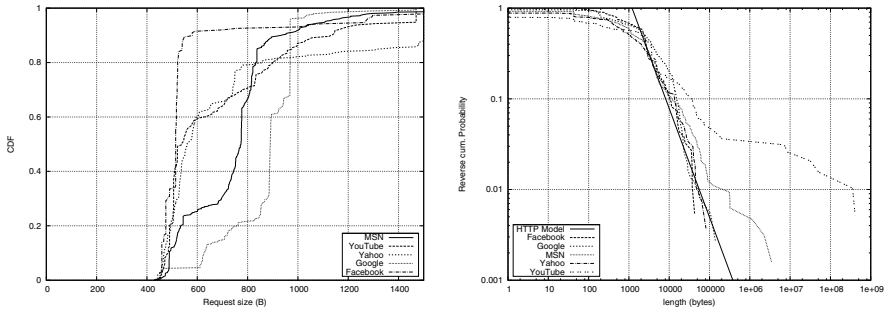


**Fig. 1.** CDF of HTTP request size from a selected pool of popular websites and reverse CFD of the relative response size

### 1.2 Characterisation of Web Traffic

Statistics published by Google [1] show that the overall size of a webpage is 320 KB on average. This study found that typical web pages consisted of many objects (around 40) that are downloaded in parallel from 7 connections on average. HTTP requests often includes also the optional body, which carry user information (the Cookie) increasing substantially the size of the request packet. The size of the response i.e. the downloaded object size follows a Pareto distribution with a mean object size of 7187 bytes and a shape parameter of 1.2 [1].

To explore these findings a set of measurements were performed using a pool of popular web sites. Figures 1a and 1b respectively show the cumulative distribution function (CDF) of the size of requests and the reverse CDF of the size of responses of some websites we considered. We observed that an HTTP request is typically 200 bytes longer than a decade ago [5], mostly due to Cookie insertions, and much more variable in length. Requests can be larger than a TCP maximum segment size (MSS). On the other hand, the characteristic distribution of HTTP responses has been preserved and can be well approximated, as predicted in [1], by a Pareto distribution with average 7.1 kB and shape parameter 1.2 (see Figure 1(b)). However, Figure 1(b) shows that the Pareto model may not be suitable when using HTTP streaming. For example, a YouTube session may embed video-clips in HTTP Streaming that would not fit this model.

Modern web scenario is characterised by heterogeneity of web technologies. Different web browsers use different policies when downloading webpage objects. When a webpage is accessed by clicking on the URL, the client sends a HTTP GET request to the server requesting the main file (index.html). Once the main file is downloaded, the client then sends requests to download the objects that are indexed by this file. These requests are sent concurrently, limited by the allowed number of concurrent connections. This number varies between browsers. RFC 2616 [9] originally stated that clients that use persistent connections should limit the number of simultaneous connections that they maintain to a given server. A single-user client should not maintain more than 2 connections with any server or proxy. However, Internet Explorer, IE 8, allows 6 concurrent connections whereas

IE 7 and earlier allowed only 4 connections [10]. Opera developers suggest a maximum of 8 connections per server. Mozilla Firefox allows a maximum of 15 connections per server [11]. Google Chrome uses a maximum of 6 parallel connections per group.

### 1.3 Characterising the Page Based Simulation Mode

A preliminary survey of simulation models found that many web traffic generators are still based on outdated parameters and have not been updated to reflect the characteristics presented in the previous section. For example, the web models available in the popular ns-2 simulation package [3,5,6,7] may not be able to capture the real web scenario. To mitigate this problem we developed a web traffic model based on [1] and our laboratory analysis. Due to lack of space, we could not report our entire analysis (Figure 1 reports the request/response sizes). However, the following text summarises the steps necessary for a client to retrieve a web page.

A HTTP transaction starts with a DNS request/response. Initially, a HTTP request is sent for the first main object. Once received, a set of concurrent connections are opened (the TCP connection used to transfer the first object may be reused). Subsequently, the client requests a DNS lookup for each server it needs to access. When the connections get the responses for their requests, they may be reused:

1. The client first sends a DNS request to determine the IP address of the first server it must connect. The client receives a DNS response. The RTT is assumed to be twice the simulated one-way delay.
2. Once the DNS response is received, the client sends a HTTP GET request to the first server. It returns a HTTP response (the main object).
3. Once a client receives the main object, it tries to fetch all the related objects from the server (or multiple servers). It sends DNS requests for multiple servers (if any). Our model uses a random number of servers (up to 7). We also model a random number of objects at each server.
4. Once DNS responses have been received, the client opens concurrent connections to each server. An inter-server delay represents the time between requests to different servers, modelling client processing.
5. The client closes all TCP connections when all the objects have been downloaded for a webpage[1].

## 2 Performance of Web Traffic with Capacity Categories

This section describes a set of simulations to evaluate the interaction between the capacity categories of DVB-RCS2 [4] and the presented web traffic model.

[1] If further objects are to be fetched from the first or main server, the client opens parallel TCP connections to fetch these objects. The TCP connection used to fetch the main file is reused.

This uses an ns-2 simulation model of the DVB-RCS2 [16]. The simulator models an RCS terminal (RCST) and the corresponding gateway that provides Internet connectivity to the satellite network. The RCST make a capacity request (CR) for a set of TCP flows from the network control centre (NCC). The NCC makes corresponding allocations, providing return link (RL) capacity towards the gateway.

An HTTP server may be accessed by an RCST via a gateway. In this case, the RL transports mostly HTTP requests and acknowledgements (ACKs). In an alternate use a web server may be provided at an RCST. The former is typical for traditional broadband satellite access using a star network topology, whereas the latter is possible when an RCST is uses as a gateway in a regenerative satellite system or as a mini-gateway in a star system enabled to support Mesh connectivity. This second scenario is used to provide a good illustration of the impact of the BoD dynamics on web service performance.

The web server receives requests for a complex webpage (43 objects) and returns the web contents through the RL. In a typical star network, the NCC incurs an allocation delay, i.e. the delay between a CR and its corresponding assignment, is about two satellite round trips (about 640 ms). The CR request period in our simulations uses one request per second when traffic activity is detected. The satellite frame period was 26.5 ms and consisted of carriers with 16 timeslots. Packets were encapsulated using a return link encapsulation (RLE) [4] with a burst size of 53 bytes. The total transmission rate is 256 kb/s.

The allocation method used a combination of RCS capacity categories. The RCST sends Rate Based Dynamic Capacity (RBDC) requests every second averaging the input traffic over the past interval. A smoothing filtering with parameter $0 \leq \alpha \leq 1$ can be also applied to the rate samples. RBDC($\alpha$) denotes in our graphs an RBDC method with filtered samples. The RCST may use a Volume Based Dynamic Capacity (VBDC) request in addition to the RBDC request in the request message. The NCC only allocates the minimum amount of slots to satisfy a request in each transmission burst time plan (TBTP).

The client used a transport based on TCP New-Reno SACK. TCP parameters were default with a packet size of 1500B (including the TCP/IP header). The simulations considered three web-pages whose object sizes were extracted from the previously described empirical distribution. The total web page sizes were 212, 178, and 315 KB.

### 2.1 Performance Analysis

Figure 2 reports the completion of the web page transfer and the corresponding allocation efficiency. The efficiency is calculated as the amount of capacity used (in bytes) including encapsulation overheads with respect to the amount allocated, and it does not consider the capacity allocated after the completion of a web transfer until the RCST channel becomes idle.

The number and the size of objects forming a web page can vary. Apart from the BoD type and page size, many other factors influence the transfer completion time (TCC), such as the burst size, carrier bitrate, order of scheduling of HTTP

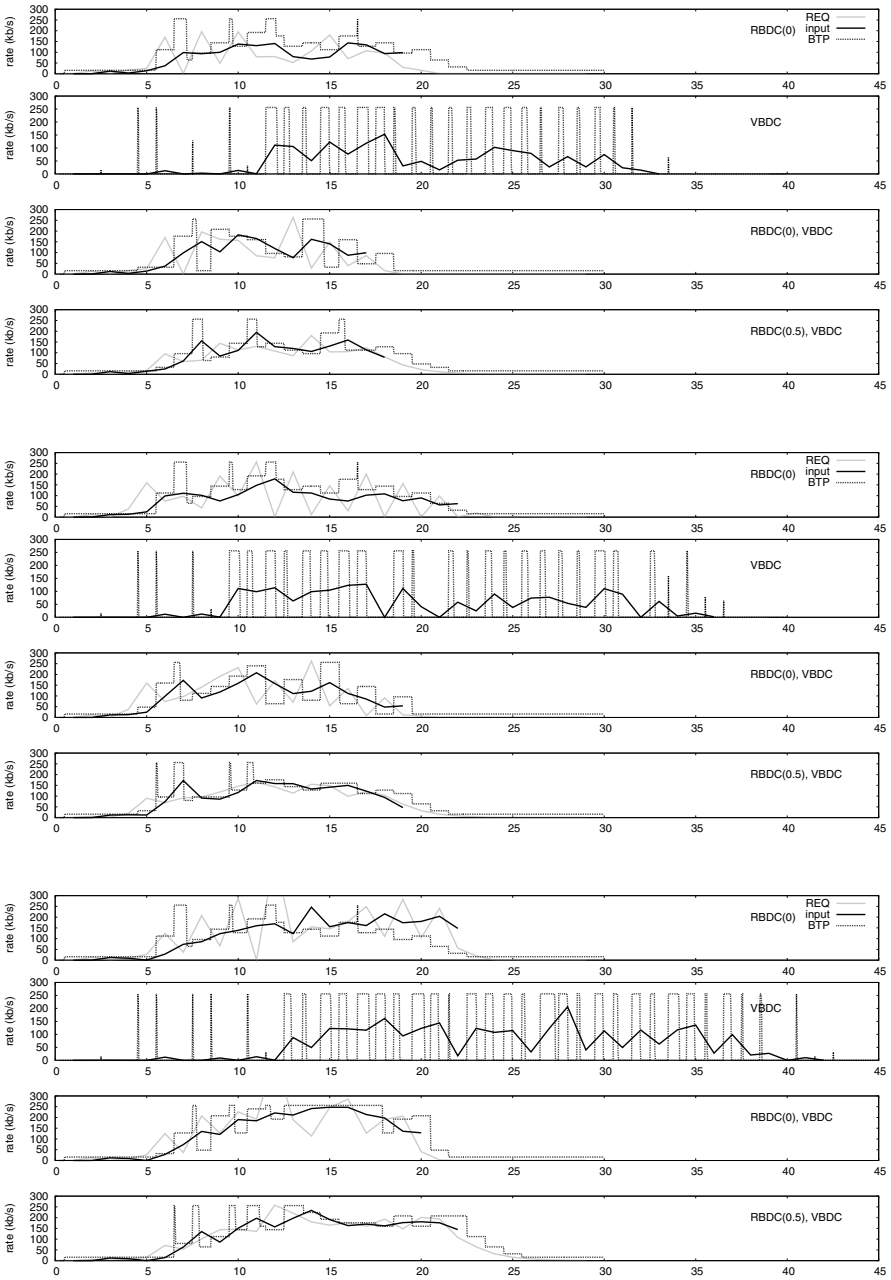


**Fig. 2.** Figure1Dynamics of application transmission rate (input), capacity requests (REQ), and allocated rate (TBTP) for four allocation methods (RBDC(0), VBDC, RBDC(0) + VBDC, and RBDC(0.5) + VBDC). Three HTTP test sets with different distributions of object size.

**Table 1.** Completion time and allocation efficiency (bytes used/allocated) for web transfers for different BoD methods

| Dataset | RBDC(0) | VBDC | RBDC(0), VBDC | RBDC(0.5), VBDC |
|---|---|---|---|---|
| Set 2 (178 kB) | 18.6 s, 0.86 | 30.8 s, 0.99 | 16.7 s, 0.81 | 17.4 s, 0.77 |
| Set 1 (212 kB) | 21.6 s, 0.87 | 33.8 s, 0.98 | 18.4 s, 0.85 | 17.8 s, 0.82 |
| Set 3 (315 kB) | 21.7 s, 0.86 | 39.8 s, 0.99 | 19.2 s, 0.85 | 21.5 s, 0.85 |

requests at the client, the encapsulation efficiency, etc. This makes it difficult to formulate precise recommendations. Despite this, our analysis highlights some important results:

RBDC allows shorter completion times compared to the VBDC. VBDC tends to request and allocate capacity in bursts (the allocation pattern is ON/OFF). This increases the RTT seen by TCP and slows down the transfer. However, the VBDC efficiency is nearly ideal (see Table 1), since VBDC tends to request the exact amount of capacity that needs to be allocated for the queued traffic at an RCST.

A combination of RBDC and VBDC, whether or not the RBDC requests are pre-processed by a filter, provides better performance than RBDC or VBDC alone.Combining RBDC and VBDC leads to a lower utilisation of the requested bandwidth with respect to VBDC. For instance, the median of the efficiency of RBDC was observed to be between 83% and 94%, and between 77% and 85% for the combined method. This is probably due to the high variability of web traffic rate during the allocation period. The compound method produces performance better than RBDC alone, and can be used to trade latency for efficiency.

## 3 Conclusions

Changes in HTTP usage have been accompanied by an increase of web document complexity and a transition from single server domains to distributed architectures. This calls for a revision of the models used to evaluate performance for web traffic.This paper proposes a model that captures the characteristics of the HTTP request/reply transaction and a multiple-server architecture.

This model was to simulate HTTP performance with standard bandwidth allocation mechanisms over a DVB satellite link. Our results show that the dynamics of the allocation for a web flow can be significantly affected by the allocation method. We found that the latency of the transmission of a web page is less when RBDC and VBDC were combined rather than using them alone. The performance gain is clearly dominated by the allocation pattern, which is much more stable when RBDC and VBDC work together.

Our research also uncovered pathologies in the standard allocation mechanism that must be considered when designing new allocation strategies. Future research will extend this to include a wider range of allocation mechanisms and provide recommendations for efficient transport of web traffic.

## References


1. Ramachandran, S.: Lets Make the Web Faster, Google (May 2010), http://code.google.com/speed/articles/web-metrics.html
2. Sooriyabandara, M., Fairhurst, G.: Dynamics of TCP over BoD satellite networks. Int. J. Satell. Commun. Network. 21, 427–449 (2003)
3. Cao, J., Cleveland, W.S., Gao, Y., Jeffay, K., Smith, F.D., Weigle, M.C.: Stochastic Models for Generating Synthetic HTTP Source Traffic. In: IEEE INFOCOM, Hong Kong (March 2004)
4. Digital Video Broadcasting (DVB), Second Generation DVB Interactive Satellite System (RCS2): Part 1: Overview and System Level Specification, DVB Document A155-1 (March 2010)
5. Mah, B.A.: An Empirical Model of HTTP Network Traffic. In: IEEE INFOCOM, Japan (April 1997)
6. Floyd, S., Paxson, V.: Difficulties in Simulating the Internet. IEEE Transactions on Networking 9(4), 392–403 (2011)
7. Abrahamsson, H., Ahlgren, B.: Using Empirical Distributions to Characterise Web Client Traffic and to Generate Synthetic Traces.In: IEEE GLOBECOM, San Francisco (December 2000)
8. Berners-Lee, T., Fielding, R., Frystyk, H.: Hypertext Transfer Protocol HTTP/1.0, IETF RFC 1945 (May 1996)
9. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1, IETF RFC 2616 (June 1999)
10. AJAX - Connectivity Enhancements in Internet Explorer 8, http://msdn.microsoft.com/en-us/library/cc304129(VS.85).aspx
11. Network. http.max-connections, MozillaZine, http://kb.mozillazine.org/Network.,http.max-connections-per-server
12. Hernandez-Campos, F., Jeffay, K., Smith, F.D.: Tracking the Evolution of Web Traffic: 1995-2003. In: IEEE/ACM MASCOTS 2003, Orlando, Florida, (October 2003)
13. Roseti, C., Kristiansen, E.: TCP behavior in a DVB-RCS environment. In: 24th AIAA International Communications Satellite Systems Conference (ICSSC), San Diego (June 2006)
14. Roseti, C., Kristiansen, E.: TCP Noordwijk: TCP- Based Transport Optimized for Web Traffic in Satellite Networks. In: 26th International Communications Satellite Systems Conference (ICSSC), San Diego (June 2008)
15. Luglio, M., Roseti, C., Zampognaro, F.: Improving performance of TCP-based applications over DVB-RCS links. In: IEEE International Conference on Communications (ICC), Germany (June 2008)
16. Secchi, R.: DVB-RCS2 for ns2 (2011), http://www.abdn.ac.uk/eng863/dvbrcs_ns2.htm