Improving Performance of Multimedia Web Transfer over WAN Connections

Hristo Valchanov[1], Milen Angelov[2]

Abstract – **The paper presents an approach to improve the performance of the data transmission between networks based on WAN technologies. The architecture and internal details of the SCTP web proxy server are shown. Some performance experiences are performed and the results indicate that the proposed approach can improve the performance of the transfer of multimedia web documents over slow speed lines compared to the TCP protocol.**

Keywords – **HTTP, SCTP, Web Proxy, WAN.**

I. INTRODUCTION

Modern web-based technologies require speed and reliability of data delivery. The current transport model is based on the TCP protocol, which guarantees delivery of data, but brings significant cost in communications. If a part of an object is not delivered due to loss in networks, the object will not be delivered to the customer, while this lost part is not re-transmitted and delivered successfully. This problem is known as "TCP Head of Line Blocking" (HOL). The transfer of multimedia pages in networks based on WAN technologies such as VSAT, 3G, GPRS and others is accompanied by a significant delay in the range of 100ms to several seconds.

Multi-stream transmission on the transport level is the ability of the transport protocols to support different data streams. For each stream an independently maintained order of delivery of packets is organized. The Stream Control Transmission Protocol (SCTP) [1] is a standardized reliable transport protocol providing multi-stream transmission. Within a SCTP association (a term indicating the session transport level) can be transmitted independent HTTP responses [2].

The use of SCTP as a transport level protocol for HTTP can solve many of the problems of the current model of exchange of multimedia documents in WAN infrastructures. Since multimedia documents are composed of objects of different type and size, the multi-stream transmission allows sending them in partially order rather than in strictly order, and may contribute to a better user perception when loading the page. At the same time, the transport is carried out within one SCTP association, so all streams use a common mechanism for flow control, which reduces the overhead on the transport level.

[1]Hristo Valchanov is with the Computer Science and Engineering Department at Technical University of Varna, 1 Studentska Str., Varna 9010, Bulgaria, E-mail: hristo@tu-varna.bg.

[2]Milen Angelov is with the Computer Science and Engineering Department at Technical University of Varna, 1 Studentska Str., Varna 9010, Bulgaria, E-mail: mgangelov@gmail.com.

The big problem of the implementation, however, is expressed in the difficulty in modifying existing web servers and clients. Currently known is only one open source development of such a server, which is based on the Apache implementation [3, 4, 5]. The project involves adapting the server core - APR (Apache Portable Runtime) to support SCTP, and modification of the browser Firefox. Unfortunately, this project currently is discontinued for financial reasons.

This paper presents a different approach - using existing software solutions (HTTP servers and clients) operating on TCP protocol, and transmission of data over WAN networks through SCTP protocol. The motivation is driven by the fact that the server and the client are in local networks, characterized by a high bandwidth. These networks are connected to the WAN technologies that have low (in magnitude) bandwidth. With this approach we can use the full multi-streaming capabilities of the SCTP protocol. The paper presents some features of an implementation of TCP / SCTP proxy server and experimental study of its performance.

II. FEATURES OF SCTP PROTOCOL

The SCTP protocol is standardized by IETF. It offers features similar to TCP - reliable session-oriented transport which guarantees order of data delivery. In addition, SCTP provides new functionalities. Unlike TCP, which is oriented to byte-transmissions, SCTP is message oriented. The session is called "association". Creating an association requires negotiation in 4 stages (4-way handshaking), where data can be included in the third and fourth negotiation message and sent when the association has already been validated. For protection against certain types of DoS attacks, in the process of negotiation is built the mechanism of cookies.

Ones of the benefits of SCTP are multi-stream transmission (multi-streaming) and maintenance of multiple addresses per host (multi-homing). By the multi-streaming the transmitted within an association data can be divided into several separate streams, each maintains its own sequence of delivery. For each stream can be configured ordered or random delivery. Lost packets from one stream do not affect the other streams. This allows elimination of the HOL problem. But because the transmission is organized within an association, all streams are subject to common flow control mechanism of data exchange and network congestion.

The support of multiple addresses allows each endpoint of the SCTP association to have multiple IP addresses. To make this, peers share a list with these addresses during the process of negotiation. Each peer uses a single port number, with respect to these addresses. Only one of them is active and is used for the exchange, while the others provide redundancy in case of failure of the active IP address. This feature of the protocol provides high reliability of the communication by maintaining multiple routes.

III. The Proxy Architecture

In computer networks, a proxy server is a server (a computer system or application program) that acts as an intermediary for the client applications which require resources or services from other servers. The client connects to the proxy server and requires some resource, such as a file, connection or web page, available from a different server. The proxy server checks the request in accordance with rules for filtering. For example, it may filter traffic by IP address or protocol. If the request is permitted, the proxy server provides the requested resource by connecting to the relevant server (which actually provides the service), stating the service on behalf of clients. The proxy server can also change the client requests or server responses, and sometimes it may serve the request without contacting to another server. In this case we are talking about so-called "caching query" to a remote server.

A. Proxy Dual Stack

The main idea of the proposed approach is that the presented TCP / SCTP proxy will work as an interface between TCP and SCTP protocols, allowing web browsers and servers to take advantage of the capabilities of the SCTP, without having to change their code. Because the goal of the current implementation is studying of the capabilities of the approach, the structure of the proxy server is made simple, providing only basic functionality necessary to conduct the study.

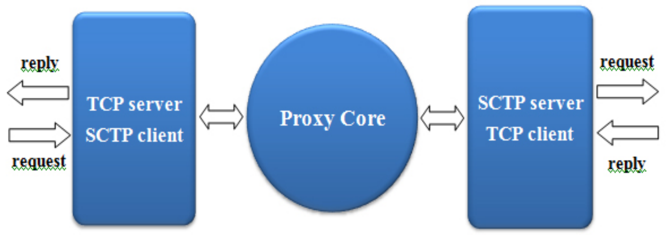

Fig. 1. Proxy dual stack

The functionality of the proxy server is implemented on a dual stack model (Fig. 1). The dual stack uses two transport protocols - TCP and SCTP, thus allows easy integration of the proxy server in the TCP / IP infrastructures. When a TCP request is received from a client (web browser), the proxy acts as a TCP server. When it forwards this request to another SCTP proxy, it operates as a SCTP client. Similarly, upon receiving of a reply from a TCP server (web server), the proxy acts as a TCP client, and upon returning the response to a SCTP proxy it operates as a SCTP server. The Proxy Core component implements the base functionality of the proxy server.

B. The Proxy Architecture

The internal structure of the proxy server consists of 4 different modules (Fig. 2).

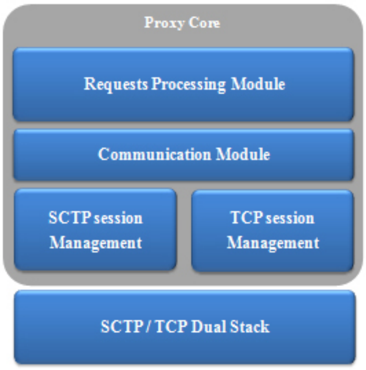

Fig. 2. Architecture of the proxy server

The communication management module (CMM) supports information to determine the correct socket to which the received packet is forwarded. In the current implementation a static configuration of the IP address and port number of the destination is done. It is planned in future versions of the proxy server dynamically to determine the IP address and to maintain associations with multiple servers based on SCTP (this version supports only one).

The communication module transfers data between TCP and SCTP management modules. To do this, it uses a buffer with a fixed size, which is set to be equal to the maximum length of user data in a package. The received data is stored in this buffer, and then read from it before the actual forwarding.

The TCP session management module is based on Linux system library. It includes the following features:

- Establishing and closing the session.
- Confirmation and avoiding traffic congestion - mechanisms by which a messages loss is detected and the flow of data is controlled.
- Validation of data - checksum is used to check whether the data were corrupted during transfer.
- Sorted delivery - the data are possible to arrive in stirred sequence. This feature cares for data to be properly ordered.
- Fragmentation - when transferred data are overwhelming they break into multiple fragments on sending and then assembled upon receiving.

The SCTP session management module uses the sctplib-1.0.11 library (RFC4960) [6]. The base functionality includes:

- Sorted delivery within a stream - for each individual stream the module monitors that data arrive in order. Loss of data in a stream does not affect the other streams.
- Establishing and closing association - includes mechanisms for creating, normal closing and interruption of association.
- Packaging data - in a single SCTP message few fragments of data can be packed.

- Fragmentation of customer data.
- Confirmation of data delivery and avoiding traffic congestion.
- Validation of data.

The Request Processing Module determines the length of transferred resource – the first response that a web server sends after receiving a request for a resource, consists in the HTTP Entity header “Content-Length” the length of the requested resource. The proxy server retrieves this length, updates it on sending of any part of the resource, and thus determines when the end of the transfer of the current object has come.

The transfer of data between TCP and SCTP dual stack uses buffer with a fixed size, which is set to be equal to the maximum length of user data in one package. The received data is stored in this buffer, and on sending, they are read from the buffer.

C. Managing Sockets and Streams

To support parallel processing of multiple requests, there is a need for an internal data structures to maintain associations between TCP sockets and SCTP streams. Implementation is based on structure *iface*, which associates a TCP socket with a stream number (Fig. 3).

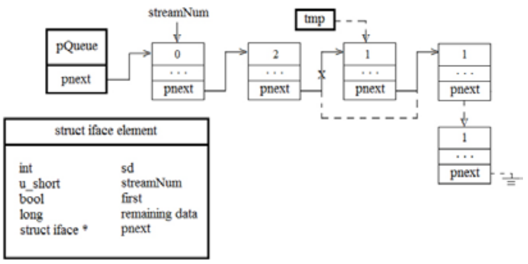

Fig. 3. TCP-SCTP association data structure

The elements of this structure form a queue of requests. Newly queries are inserted at the end of the queue. An item from the queue is retrieved as follow: getting the first element corresponding to the required number of stream.

Upon receiving a TCP request, the proxy server accepts a socket, using the system call *accept ()*. CMM allocates an element of the structure *iface* and stores into them the returned from *accept()* socket identifier.

Selecting a SCTP stream on which the request must be sent is made according Eq. 1.

$$S = (O_{prev} + 1) \bmod N \qquad (1)$$

,where

S - number of needed output stream;

O_{prev}- number of previously used output stream;

N - maximum count of streams.

It is implemented a mechanism to reject / retain new requests if all streams are in use. This mechanism is used to avoid the problem with streams overlapping, because this may destroy the correct operations of the server. In addition, this can be used as a control mechanism that manages the maximum number of parallel transmitted objects. The new arrived request is stamped as newly, which then affects how to process it upon receiving of a SCTP reply. Finally, the request is read from the socket into a buffer and sent to the next proxy server through the selected SCTP stream.

Upon receiving the SCTP reply, CMM performs a linear search from the beginning of the queue, up to the first element whose stream number matches the number of stream from which data were arrived. The item is removed from the queue and stored in a temporary position. From this item is obtained the identifier of the socket on which the request was sent. Next, SCTP data are extracted and sent by the corresponding TCP socket.

If the request is stamped as newly, the SCTP reply will contain the length of the resource to be transferred and proxy derive it and stored in the temporary variable. Otherwise, from the remaining length of transmitted resource (that has been identified at the moment of first inserting of the request in the queue) the CMM subtracts the number of bytes that have just been sent. If there are still data to send, the temporary variable is inserted back into the queue, because the request shall not be considered for newcomers. Otherwise, the TCP socket is closed and the memory allocated by the temporary variable is freed.

IV. EXPERIMENTAL STUDY AND RESULTS

The purpose of the experimental research is to study the behavior of the two transport protocols in network infrastructures, based on WAN technologies. These types of technologies provide bandwidth in order of magnitude, smaller than the technologies used for LANs. This is an important factor that globally influences the response time of web servers.

The analysis of the performance of the proposed approach is made on network infrastructure representing slow WAN connections as shown on Fig. 4. The test environment includes Cisco routers 2901, VLAN Cisco Catalyst 2960 switches, PCs HP Desktop 500B CPU Intel Core Duo E5800 3,2 GHz with 2G RAM. The web server platform is based on Slackware Linux 2.6 and gcc 4.4.3. As a client is used Windows 7 OS and browser Google Chrome 28.0.1500.

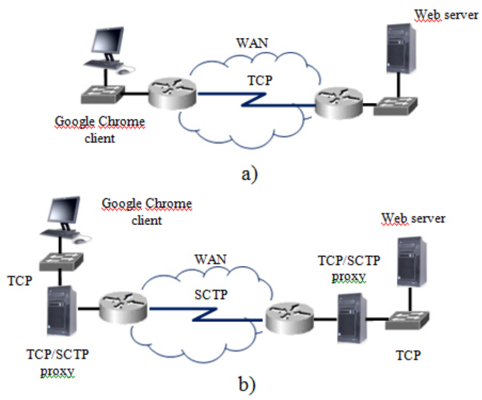

Fig. 4. The test environment

Between routers serial links based on protocol HDLC are built. By their configuration with different speeds is simulated a WAN transmission media with different bandwidth. Experiments were carried out in two directions:

- direct communication between web client and server using TCP protocol (Fig. 4-a);
- indirect communication between web client and server through TCP / SCTP proxy (Fig. 4-b).

The links between client, proxies and routers are with speed of 100Mb, thus simulates typical LAN infrastructures.

Two test groups were performed - transferring files and loading web pages. Experiments were performed at various speeds of the serial links between routers - 64000 bps, 115200 bps and 128000 bps. The client downloads a multimedia page of fixed size, which contains various objects located on the same server and a file with size of 2 Mb. Certain measurements are made and average download speeds of data are recorded.

Figure 5 shows the obtained results.

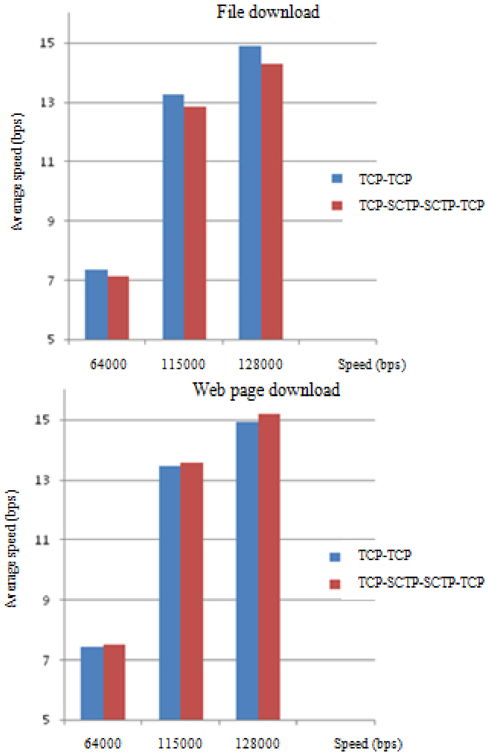

Fig. 5. The obtained results

The test with loading a webpage using SCTP protocol has better response times. This is due to the common flow control mechanism of SCTP, which reduces the amount of used control information, in comparison with TCP protocol, especially when multiple resources are transmitted in parallel. Page loading is carried out within a single SCTP session (unlike TCP, where for each resource a new session is created).

When downloading a single file, the mechanism for controlling the SCTP streams is not efficient, because it uses only one stream. In contrast, the TCP protocol needs only one session for the file transfer. This introduces negligible delay compared to a scenario where multiple resources are transferred.

In addition may be noted that both protocols provide reliable delivery of data without loss of information.

V. Conclusion

The paper presents an approach for improving performance of downloading multimedia web pages through slow WAN connections. The proposed approach is based on using existing software solutions (HTTP servers and clients) operating on TCP protocol, and transmission of data over WAN networks through SCTP protocol. The paper presents some aspects of the implementation of developed TCP / SCTP web proxy server. Several experimental tests are performed. The obtained results show that the proposed approach is fully functional and applicable for transferring multimedia data over the networks with low bandwidth.

Goal of future work is to develop a multithreaded architecture of the proxy server. This will allow it to serve simultaneously multiple clients. Another future work is to add capabilities for data caching and traffic filtering.

References

[1] An Introduction to the Stream Control Transmission Protocol (SCTP), http://tools.ietf.org/html/rfc3286.
[2] Hypertext Transfer Protocol - HTTP/1.1, http://tools.ietf.org/html/rfc2616.
[3] The Apache Software Foundation, http://www.apache.org.
[4] P. Natarajan, J. Iyengar, P. Amer. SCTP: An Innovative Transport Layer Protocol for The Web. Proc. of 15th Conf. of WWW, pp.23-26, 2006.
[5] SCTP apache server: http://www.eecis.udel.edu/~leighton/firefox.html.
[6] SCTP library: http://www.sctp.de/sctp.html.