Performance Comparative Study of eXtended Satellite Transport Protocol over Traditional Satellites Networks and Nanosatellite Constellations

Maria-Mihaela BURLACU
Dept. MIPS/GRTC, University of Haute Alsace
Colmar, France
e-mail: maria-mihaela.burlacu@uha.fr

Pascal LORENZ
Dept. MIPS/GRTC, University of Haute Alsace
Colmar, France
e-mail: lorenz@ieee.com

Joséphine KOHLENBERG
Dept. RST, IT/Télécom SudParis
Evry, France
e-mail: Josephine.Kohlenberg@it-sudparis.eu

Abstract*—The design of efficient communication mechanisms for small satellite networks is a challenging task, requiring the definition and implementations of specific protocols and architectures appropriate to space's critical conditions. In this paper, we have proposed a specific nanosatellite mission and we have evaluated various nanosatellite constellations, using SaVi simulator, in order to identify the best constellation which satisfies mission requirements in terms of coverage and minimal number of nanosatellites. Next, XSTP (eXtended Satellite Transport Protocol) has been identified as candidate protocol for nanosatellite networks. Foremost, we implement XSTP in NS-2 simulator. The simulations were done for LEO traditional satellite network and nanosatellite constellation respectively. Finally, through analysis and simulations in NS2, we evaluated the performance of XSTP over traditional satellite networks and nanosatellite networks. Also, we were interested to compare XSTP performance to some TCP clones, in case of a high BER environment. The specific scenarios, implementations aspects and simulation approaches are presented in detail along with the respective results.

Keywords - transport protocol; nanosatellite; constellation; STP; XSTP; simulation.

I. INTRODUCTION

Traditional satellite missions are extremely expensive to design, build, launch and operate. Consequently, both the space industry and the research community have started directing their attention to missions involving many, small, distributed and inexpensive satellites. Furthermore, many space projects in universities laboratories are focused on the development of micro-, nano- and pico-satellites for both scientific and educational purposes.

New concepts arise as small satellite domain imposes itself as a particular field. Therefore, the concept of constellation became popular because of its potential to perform coordinated measurements for remote control missions and its capacity of long-term mission. A satellite constellation is a group of similar satellites, with coordinated ground coverage, that are synchronized to orbit the Earth in some optimal way. Also, formation flying mission aims to replace a large satellite with a "virtual satellite" – a cluster of smaller satellites, flying in very precise relative positions.

Making small satellites more cost-effective demands new technologies that must be certified for spaceflight. Certainly, there is a higher risk associated with uncertified technology. Thus, a small satellite mission is the best way to perform a first flight verification.

The small satellite technology has opened a new era of satellite engineering by decreasing space mission cost, without reducing the performance. However, the biggest long-term challenge for the small satellite community is to develop a robust commercial market capable of industrializing the process of building small satellites.

The proliferation of low-cost, "micro-", "nano-" and "pico-satellite" missions in low-earth orbit has presented new challenges to the research community.

The unique challenges imposed by nanosatellite networks (e.g., onboard resources, limited communications opportunities, limited bandwidth, scalability, redundancy, power availability, high-speed node mobility, the type of communication among satellites, assigning or not a separate communication channel for positioning, timing and synchronization issues) requires us to revise communication protocol design, network management, and to consider novel routing mechanisms to accomplish "more with less".

In order to identify candidate protocols that can be used or adapted for small satellite networks, we conducted a study of routing mechanisms in traditional satellite network, Ad Hoc network and sensor networks. This study is part of PERSEUS (*Projet Etudiant de Recherche Spatiale Européen*

Universitaire et Scientifique) program, launched by CNES (*Centre National d'Etudes Spatiales*) in June 2005 [1, 2]. Based on this study, XSTP (eXtended Satellite Transport Protocol) has been identified as transport protocol targeted for small satellite constellations.

The main objective of this paper is to propose a dedicated nanosatellite constellation mission and a nanosatellite constellation model. Various nanosatellite constellation configurations have been evaluated in order to identify the optimal constellation which satisfies the mission objectives. Secondly, the performance of XSTP-probing mechanism, proposed by Maged E. Elaasar in paper [3] is evaluated, through NS2 simulations, in satellite network and nanosatellite network scenarios respectively.

The reminder of the paper is organized as follows. Section II describes the mission that we envisaged for our nanosatellite constellation. Then, Section III presents the nanosatellite network model that we proposed in order to accomplish our mission. Section IV briefly explains STP and XSTP protocols, with a point on XSTP-probing mechanism. The simulation configuration, the performance metrics and the implementation solution are described in Section V. Simulation results in terms of nanosatellite constellation configurations and XSTP performance are discussed in Section VI and Section VII. Finally, Section VIII concludes the paper.

II. MISSION DESCRIPTION

Worldwide, there are a lot of unexploited regions in terms of mineral resources. Indeed, the Simpson Desert (in Australia) is rich in uranium, the Sahara Desert is rich in iron ore and salt, the Atacama Desert (Chile) is rich in iron and copper ore. Therefore, it is highly likely that in the near future, industrial companies will exploit those areas for their precious wealth.

As mentioned in paper [1], the global demand for lithium, the lightweight metal used to make high-powered batteries for cell phones, laptops, and hybrid cars, is expected to triple in the next 15 years. Fifty to 70 percent of the world's supply of this critical mineral is contained in just one place – Bolivia's Uyuni salt flats, shown in Fig. 1.

The United States Geological Survey [5] says that 5.4 million tons of lithium could potentially be extracted in Bolivia, compared with 3 million in Chile, 1.1 million in China and just 410,000 in the United States.

Therefore, we focus on the Salar de Uyuni, the world's largest salt flat desert of 10,582 square kilometers. It is located in the southwest Bolivia (Fig. 1), near the crest of the Andes, and is elevated 3,656 meters above the mean sea level.

At present, the reserves of lithium are at the centre of the attentions of several multinationals, as well as the government. The latter intends to build its own pilot plant with a modest annual production of 1,200 tons of lithium and to increase it to 30,000 tons by 2012. [6]

Comibol, the state agency that oversees mining projects, is investing about $6 million in a small plant near the village of Río Grande on the edge of Salar de Uyuni, where it hopes to begin Bolivia's first industrial-scale effort to mine lithium from the white, moonlike landscape and process it into carbonate for batteries. [17]

Figure 1. Salar de Uyuni viewed from space, with Salar de Coipasa in the top left corner.

Considering this context, we propose to deploy a nanosatellite operator that provides communications services (voice, SMS and images) for an industrial company in charge of lithium resources exploitation in Salar de Uyuni desert. It is important to mention that this small satellites system can be applied to any similar remote area. Unless stated differently, in this paper, the term "nanosatellite" means any satellite with a mass of 50 kg.

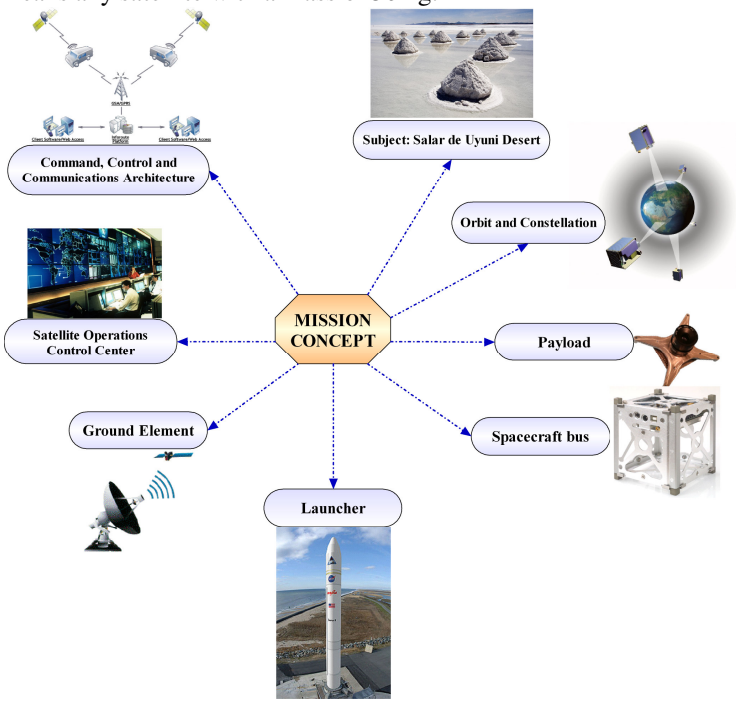

Figure 2. Mission concept basic elements.

Figure 2 presents all the elements of our mission, which implies high-level processes from mission analysis and design to cost estimation models.

Firstly, we have investigated the existing Bolivian mobile operators and theirs coverage areas. There are 3 mobile operators:

- Telefonica Celular De Bolivia S.A. (TELECEL BOLIVIA), operating within GSM850 band;
- Entel SA, operating within GSM1900 band;
- Nuevatel PCS De Bolivia SA, operating within GSM1900 band.

As seen in Fig. 3, 4 and 5, none of the operators have coverage over or close to the Salar de Uyuni desert.

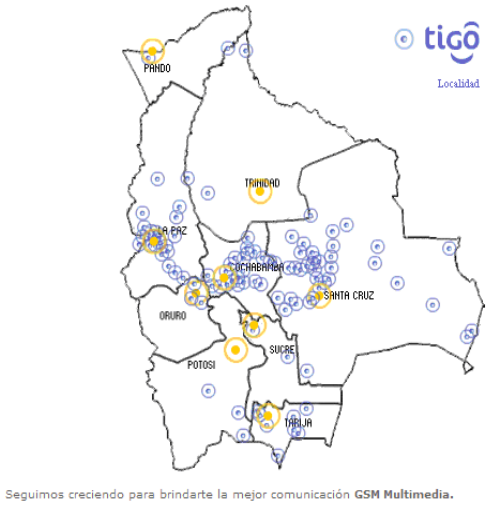

Figure 3. TELECEL BOLIVIA coverage map.

Figure 4. Entel SA coverage map (Credits : 2009 GSM Association; Europa Technologies Ltd.).

Figure 5. Nuevatel PCS De Bolivia SA coverage map (Credits : 2009 GSM Association; Europa Technologies Ltd.).

Secondly, an analysis of possible needs of the industrial company personnel yields the following requirements:

1. Continuous coverage of the target area (24h/24).
2. Mobile terminals with voice and data capabilities (e.g., voice, SMS, imaging).
3. Group Voice communications among on-site personnel.

The network architecture should be 'flexible' in that it is able to provide direct coverage to the area without having to go through a hierarchical command center.

Our system architecture is divided into three segments:

- Space segment is represented by the nanosatellite constellation;
- Ground segment is represented by Mobile Ground Station (or MGS). Based on the same principle as the i-c@r, used to provide WiFi Internet over a certain area via satellite, we can consider a similar, modified MGS, with an S-band transceiver to ensure the satellite link via a 3m wide satellite dish.
- User segment is represented by Mobile User Terminals (or MUT) with voice and data capabilities.

III. NANOSATELLITE CONSTELLATION MODEL

The design of a satellite constellation is very complex due to all the factors that need to be considered, from orbit elements to perturbations that act on each satellite.

Specifying all orbit elements for each satellite of the constellation is inconvenient and overwhelming. A reasonable approach is to begin with satellite constellation in circular orbits and at common inclination angle and altitude. In this case, the period, velocity and node rotation rate will be the same for all satellites.

The constellation size and structure has a strong impact on the system's cost and performance, so it is necessary to evaluate various constellation designs and to explain the reasons for final choice.

In paper [7], James R. Wertz states that if the regions of interest do not include the poles, then an equatorial constellation may provide all the coverage with a single orbital plane, which leads to flexibility, multiple performance plateaus and graceful degradation.

Thus, Salar de Uyuni desert is placed on 20º S latitude so, a constellation having several equatorial nanosatellites with enough altitude to provide the appropriate coverage at the smallest elevation angle (ε) is the best solution for our mission.

The formal mathematical problem definition could be written as:

Objectives: min *Ns* and max *Cov*

Constraints: Subject to

$$\textbf{Altitude } 500 \text{ Km} \leq h \leq 2000 \text{ Km}$$

$$\textbf{Minimum elevation angle } 5^{o} \leq \varepsilon_{min} \leq 30^{o}$$

Given: Latitude L = 20º S

Sun-synchronous, equatorial orbit

The purpose is to find the best constellation design which satisfies simultaneously the two mission objectives:

1. the number of nanosatellites has to be minimized;
2. the coverage of the desert has to be maximized (the nanosatellite has to stay as long as possible over the target area so, time in view need to be maximized).

We model our problem as a box with inputs and outputs. Therefore, some parameters are defined as input data for a constellation module (box) that will delivers output data. Fig. 6 illustrates key inputs and outputs of the constellation module.

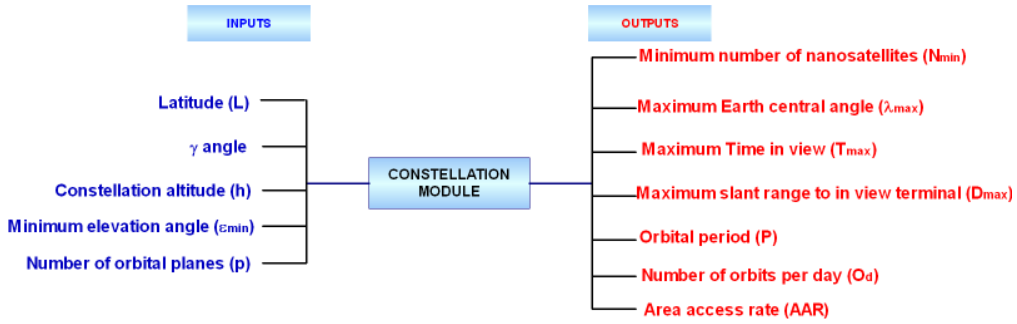

Figure 6. Inputs and outputs of constellation module.

Two vectors were defined: design vector and constants vector. The first one contains the attributes that will distinguish and differentiate alternative nanosatellite constellations. The latter contains attributes that will not differentiate alternative system architectures. For example, the latitude of Salar de Uyuni dessert is a constant value of 20º regardless of the other attributes of the architecture.

Six variables – minimum number of nanosatellites, constellation altitude, minimum elevation angle, number of satellites per orbital plane, number of orbital planes in the constellation, and maximum Time in view – make up the design vector.

Latitude (***L***), inclination angle (***i***) and ***γ*** angle are variables of constant vector.

Our nanosatellite constellation model includes some assumptions that simplify numerical calculations. We assumed that nanosatellites are placed on an equatorial, sun-synchronous LEO type orbit and it is passing near a target represented by a ground station. Also, we assume that Earth is a perfect sphere, an adequate assumption for most mission geometry applications. For precise calculation, a correction for oblateness must be applied. In our calculations, we neglected the Earth's rotation in the short period for which the nanosatellite passes over the interested area.

Figure 7 illustrates computational geometry of satellite.

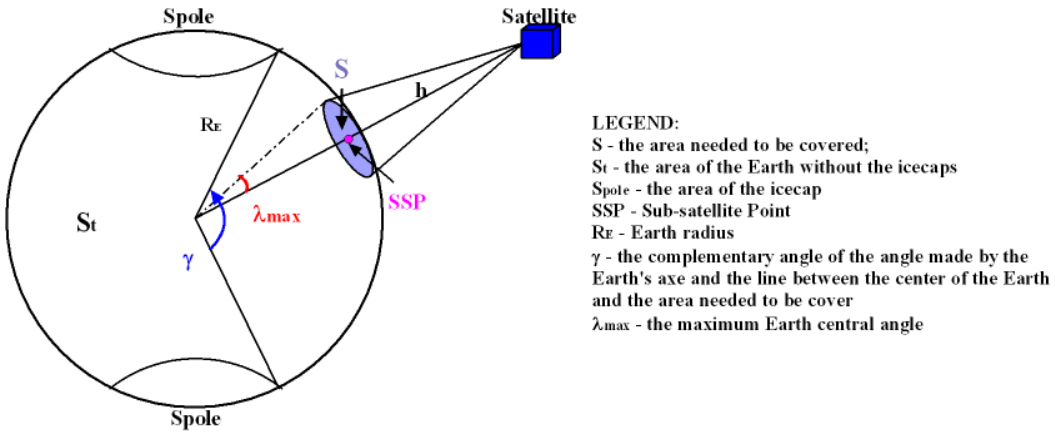

Figure 7. Computational geometry of satellite parameters.

According to Fig. 7, the area of the Earth, excluding icecaps is calculated as:

$$S_t = S_{Earth} - 2 \times S_{pole} \quad (1)$$

The Earth area is:

$$S_{Earth} = 4\pi R_E^2 \quad (2)$$

The area of every icecap is:

$$S_{pole} = 2\pi R_E^2 \times (1 - \cos\gamma) \quad (3)$$

By replacing Eq. 2 and Eq.3 in Eq. 1, we obtain:

$$S_t = 4\pi R_E^2 \times \cos\gamma \quad (4)$$

The area that needs to be covered is:

$$S = 2\pi R_E^2 \times (1 - \cos\lambda_{max}) \quad (5)$$

We estimated the minimum number of nanosatellites needed to cover the surface S as:

$$N_s = \frac{S_t}{S} \quad (6)$$

By replacing Eq. 4 and Eq. 5 in Eq. 6:

$$N_s = \frac{2\cos\gamma}{1-\cos\lambda_{max}} \quad (7)$$

As James R. Wertz states in paper [7], for communications, the satellite must be more than 5° above the horizon. In practice, we select a specific value of ε_{min} and we use this value. This parameter has a major influence on other computed parameters.

Given minimum elevation angle ε_{min}, we define the maximum Earth central angle λ_{max}, the maximum nadir angle η_{max}, measured at the satellite from nadir to the ground station and the maximum slant range D_{max} at which the satellite will still be in view. All these parameters are calculating using formulas presented in paper [15]:

$$\sin\eta_{max} = \sin\rho\cos\varepsilon_{min} \quad (8)$$

$$\lambda_{max} = arccos\left(\frac{R_E}{R_E+h}\cos\varepsilon_{min}\right) - \varepsilon_{min} \quad (9)$$

$$\eta_{max} = arcsin\left(\frac{R_E}{R_E+h}\cos\varepsilon_{min}\right) \quad (10)$$

$$\lambda_{max} = 90° - \varepsilon_{min} - \eta_{max} \quad (11)$$

$$D_{max} = R_E\frac{\sin\lambda_{max}}{\sin\eta_{max}} \quad (12)$$

The maximum time in view T_{max} for any point P on the surface of the Earth occurs when the satellite passes overhead:

$$T_{max} = P\frac{\lambda_{max}}{180°} \quad (13)$$

The orbit period P of each satellite may be calculated as a function of the constellation altitude h:

$$P = 1.658669 \times 10^{-4} \times (R_E + h)^{3/2} \quad (14)$$

The area access rate as the satellite sweeps over the ground for the target region is:

$$AAR = \frac{2K_A\times\sin\lambda}{P} \quad (15)$$

where K_A=2.55604187*10^8.

According to Eq. (7), (9) and (13), we observe some interesting variations useful for space segment dimensioning:

- the minimum number of nanosatellites is increasing as the minimum elevation angle is increasing, for the same altitude;
- for the same elevation angle, the minimum number of nanosatellites is decreasing as the altitude is increasing;
- the maximum time in view of any given point on Earth is increasing as the altitude is increasing, for the same elevation angle.

We have validated our nanosatellite network model with small satellites constellations currently in orbit. Two examples are important to be mentioned here: RapidEye constellation [15] and Disaster Monitoring Constellation (DMC) [16].

IV. STP AND XSTP OVERVIEW

This section discusses the Satellite Transport Protocol (STP) and explains the general design of XSTP protocol. There is a particular focus on XSTP-probing mechanism as it is most relevant to this paper.

A. Satellite Transport Protocol

The Satellite Transport Protocol (STP), proposed by Katz and Henderson [8, 9] is a transport protocol, which is specifically optimized for the unique constraints of satellite network environment. STP is found to outperform TCP in environments characterized by high BER, severe asymmetry and varying RTTs, typically characteristics of LEO satellite links.

Based on paper [4], the main features of STP can be summarized as follows:

- Enforcing the separation between data and control information in order to minimize the control overhead in smaller data segments;
- Mechanism that adapts to the amount of rate control required in the network, ranging from no rate control to explicit rate control. Unlike TCP, which uses a self-clocking property, STP depends on a delayed send timer to pace transmissions uniformly over the estimated RTT. The main benefit of the pacing mechanism is the reduction of the risk of introducing large bursts to the network.
- Segment type overloading for supporting a fast connection start mechanism.
- Efficient acknowledgement strategy
  STP employs an automatic repeat request (ARQ) mechanism that uses selective negative acknowledgements (NACK). By using this mechanism, only segments reported missing by receivers are retransmitted. The advantage is lower reverse link traffic when the loss is negligible and a speedy recovery when the loss is severe. In contrast with TCP, there is no RTO mechanism in STP, which makes it more resilient to RTT variations.

Finally, it is important to mention that even if STP includes many of the basic principles found in TCP, it is only functionally but not semantically equivalent to it. Unfortunately, the STP protocol inherits the congestion control bias from its ancestor protocols (i.e., TCP, SSCOP [10]). Although the protocol can efficiently recover from multiple losses in the same round trip, its error recovery tactics can negatively affect its overall performance.

B. eXtended Satellite Transport Protocol

XSTP is a software implementation of the STP protocol in the PIX (Protocol Implementation Framework for Linux) framework. [11] The protocol is used to host a new error control strategy, called XSTP-probing. Typically, XSTP protocol can be deployed on top of a network protocol (e.g., IP). The protocol provides a reliable connection-oriented byte streaming service to application protocols (e.g., FTP).

An XSTP session is composed of one lower and one upper session. Fig. 8 depicts a typical configuration for a

communication suite including XSTP. As Maged E. Elaasar explains in paper [3], when such a suite is initialized, an instance of the XSTP protocol is created, configured and then installed in the appropriate location in the protocol hierarchy. Once there, application level protocols can use the service of the protocol to manipulate XSTP sessions.

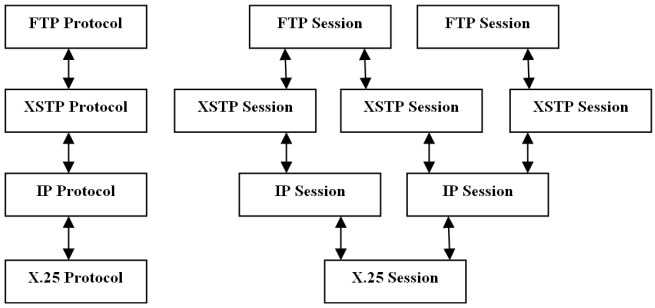

Figure 8. PIX protocol and session configurations including XSTP. [3]

An XSTP session plays double role (sender and receiver), which implies defining two new classes: an XSTP sender and an XSTP receiver. An instance of each of those classes is created in the private state of the session's object. As depicted in Fig. 9, these two instances play the sending and receiving roles of the session.

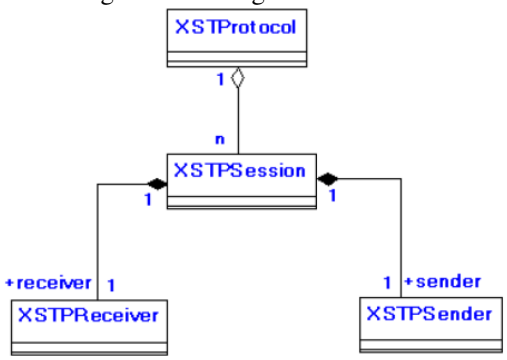

Figure 9. XSTP class diagram. [3]

C. XSTP-probing mechanism

The aim of developing the XSTP-probing mechanism was to stretch the STP protocol's ability to adapt to the different types of error found in LEO satellite networks.

According to papers [3, 4], the goal of any error control strategy is to adapt the sender's transmission rate to the varying error conditions in the network. This goal is usually accomplished by taking an aggressive attitude when the error is detected to be transient and a conservative one, when it is persistent. The XSTP-probing mechanism makes no exception to this principle.

The mechanism is triggered upon detecting a segment loss to assess the level of congestion in the network. If congestion is detected, the mechanism responds by invoking congestion control; otherwise, it resumes with *Immediate Recovery* (restoring congestion window to the same level as before probing).

Additionally, this mechanism evaluates the connection for possible error free conditions and only transmitting in those windows. As described in paper [4], it suspends new data transmission upon detecting a loss and initiates a probing cycle to collect RTT statistics on the connection. Then, it compares these RTT statistics to the RTT estimate available when the loss was discovered. It is interesting to observe that the duration of that probing cycle is proportional to the level of error in the network, which helps the connection sit out the error conditions. After the cycle is finished and if congestion is detected by proliferating RTTs, congestion control is immediately invoked. Otherwise, transmission levels are restored without taking any action. Finally, the missing segments are immediately retransmitted. Fig. 10 presents the basic algorithm of XSTP-probing mechanism.

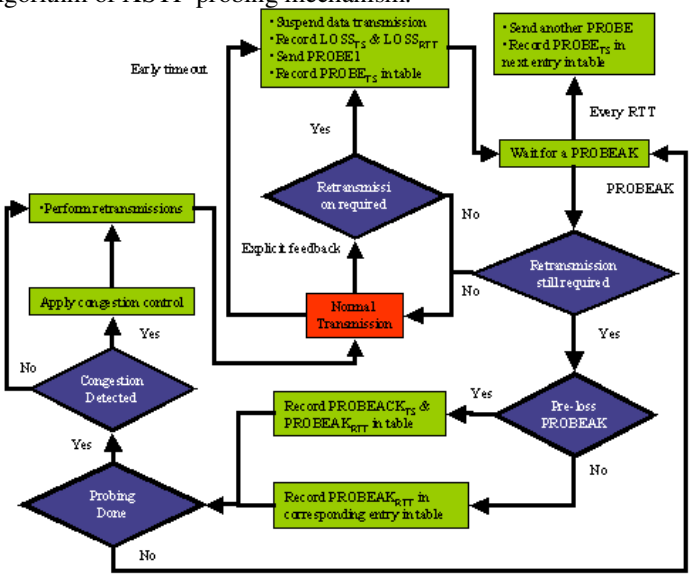

Figure 10. XSTP-probing mechanism.[3]

The XSTP–probing mechanism is implemented as a configurable option on the XSTP session. The mechanism is modeled as a class called XSTPProbing. For a more comprehensive overview of STP, XSTP and its probing mechanism, the interested reader is directed to papers [3] and [4].

V. SIMULATION CONFIGURATION

The performance of a protocol for varying network conditions and settings can effectively be evaluated using simulations. This section describes simulation environment, performance metrics and test scenarios.

A. Simulation environment

To analyze the performance of XSTP protocol, we have implemented the proposed protocol in the discrete-event network simulator NS-2 [12]. We used TCP modules corresponding to common variants of TCP (e.g., New Reno, Reno, SACK, Tahoe, Vegas), and wrote two new simulation modules for STP and XSTP.

We used SaVi simulator [13] for evaluating various nanosatellite constellations in terms of coverage area. SaVi

allows satellite orbits and coverage simulations, in two and three dimensions and is particularly useful for simulating satellite constellations.

B. Simulation scenarios

Using NS2 simulations, XSTP-probing mechanism is tested in various error conditions and performance is quantified.

We defined two scenarios – one-way communication, aiming to meet symmetric channels, and bidirectional communication for considering asymmetric channels. Each time, we quantify the performance metrics defined in Section V.E.

C. XSTP implementation solution

XSTP protocol is a derived class from STP class, the latter being derived from transport Agent class. Firstly, TCP like congestion mechanism is implemented. Then, we extended STP to XSTP by implementing the probing mechanism, described in Section IV.C, with 3 configuration parameters:

- Maximum number of trackable probe exchanges (MAX_PROB);
- Number of requested probe exchanges (REQ_PROB);
- RTT tolerance ratio (RTT_TOL).

The simulation configuration consists of 2 network nodes: source node and destination node. In the first scenario, the destination node is considered as a well of data, while in the second one, both endpoints are going to play the role of transmitter / receiver at the same time.

As Fig. 11 illustrates, we attach an XSTP agent to the source node and a STPSink agent to the destination node. Because an XSTP agent does not generate application data, we connected it to a FTP traffic generator so that we can send large data packets.

By using a background HTTP traffic generator, HTTP traffic is added for emulating the current use of WWW. The purpose was not to block the network, but to add a variability component to simulation.

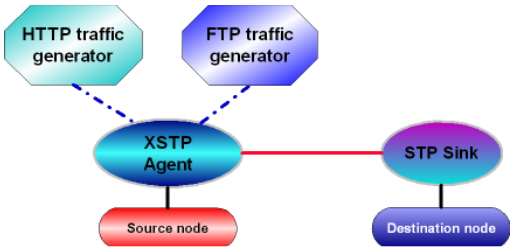

Figure 11. Nodes configuration.

The size of packages sent by the source node is 1000 bytes. The size of receiver's window is fixed to 200 and the initial size of transmitter's congestion window is 1. The maximum number of trackable probes is set to 4, and the number of consecutive RTT measurements sufficient to finish the probing cycle is set to 2. The polling frequency is set to 3 per RTT, and when the probing mechanism is triggered, the polling rate becomes 1 per RTT. The duration of every simulation is 60 seconds. The BER varies between 10^{-8} and 10^{-3}.

Due to the random behavior of the Web traffic, every simulation is repeated four times and the final results are calculated by making the average between the intermediate simulation results.

Table 1 presents the simulation parameters.

TABLE I. SIMULATION PARAMETERS

Parameter	*Value*
Sender buffer size	64 Kb
Receiver buffer size	64 Kb
MSS (maximum segment size)	1000 bytes
Maximum window size	64 segments
Sender's initial congestion window	1 segment
Maximum burst size	8 segments
Initial RTT	0.08 s

D. Simulation network topologies

For all the envisaged scenarios, we have used two network topologies:

- a satellite network (Fig. 12) based upon the LEO satellite constellation proposed by Teledesic [14];
- a nanosatellite network (Fig. 13) based on SaVi simulations and numerical results presented in Section VI.

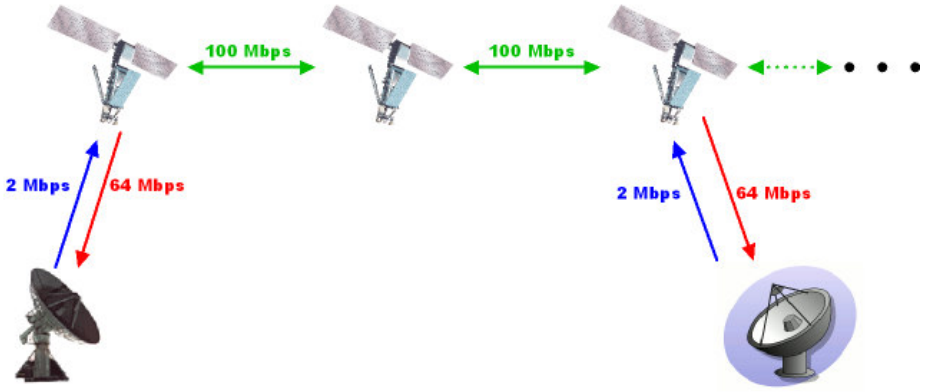

Figure 12. Satellite network model.

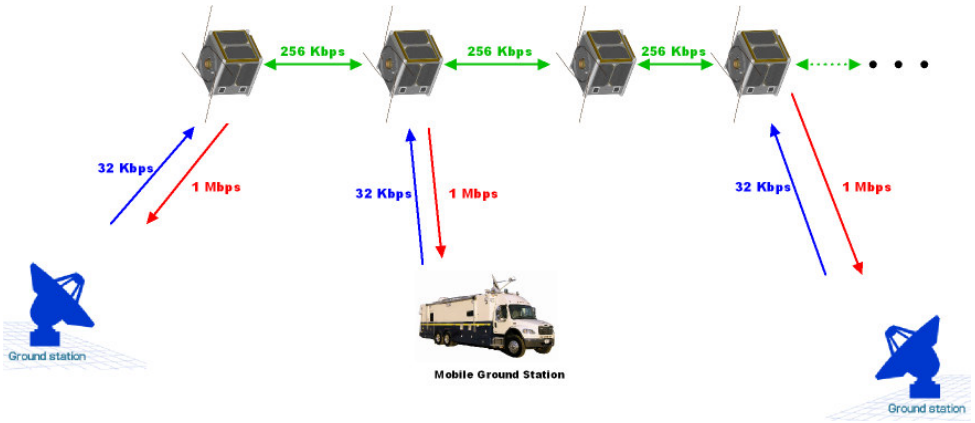

Figure 13. Nanosatellite network model

Table 2 presents the main constellations parameters, while table 3 summarizes data link parameters.

TABLE II. CONSTELLATION PARAMETERS

Parameter	*Satellite constellation*	*Nanosatellite constellation*
Nominal altitude	1375 km	1500 km
No. of planes	12	1
No. of satellites/plane	24	9
Planes separation distance in altitude	2 km	-
Nominal inclination	84.7 °	0º
Minimum Earth elevation angle	40º	15º
Orbital period	113.23 min	115.98 min

TABLE III. DATA LINK PARAMETERS

Parameter	*Satellite constellation*	*Nanosatellite constellation*
Uplink frequency	28.6 - 29.1 GHz	VHF, UHF
Downlink frequency	18.8- 19.3 GHz	S band (2-4GHz)
Uplink data rate	2 Mbps	32 Kbps
Downlink data rate	64Mbps	1 Mbps
Inter-satellite link (ISL) rate	100 Mbps	256 Kbps
ISL propagation delay	10 ms	50 ms
Ground-to-satellite link propagation delay	5 ms	15 ms

E. Performance metrics

The relevant performance metrics defined for NS2 simulations are based on paper [4]:

1) Effective throughput is defined as the average data rate (bps) as seen by the data link session and it is calculated using the following formula:

$$Effective\ throughput = \frac{original\ size}{simulation\ time} \quad (16)$$

2) Transmission overhead is defined as the percentage of extra bytes expended in the reliable transmission of the original data bytes. The transmission overhead is calculated, in %, using the following formula:

$$Transmission\ overhead = \frac{total\ size - original\ size}{original\ size} \times 100 \quad (17)$$

3) Efficiency describes the channel utilization. It is defined as the ratio between the packet original size and the total size of transmitted data:

$$Channel\ efficiency = \frac{original\ size}{total\ size} \quad (18)$$

4) Reverse channel utilization describes the backwards channel utilization. It shows the protocol efficiency on asymmetric links where the bandwidth is not the same in both directions. It is calculated using the following formula:

$$Reverse\ channel\ utilization = \frac{backward\ original\ size}{simulation\ time} \quad (19)$$

VI. SAVI SIMULATION RESULTS

This section presents SaVi simulation results in terms of coverage. Table 4 describes the parameters for four types of nanosatellite constellations simulated in our study. The numerical calculations of these parameters were made based on equations defined in Section III.

According to Table 4, we observe that constellation C_4 satisfies our mission objectives because:

- it has a minimum number of nanosatellites (9 nanosatellites);
- it offers a coverage band between 0º and 22º S latitude, thus assuring a total coverage of Salar de Uyuni Desert;
- the time in view of every nanosatellite is maximized.

Therefore, constellation C_4 has been implemented in NS2 for studying the XSTP performance over nanosatellite networks.

TABLE IV. CONSTELLATION CANDIDATES

Constellation	C_1	C_2	C_3	C_4
Minimum number of nanosatellites within the constellation (N_{min})	**18**	**14**	**14**	**9**
Coverage latitude	**0º - 18º**	**0º - 19º**	**0º - 20º**	**0º - 22º**
Maximum Time in view (T_{max})	**8.9 min**	**10.62 min**	**10.73 min**	**15.17 min**
Minimum elevation angle (ε_{min})	**15º**	**10º**	**15º**	**15º**
Number of orbital planes (N_p)	1	1	1	1
Constellation altitude (h)	800 km	800 km	1000 km	1500 km
Orbital period (P)	100.87min	100.87 min	105.11 min	115.98 min
Number of orbits per day (O_d)	14.23	14.23	13.66	12.38
Maximum slant range (D_{max})	2032 km	2367 km	2408.38 km	3258.45 km
Maximum Earth central angle (λ_{max})	15.87º	18.95º	18.38º	23.55º
Area Access Rate (AAR)	$1.38*10^6$ km^2/min	$1.645*10^6$ km^2/min	$1.53*10^6$ km^2/min	$1.76*10^6$ km^2/min
Nanosatellite velocity (v)	7.4561 m/s	7.4561 m/s	7.3507 m/s	7.1136 m/s

For Fig. 14 to 17, satellite coverage, represented in yellow/red, is intended to give an idea of the number of nanosatellites visible from a point on Earth. The higher the number of nanosatellites covering a point, the deeper the shade of red is. Also, coverage decay, illustrated in shades of blue, gives an idea of where a satellite footprint has been and is going, even when you look at a still map snapshot.

Constellation C_1 is constituted of 18 nanosatellites placed on an equatorial orbit at 800 km of altitude and having a minimum elevation angle of 15°. As seen in Fig 14, the coverage area will be between 0 and 18 ° of latitude S, but our target region is situated at 20° S latitude. Thus, this configuration does not satisfy our mission goal in terms of coverage.

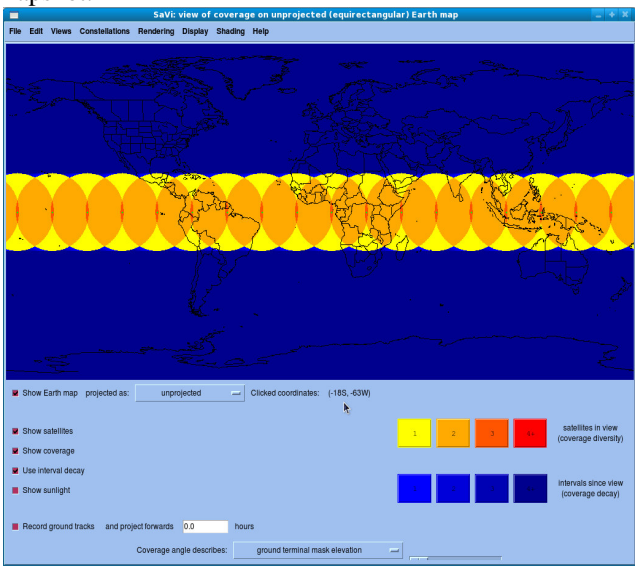

Figure 14. Coverage of nanosatellite constellation including 18 nanosatellites placed on a LEO, equatorial orbit at an altitude of 800 Km and minimum elevation angle of 15º

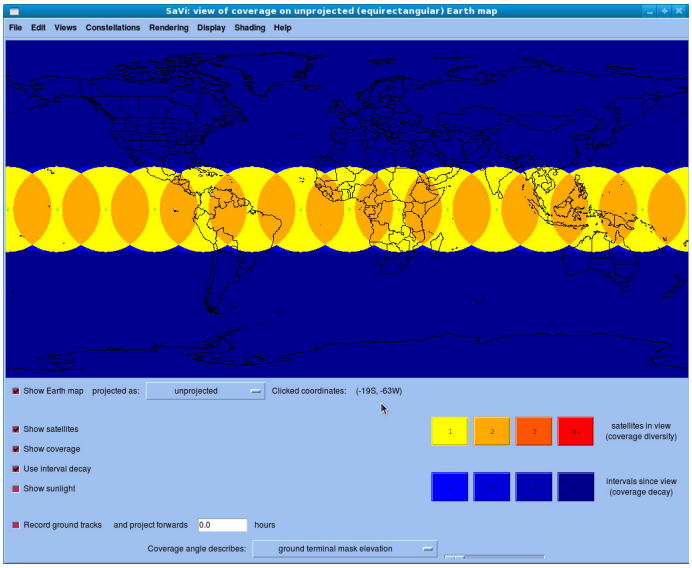

Figure 15. Coverage of nanosatellite constellation including 14 nanosatellites placed on a LEO, equatorial orbit at an altitude of 800 Km and minimum elevation angle of 10º

With the purpose of obtaining the desired coverage, we modified the minimum elevation angle of 10º and we obtained constellation C_2 of 14 nanosatellites placed at 800 km of altitude. Unfortunately, the coverage area (Fig. 15) will be between 0º and 19º S latitude, solution that still not corresponds to our mission. Additionally, this constellation might suffer of bad visibility, given the natural and manmade obstacles that would block nanosatellites at lower elevation angles out of view.

Knowing that Salar de Uyuni desert has a flat surface, we thought that a minimum elevation angle of 15º is sufficient to have visibility at any given point on the desert. Thus, a possible solution to our coverage problem seems to be constellation altitude increasing. By increasing altitude at 1000 km and for ε_{min}=15º, we obtain constellation C_3 of 14 nanosatellites (Fig. 16). This constellation defines a coverage band between 0º and 20º S latitude, solution that satisfy the second mission objective (the coverage), but not the first one (minimizing the number of nanosatellites within the constellation).

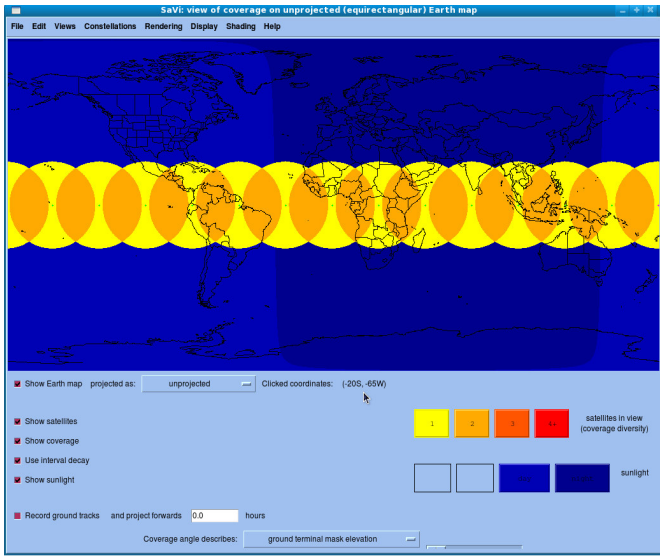

Figure 16. Coverage of nanosatellite constellation including 14 nanosatellites placed on a LEO, equatorial orbit at an altitude of 1000 Km and minimum elevation angle of 15°

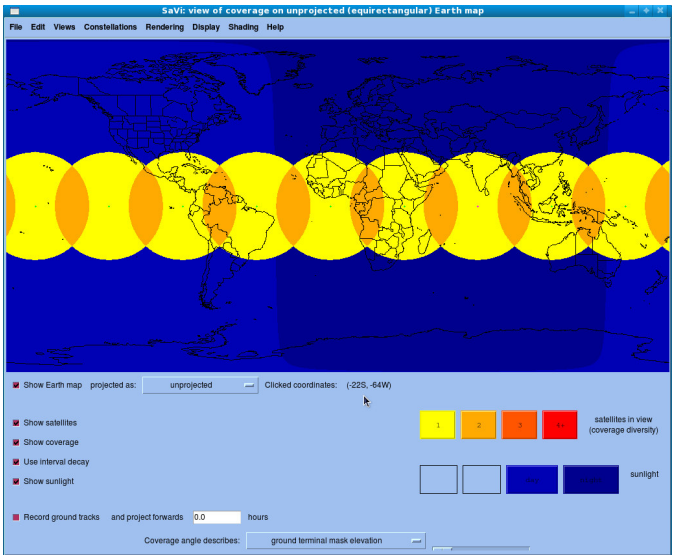

Figure 17. Coverage of nanosatellite constellation including 9 nanosatellites placed on a LEO, equatorial orbit at an altitude of 1500 Km and minimum elevation angle of 15°

Our numerical calculations have showed that the number of nanosatellites is decreasing with altitude increasing. Considering this type of variation and for minimizing the number of nanosatellites, a solution might be to increase constellation altitude to a value that satisfies our requirements. Thus, for an altitude of 1500 Km and ε_{min}=15°, we obtain constellation C_4 of 9 nanosatellites (Fig. 17), which is the best architecture that satisfy our two mission goals.

VII. NS2 SIMULATION RESULTS

In this section, NS2 simulation results for satellite network and nanosatellite network scenarios are reported and discussed. Also, we were interested to compare XSTP performance to some TCP clones, in case of a high BER environment.One-way transmission scenario

In this scenario we consider symmetric channels.

Fig. 18a and Fig. 18b illustrate effective throughput variation with respect to BER for satellite network and nanosatellite network respectively. For both networks, XSTP outperforms all TCP clones for high BER conditions (10^{-3}), mainly due to its probing mechanism. Unlike XSTP, STP and TCP clones reduce their transmission rate at every error detection.

In satellite network scenario, TCP Sack has a comparative throughput to STP and XSTP (roughly, 1400 Kbps) for low BER, but a significant difference is observed as BER increases.

Compared to satellite network case, in nanosatellite scenario, TCP Sack outperforms XSTP only for low BER environment. For high BER rates, XSTP assures 2 times more effective throughput than TCP clones. Also, TCP Vegas has the best performance among TCP clones due to its robust detection mechanism that minimize packet losses. We have also noticed that effective throughput of TCP clones is significantly reduced. The reason is that the latter make a processing operation after congestion detection, which is not the case of XSTP that makes the difference between an error due to congestion and an error due to transmission.

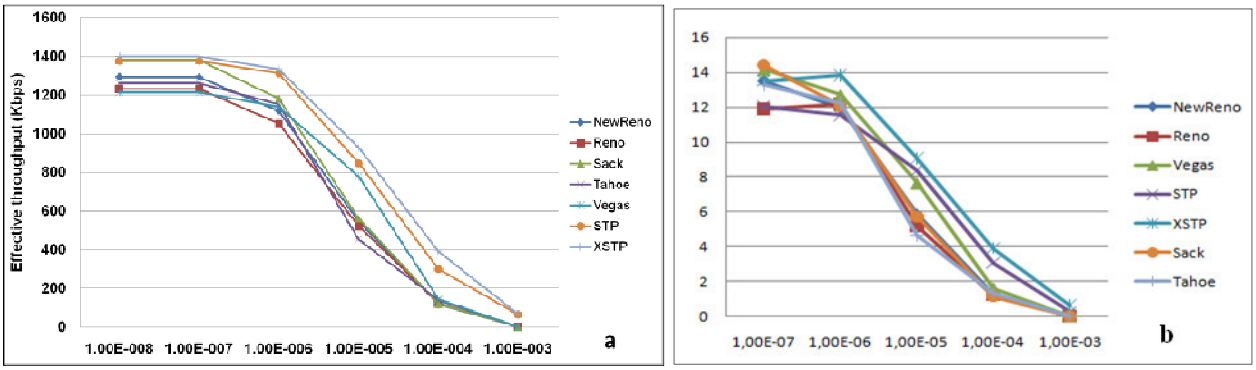

Figure 18. Effective throughput variation (one-way scenario): a) satellite network; b) nanosatellite network.

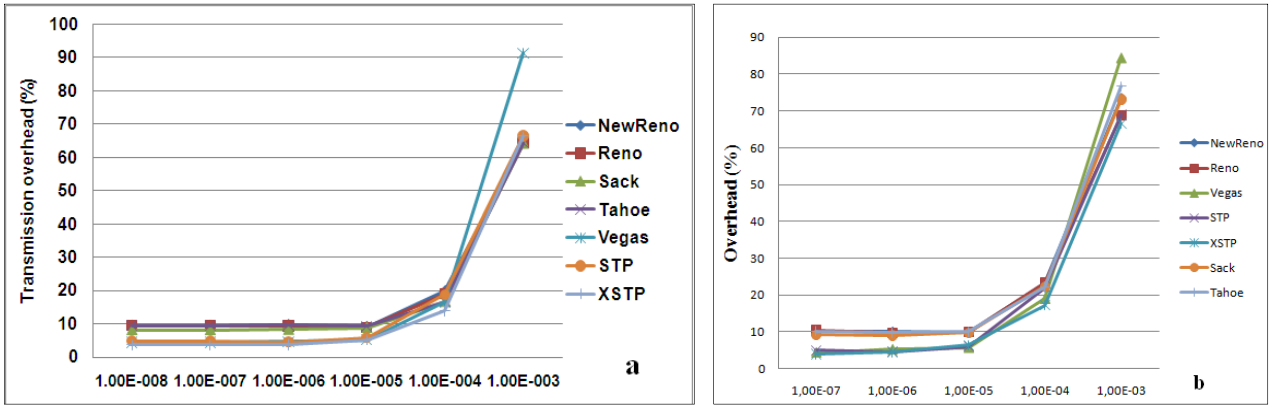

Figure 19. Transmission overhead variation (one-way scenario): a) satellite network; b) nanosatellite network.

Fig. 19a and Fig. 19b illustrates overhead variation as a function of BER for satellite network and nanosatellite network respectively.

Overhead packets are supplementary packets transmitted over the network. More the protocol is powerful less overhead packets are transmitted. Protocol designers have always tried to minimize the overhead. In the satellite networks domain, reducing overhead means minimizing the energy spent uselessly by the satellite, knowing that the energy used for overhead transmission is because the energy spent during overhead transmission is considered as lost energy. Considering this, our simulations shows that XSTP protocol consumes less energy than TCP clones for overhead transmission thus, it has more energy for data transmission.

According to Eq. (17) and Eq. (18), overhead test is complement to efficiency test. Unlike efficiency, the overhead increases with BER increasing.

During probing cycle, receiver sends one POLL per RTT and stops data transmission in order to avoid data losses, reducing this way the number of control packets. At the end of this cycle and if there is no congestion, sender doesn't reduce its congestion window; thus, it gives user the possibility to send much more data over the network.

Contrary, STP maintains 3 POLL per RTT and because it doesn't stop data transmission during polling cycle, the number of control packets increases due to successive losses. This is the raison why STP has much more overhead than XSTP.

As seen in Fig. 19a and Fig. 19b, XSTP has the lowest overhead, offering two times less than NewReno, Reno, Tahoe and Sack, in case of low BER conditions (10^{-8}–10^{-6}). We have also noticed that among TCP clones, TCP Vegas has the biggest overhead for high BER conditions and the lowest value for low BER. This means that its congestion control mechanism is better than other TCP mechanisms in a low BER environment.

One of the most important aspects in satellite networks is energy consumption. Researchers have always tried to minimize the energy spend by satellites for data transmission. Channel efficiency shows channel utilization. In case of significant amount of data user, efficiency is closed to 1, which means that channel is well used. Contrary, if efficiency is closed to 0, the channel is not well exploited.

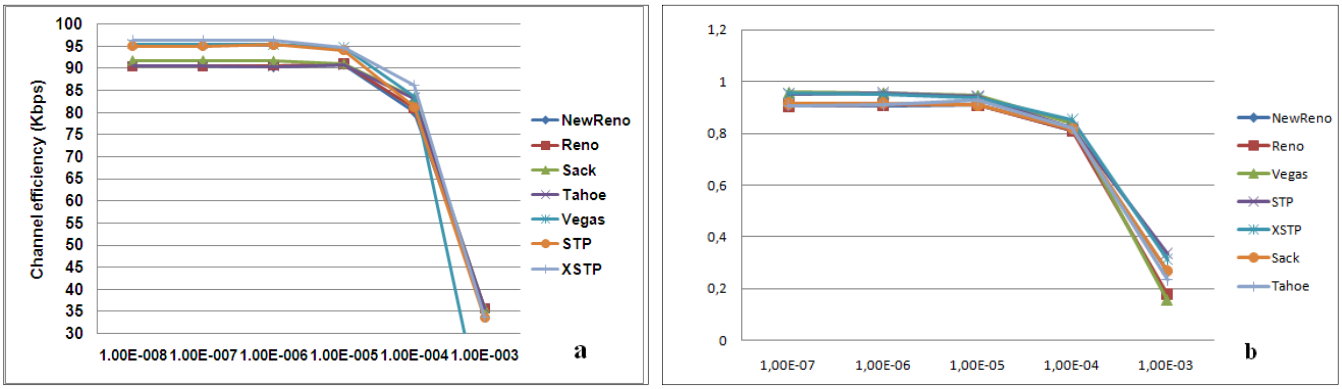

Figure 20. Channel efficiency (one-way scenario): a) satellite network; b) nanosatellite network.

According to Fig. 20a and Fig. 20b, STP and XSTP have a slightly higher performance than TCP clones for low BER conditions, thus exploiting better the communication channel. For both networks (i.e., satellite and nanosatellite), for high BER conditions (10^{-3}), TCP Vegas attains the lowest channel efficiency, which is four

times less than the other protocols. In other words, in high BER conditions, Vegas lose a lot of data packets.

Generally, reverse channel is used for ACKs transmission. Reverse channel bandwidth varies as a function of the number of ACKs transmitted over the channel, their type and size. It is important to mention that reverse channel bandwidth has to be minimized at the very most due to satellite link asymmetry.

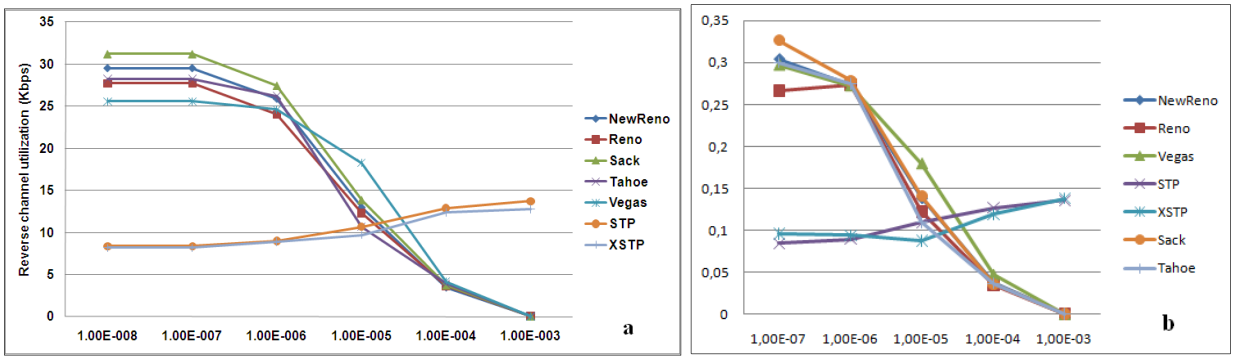

Figure 21. Reverse channel utilization (one-way scenario): a) satellite network; b) nanosatellite network.

Fig. 21a and Fig. 21b illustrates reverse channel variation with respect to BER for satellite and nanosatellite scenarios respectively. We have noticed that XSTP and STP bandwidth for reverse channel increases with BER increasing; this means that XSTP needs a low bandwidth for reverse channel. Instead, reverse channel bandwidth for TCP clones decreases with BER increasing. The explanation of these variations lies directly on every protocol's principle of using bandwidth.

In fact, TCP clones use reverse channel for acknowledgements transmission. Thus, the ACKs are sent when data packets are received, which explains why TCP clones use a lot of bandwidth in low BER conditions. In other words, if there are no losses, the reverse channel bandwidth is increasing as many packets are received. Contrary, in high BER conditions, receiver doesn't transmit many ACKs; therefore, reverse channel bandwidth decreases.

Unlike TCP clones, STP and XSTP send STAT and USTAT messages over the reverse channel. When BER is low, receiver sends many small size STAT messages that demand a low reverse channel bandwidth. For high BER, reverse channel utilization is significant due to large size USTAT messages that demand a lot of bandwidth.

An important remark is that XSTP needs a lower reverse channel bandwidth than STP because the number of STAT messages transmitted during probing cycle is decreasing as the number of POLL per RTT decreases (1 STAT message per POLL).

In case of STP, the number of POLL per RTT remains unchanged (i.e., 3 POLL per RTT) during Polling cycle. Because STP doesn't suspend transmission, it sends many USTAT messages even when BER is high.

A. *Bidirectional transmission scenario*

In this scenario, data transmission is made in both ways (a node is sender and receiver too). As compare to one-way communication case, data rate of all protocols decreases because of reverse path transmission.

XSTP effective throughput has a smoothness decrease as compare to one-way scenario.

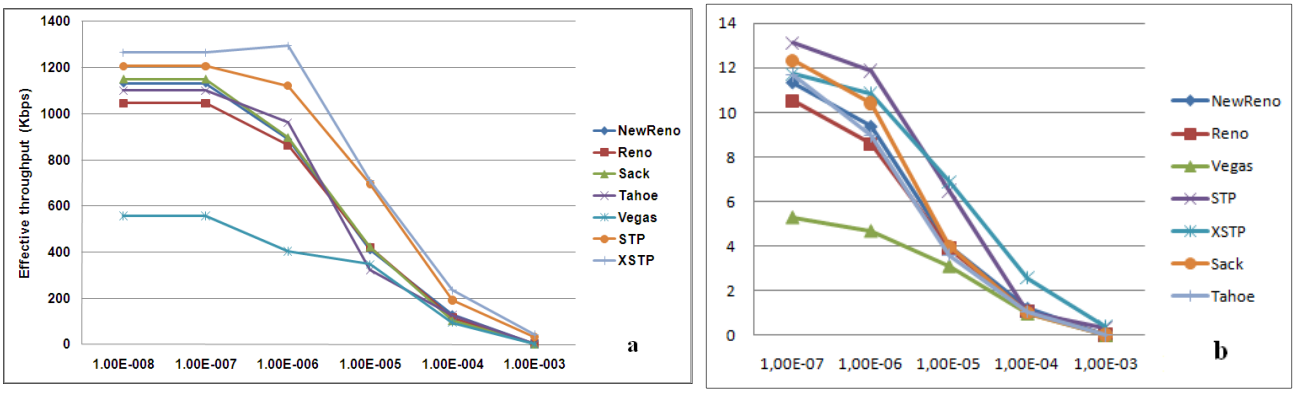

Figure 22. Effective throughput variation (bidirectional scenario): a) satellite network; b) nanosatellite network.

As seen in Fig. 22a, XSTP has a comparative performance with STP and TCP clones for low BER conditions. Instead, XSTP outperforms all TCP clones in case of high BER (10^{-3}), by offering an effective throughput almost 30 times more than TCP clones.

In nanosatellite scenario (Fig. 22b), STP attains the best performance for low BER environment (10^{-7}). For example, the effective throughput of STP is better than TCP Sack because STP doesn't use a timeout for ACKs reception.

For both network types considered in our simulations, we have also observed that TCP Vegas is strongly influenced by reverse path transmission, having, even in low BER conditions, a data rate 3 times less than the other protocols.

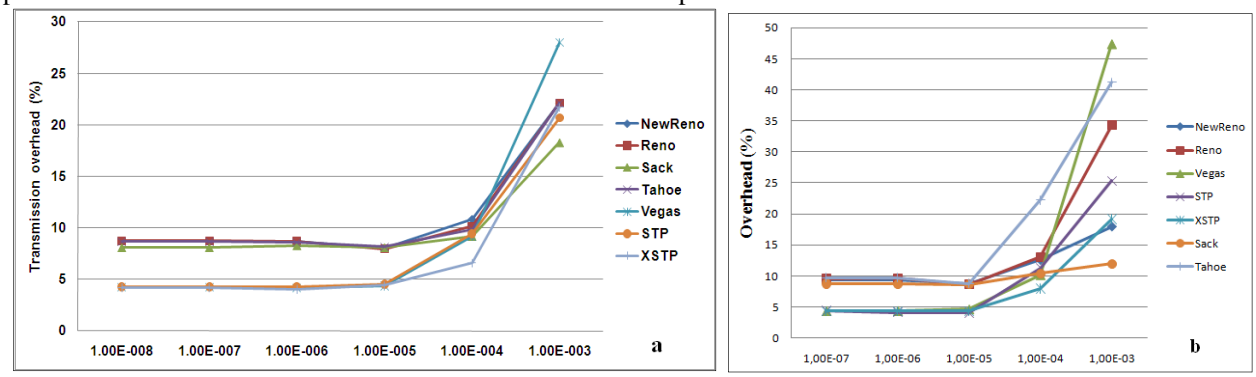

Figure 23. Transmission overhead variation (bidirectional scenario): a) satellite network; b) nanosatellite network.

In case of satellite network, by comparing Fig. 19a and Fig. 23a, we observed that XSTP offers 3 times less overhead than in one-way scenario, for high BER conditions. Also, all protocols have almost the same performance for high bit error rates, except for Vegas which still has the worst performance.

For nanosatellite scenario (Fig. 23b), STP, XSTP and Vegas attain an overhead two times less than the other protocols for low BER (10^{-7} – 10^{-5}). Instead, for high BER (10^{-3}), TCP Sack offers the lowest overhead, which is 2 times less than STP and 0.6 times less than XSTP.

The main difference is that the overhead does not increase as much as in one-way transmission when the BER is very high. This is due to the fact that all protocols decrease their transmission rate when there are other transmissions on the reverse channel.

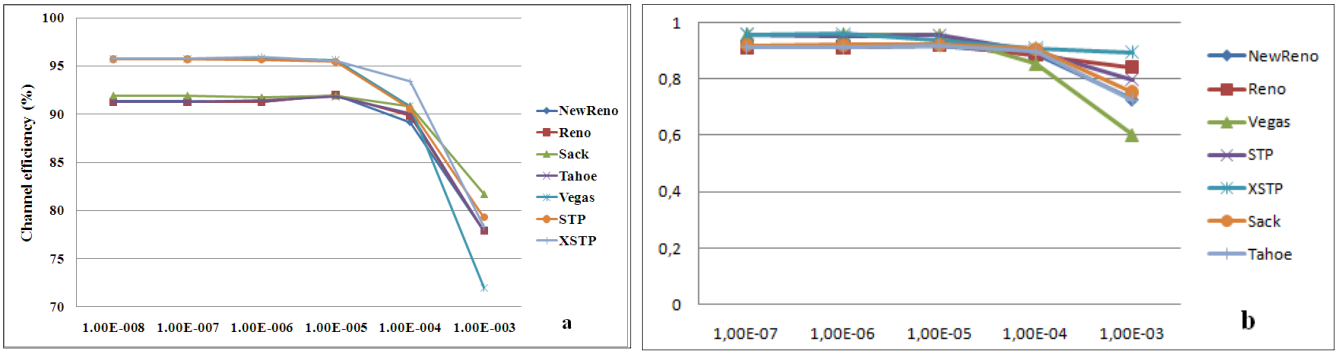

Figure 24. Channel efficiency (bidirectional scenario): a) satellite network; b) nanosatellite network.

Figure 24 presents channel efficiency variation with respect to BER for satellite network and nanosatellite network respectively. XSTP remains the best protocol with regard to STP and TCP clones.

Considering satellite network case (Fig. 20a and Fig. 24a), we observe a significant improvement of XSTP efficiency with respect to the first scenario (78% versus 33%), for high BER conditions (10^{-3}). Same trend is seen for nanosatellite network scenario (Fig. 20b and Fig. 24b), where XSTP efficiency is 3 times more in bidirectional transmission for BER=10^{-3}.

Our simulations have shown that, in high BER environment, XSTP efficiency for nanosatellite network is better than in case of conventional satellite network (90% versus 78%).

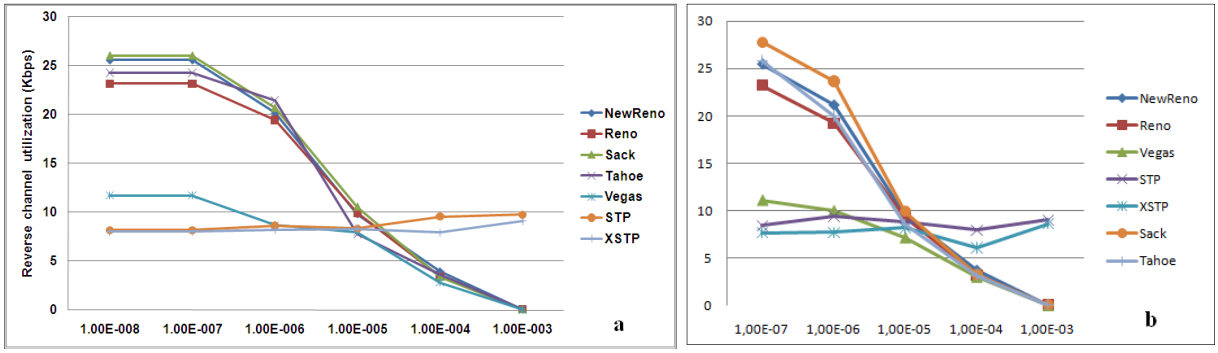

Figure 25. Reverse channel utilization (bidirectional scenario): a) satellite network; b) nanosatellite network.

Generally, all the explanations given for one-way scenario remains valid. We have noticed that the curves follow the same shape as in the one-way transmission, by regarding Fig. 21 and Fig. 25. TCP Sack uses a lot of bandwidth for reverse channel. Interestingly, TCP Vegas uses a low reverse channel bandwidth (2.83 Kbps), for high BER conditions, performing much better than STP and XSTP (Fig. 13a).

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a small satellite mission intended to deploy a nanosatellite network over a specific desert region (Salar de Uyuni Desert). Also, a nanosatellite constellation model is proposed for the envisaged mission. Several nanosatellite constellations are simulated and analyzed in order to find the best nanosatellite configuration which responds to our mission objectives.

After STP and XSTP protocol description, we presented simulation results of XSTP implementation for a conventional satellite network and nanosatellite network respectively. According to our simulations, XSTP has shown its efficiency by proving that it is useless to continue data transmission during loss detection and to invoke directly congestion control procedure, which induces a degradation of effective bandwidth. Also, XSTP attained higher effective throughput, much lower overhead, and better channel efficiency as compare to TCP clones. In spite of all these performances, XSTP protocol for satellite networks is not perfect. We propose here some future research guidelines concerning XSTP protocol. We observed that transmission overhead on the return channel is significant in high BER conditions (10^{-3}) and this need to be reduced. Also, at the probing-mechanism level, the decision principle needs to be improved in order to discriminate between congestion and other types of errors that might be found in satellite networks. Another important aspect is the energy level spent during probing cycle. An interesting research will be to find how can we measure and quantify this energy. Other future studies could be directed towards XSTP performance over other types of topologies (i.e., Flower constellation, clusters, hybrid constellation – conventional satellites and nanosatellites). Another proposal is a comparison study between XSTP probing and TCP probing mechanisms, considering that both protocols can be configured with similar set of parameters as in our survey. This comparison might show the most effective mechanism in terms of adaptation to various satellite links errors. Finally, probing mechanism could be studied in wireless communication context or in a similar domain characterized by various types of communications errors.

REFERENCES

[1] http://www.perseus.fr/presentation.php; March 30th, 2010

[2] http://www.cnes.fr/web/CNES-fr/6115-communiques-de-presse.php?item=1313; March 30th, 2010

[3] Maged E. Elaasar – "XSTP: eXtended Satellite Transport Protocol", Master Thesis, Ottawa-Carleton Institute for Computer Science, Carleton University, Ottawa, Canada, January 8th, 2003

[4] Maged E. Elaasar, Zheyin Li, Michel Barbeau, and Evangelos Kranakis – "The eXtended Satellite Transport Protocol: Its Design and Evaluation", 17th Annual AIAA/USU Conference on Small Satellites

[5] http://www.nytimes.com/2009/02/03/world/americas/03lithium.html; September 5th, 2010

[6] http://en.wikipedia.org/wiki/Salar_de_Uyuni; July 8th, 2010

[7] James R. Wertz, Wiley J. Larson - "Space Mission Analysis and Design", 3rd edition, 2007, Space Technology Library, Microcosm Press and Springer

[8] Thomas R. Henderson and Randy H. Katz - "Transport Protocols for Internet-Compatible Satellite Networks", IEEE Journal on Selected Areas of Communications, 1999

[9] R. Katz – "Satellite Transport Protocol", PhD thesis, Dec. 1999

[10] T. Henderson and R. Katz – "Satellite transport protocol (STP): An SSCOP-based transport protocol for datagram satellite networks", Proceedings of 2nd Workshop on Satellite-Based Information Systems, 1997

[11] M. Barbeau – "Protocol implementation framework for Linux (PIX), Technical report, 2002

[12] http://www.isi.edu/nsnam/ns/; February 5th, 2010

[13] http://savi.sourceforge.net/; September 8th, 2010

[14] http://web.archive.org/web/19981206074614/www.teledesic.com/tech/details.html; September 7th, 2010

[15] http://www.rapideye.de/; September 8th, 2010

[16] http://www.dmcii.com/; September 8th, 2010

[17] http://www.landcoalition.org/cpl-blog/?p=1387; September 8th, 2010