Design and Performance Analysis of CZML-IPSec for Satellite IP Networks

Zhan Huang and Xuemai Gu

Communication Research Center, Harbin Institute of Technology,
150001, Harbin, P.R. China
robbiehwang@yahoo.com.cn

Abstract. This paper analyzes the conflict between performance enhancing technology and IPSec in satellite IP networks, and proposes a solution called multilayer IP security with changeable zone (CZML-IPSec). It enables licensed intermediate nodes not only access TCP header, but also object links of upper layer in the form of HTML by converting static zone mapping to changeable dynamic mapping and building up composite security association correspondingly. A prototype is implemented to demonstrate the practical feasibility of CZML-IPSec. Measurements and performance analysis indicate that CZML-IPSec does not add unacceptable bandwidth overheads and delay, and it does not increase substantially processing hardware requirements. CZML-IPSec can help satellite IP networks provide both end-to-end security and performance enhancement.

Keywords: satellite IP networks, performance enhancing proxy, CZML-IPSec, HTTP accelerating proxy.

1 Introduction

Satellite channels are characterized by long propagation delays, large bandwidth-delay products and high bit error rates. These unfriendly features lead to TCP protocol performance degradation for satellite end-to-end reliable transmission in Internet application, which is widely utilized in terrestrial networks nowadays. Since TCP has been widely adopted by Internet hosts, many solutions have been proposed to overcome the problems of TCP over satellites, of which TCP performance enhancing proxy and HTTP accelerating proxy are comparatively efficient [1][2][3][4].

With the proposal of IP Security and adoption by the IETF, there are more and more IPSec-support services in Internet. IPSec is the most general way to supply end-to-end security at network layer for future full IP wireless networks. Therefore, it is considered suitable for data confidentiality and authentication in satellite IP networks. However, monolithic IPSec conflicts with TCP PEPs and HTTP accelerating proxies. To solve the problem, we analyze the collision among TCP PEPs, HTTP accelerating proxies and IPSec, propose the limitation of previous research, and present a compatible Multilayer IP Security with Changeable Zone (CZML-IPSec) scheme. The feasibility is validated by implementation. It proves that CZML-IPSec only adds marginal overhead and processing delay, and it can provide both network layer security and performance improvement for satellite IP networks.

K. Li et al. (Eds.): NPC 2007, LNCS 4672, pp. 277–286, 2007.

2 Analysis on IPSec Implementation in Satellite IP Networks

2.1 Conflicts Between IPSec and Performance Enhancement

IPSec is a standard mechanism for providing secure communications over the public Internet and its end-to-end model suits well in networks employing layer-architecture-packet like IP networks. However, IPSec's end-to-end mechanism conflicts with performance improvement methods in satellite IP networks. TCP PEPs and HTTP accelerating proxies operate on state information in IP header (IP destination and source addresses), TCP header (sequence number and flow identification), and upper layer data (objects following HTTP header). When TCP sessions are transmitted under the protection from IPSec ESP transport tunnel mode, in case of employing TCP snooping proxies, it is impossible to access TCP header and any information of upper layer data which are encrypted inside the ESP header at any intermediate nodes. If tunnel mode is in use, even original IP header can't be accessed. As far as TCP split connection proxies are concerned, IPSec transmission have to be split into two segments and IPSec packets ought to be checked and decrypted entirely at proxies, and then re-encrypted. In this case, a great deal of complicated data encryption and verification processing leads to loss of networks throughout. Furthermore, all data would exist in state of plaintext at one time or another. This is usually unacceptable by user's security policy. Similarly, HTTP accelerating proxies can't coexist with IPSec.

2.2 Related Solutions

The purpose of the study is to solve the conflicts between performance improvement and IPSec in satellite IP networks. So, we analyze the limitations of four existing related solutions.

2.2.1 Replacing IPSec with a Transport Layer Security Mechanism

Transport layer mechanism contains Secure Sockets Layer (SSL), Transport Layer Security (TLS), and so forth. They only encrypt TCP payload, so as to enable intermediate nodes to access plain TCP header and realize TCP performance enhancing function. However, permitting the entire TCP header to appear in plaintext exposes several vulnerabilities of the TCP session to a large amount of TCP protocol attacks, because the identity and transmission ports of sender and receiver would be visible without confidentiality protection. Additionally, SSL/TLS works only on TCP, but not on user datagram protocol (UDP), thus, the range of applications is more restricted than IPSec. Furthermore, HTTP accelerating function can't be implemented in this instance, because HTTP object zone is still encrypted.

2.2.2 Employing Transport Layer Friendly ESP (TF-ESP)

Transport layer friendly ESP format is proposed by Bellovin from AT&T Research [6]. It modifies the original ESP header to include some TCP header state information such as sequence numbers and flow identifications, in a disclosure header outside the encryption scope (but authenticated) for snooping purpose. However, TF-ESP lacks suitable integrity check verification protection, and disclosure state information is

easy to misuse by untrustworthy intermediate nodes. In addition, TF-ESP has not enough flexibility to support all of the upper protocols.

2.2.3 Tunneling a Transport Layer Security Mechanism Within IPSec

It is likely to adopt tunneling SSL/TLS within IPSec, in which SSL/TLS protects TCP payload segment and IPSec takes charge of protecting TCP header. Intermediate nodes can process TCP header related security affairs, and TCP performance enhancing can be realized accordingly, nevertheless TCP payload should be encrypted, authenticated, and decrypted twice, because IPSec treats TCP header and TCP payload as a whole. This is obvious an unnecessary waste of limited satellite networks resources.

2.2.4 Multiple Layer IPSec (ML-IPSec)

ML-IPSec improves standard IPSec by encrypting and authenticating data according to different zones on the basis of IP layers. Zone mapping and zone lists are defined in a composite security association (CSA). The former defines the coverage of zones, while the latter list all SAs. Compared with above three solutions, ML-IPSec is compatible with TCP snooping proxies, supplies limited, controllable access to upper layer data and supports IPSec end-to-end property. However, static zone mapping and inflexible CSA of ML-IPSec become the serious obstacles. In practical networks, IPSec zone coverage should be modified with the change of IP header and TCP header option's length, and encryption and authentication algorithm should be flexible according to the security requirement of the payload data. ML-IPSec couldn't function well, if the end nodes change the use of TCP or IP option, as the offsets and lengths of the headers change, causing the zone mapping to fail. ML-IPSec can't support HTTP accelerating proxies in addition, because it is not aware of the variable length of object links' information correctly in advance, in respect that the zone length is fixed in CSA.

Considering the limitation of above solutions, we propose multilayer IPSec with changeable zone (CZML-IPSec) by designing a novel IPSec ESP header and constructing a more reasonable composite security association. It enable TCP/IP header and TCP payload deserve efficient protection, meanwhile, TCP header and HTTP object links of variable length can be accessed by licensed TCP PEPs and HTTP accelerating proxies so that the performance of satellite IP networks is improved. In the following parts, the principle, design and implementation would be described in detail.

3 Design and Analysis of CZML-IPSec

3.1 Principle of CZML-IPSec

CZML-IPSec is characteristic of flexible dynamic zone mapping and modified composite security association (CSA). It enables each data packet has individual zone mapping to apply selective protection in different zones with different keys and algorithms. Zone mapping is defined in ESP header rather than SA, and transmitted within IP datagram. In the process of transmission, CZML-IPSec can adjust zone

quantity and zone coverage according to the change of data information, security levels and user's security policies. A model of CZML-IPSec transport mode protection is given in Fig. 1, in which TCP payload is divided into two parts: HTML object links (length variable) and HTML basic page. HTML object links and TCP header make up of the first zone for TCP PEPs and HTTP accelerating proxies accessing at the intermediate nodes, while HTML basic page part is the second zone unavailable for any intermediate node.

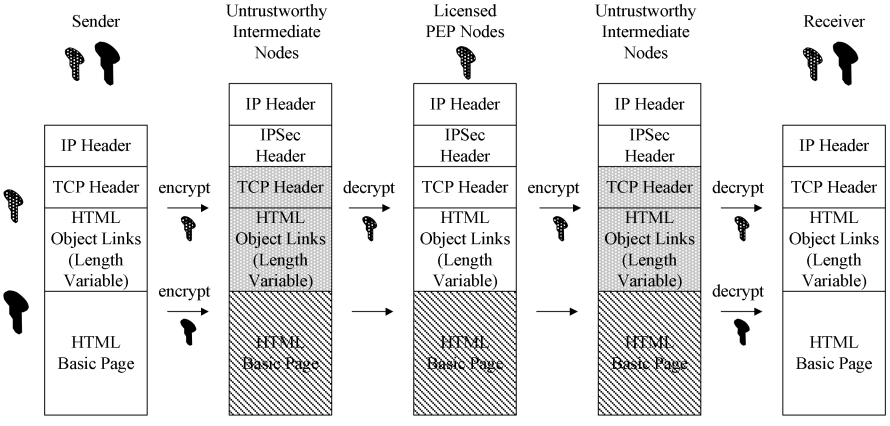

Fig. 1. CZML-IPSec transport mode protection model for TCP/HTTP

3.2 Design of CZML-IPSec

3.2.1 Dynamic Zone Mapping

Fig. 2 shows a two-zone CZML-IPSec ESP header format. Zone mapping information (4 octets) is located in the ESP header instead of original SA for dynamic mapping. The 4 octets contain zone quantity (1 octet), zone serial number (1 octet), and the length of the zone (2 octets). The zone quantity of IP datagram can be no more than the quantity in the CSA of the end nodes, because when processing bursting data of high security level, networks throughout would rather be lost to a certain extent than permission of any one of the intermediate nodes accessing HTTP segment or TCP header of the IP datagram under CZML-IPSec protection. In some special cases, CZML-IPSec can be transformed to original IPSec smoothly in the state of single zone. Zone serial number has the function of index, and it must conform to the SA number in CSA. Zone mapping information exists as plaintext in the entire process of IP datagram transmission, but it must be authenticated.

3.2.2 Composite Security Association

CZML-IPSec requires much more complex security relationship among sender, receiver and licensed intermediate nodes. So, CSA should be built to define the relationship for every zone of IP datagram. Zone mapping information is removed from CSA, and parameters are classed to two segments: mutual segment and individual

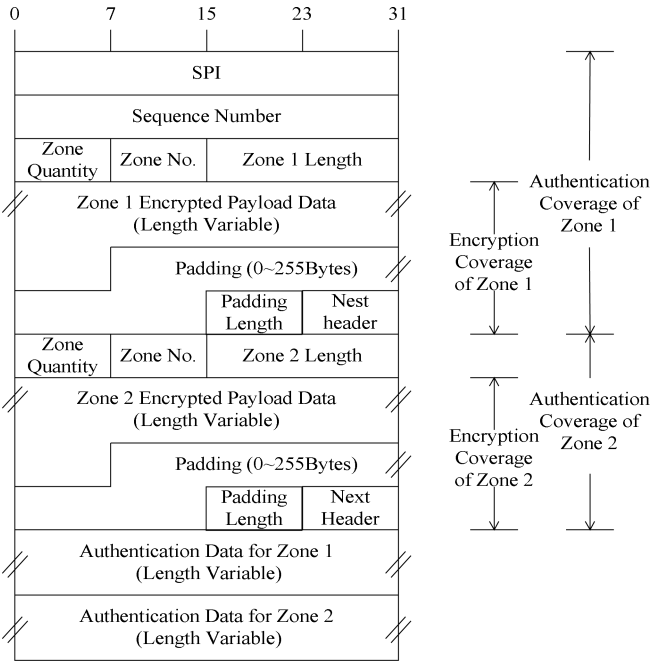

Fig. 2. Two-zone CZML-IPSec ESP header

segment. There are sequence number counter, anti-replay window, sequence counter overflow and security mode in the former one. The latter one consists of individual SA lifetime, keys of the encryption and authentication algorithms. Upon receipt of a packet contain ESP header, the receiver determine the appropriate SA based on destination IP address, SPI value and security protocol. Besides above three elements, zone serial number participates in CSA lookup for zone indexing.

3.2.3 Overhead Analysis

Overhead introduced by CZML-IPSec consists of the increase of IP data packet's length and processing loads. They are brought on by extra header information, multiple data padding and integrity check values. Zone mapping information adds 4 Bytes on the ESP header of CZML-IPSec. Individual encryptions and authentication for different zone lead to definite expansion on IP data packets due to separate padding and synchronization data. For example, a two-zone CZML-IPSec packet employing HMAC-MD5-96 would increase 12 Bytes except for data padding.

We assume the TCP/IP datagram contains two zone, 20-Byte IP header and 20-Byte TCP header without any option, the length of the HTML object links is m Bytes, and the length of the HTML basic page information is n Bytes. We calculate the overhead of IP datagram respectively under the CZML-IPSec protection of transport mode and tunnel mode and compare it with the origin IP and IPSec. Rijndael-CBC is assumed for encryption, and HMAC-MD5-96 is assumed for authentication. The results of six cases are summarized in Table 1 and Table2.

Table 1. Comparison on data packet length (Bytes)

	Transport Mode	Tunnel Mode
Origin IP	$40 + m + n$	$40 + m + n$
IPSec	$76 + \lceil (m + n + 6)/16 \rceil \times 16$	$92 + \lceil (m + n + 10)/16 \rceil \times 16$
CZML-IPSec	$112 + [\lceil (m + 6)/16 \rceil + \lceil (n + 2)/16 \rceil] \times 16$	$128 + [\lceil (m + 10)/16 \rceil + \lceil (n + 2)/16 \rceil] \times 16$

Table 2. Data packet length overhead (Bytes)

	Transport Mode	Tunnel Mode
Origin IP → IPSec	[27,42]	[47,62]
Origin IP → CZML-IPSec	[50,80]	[70,100]
IPSec → CZML-IPSec	20□36 or 52	20□36 or 52

It is concluded from Table 1 and Table 2 that, CZML-IPSec adds 36-Byte overhead on average compared with IPSec. CZML-IPSec adds 6.7%, if the average length of IP data packet is 536 Bytes. Employing HMAC-MD-32 at licensed intermediate nodes can bring overhead increase down to 5.3%. If Triple-DES-CBC (64-bit block) substitutes for Rijndael-CBC (128-bit block), the overhead increase is down to 3%.

4 CZML-IPSec Testbed Setup and Implementation

The testbed we setup for implementation of CZML-IPSec is shown in Fig. 3. CZML-IPSec is implemented in transport mode on Linux OS (kernel version 2.6) by modifying functions on free software FreeS/WAN (version 2.03). FreeS/WAN provides a base IPSec implementation on the base of which we can realize our design. We remove Linux kernel IPSec module, because it has combined original IPSec since version 2.5.47.

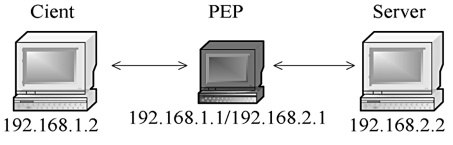

Fig. 3. CZML-IPSec transport mode testbed

According to the principle of CZML-IPSec, We modify ESP header's data structures at KLIP part in FreeS/WAN, and realize CSA as shown in Fig.3. Apache Web Sever is built up at the server for HTTP service. However, we define the length of HTTP object links information by external commands instead of actual length for flexibility. The length parameter is transmitted to the zone mapping information of the ESP header as formal variable. So, the zone length can be changed flexibly so as to gain the relevant test results conveniently. Rijndael as AES winner and RC5 are introduced as encryption algorithms besides DES and Triple-DES still employed. RC5, developed by RSA Lab, is a fast symmetric block cipher fit for hardware or software implementations. The more rounds in RC5, the higher level of security is gained. The performance of RC5 is directly proportional to the quantity of the round, and is not affected by the key size. This flexibility of RC5 makes it ideal for our implementation of CZML-IPSec, as the tradeoff between speed (and hence throughput) and security can be balanced via appropriate parameter settings. A word size of 32 bits, a round number of 12, and a key length of 16 Bytes are recommended by RSA as the nominal choice of parameters for RC5. Manual keying is used for key exchange. It is assumed that TCP/IP datagram contains two zone, 20-Byte IP header and 20-Byte TCP header without any option.

For the purposes of timing measurements, the time stamp counter on Pentium chip is introduced. RDTSC (Read Time Stamp Counter) is a two-Byte assemble instruction, which returns the number of CPU clock cycles. Therefore, by reading this counter twice we can compute the number of cycles that have elapsed between the first and the second call. This provides us with a much more accurate and meaningful measurements of time, which are independent of the CPU clock speed. For TCP snooping proxies, we improve networks monitoring and analyses tool TCPdump to implement snooping function under Linux OS, and we enable it to access zone mapping length and check integrity in HMAC-MD5-96. So, we can calculate CPU clock cycles elapsing in authenticating and decrypting zone 1 of the inbound IP datagram under the protection of CZML-IPSec at intermediate nodes. All of the programs are complied by gcc3.4 with -O2 option. Each test section is repeated 10 times, and the results are averaged.

5 Test Results

5.1 IPSec Outbound Test Results

The performance of original FreeS/WAN with Rijndael and RC5 is tested firstly. Fig. 4 shows CPU clock cycles occupied by processing IPSec outbound through FreeS/WAN. The test involves eight combinations of encryption algorithms DES, Triple-DES, Rijndael, RC5, and integrity check algorithm HMAC-MD5, HMAC-SHA1.

As shown in Fig. 4, RC5 (32/12/16)-CRC + HMAC-MD5 and Rijndael-CBC + HMAC-MD5 are better choices with less CPU cycles. When data plaintext length is 64 Bytes, the advantage of RC5 +HMAC-MD5 is most obvious, 966.6 cycles fewer than Rijndael-CBC + HMAC-MD5. The block of RC5 is 32 Bytes, which make the padding overhead fewer than Rijndael. Taking flexible encryption parameter and processing speed into account, RC5-CBC + HMAC-MD5 is most suitable for Zone 1

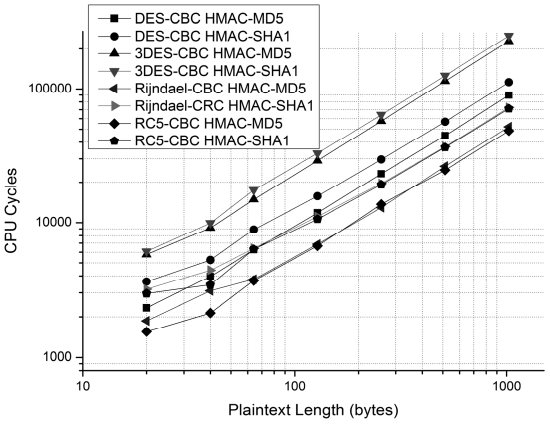

Fig. 4. CPU clock cycles occupied by processing IPSec through FreeS/WAN2.03

of CZML-IPSec with short and variable zone mapping. Hence, in the following test, it is adopted in Zone 1. Rijndael-CBC and HMAC-MD5 is used in Zone 2.

5.2 CZML-IPSec Overhead at End Node

It is assumed that TCP/IP datagram contains two zone, 20-Byte IP header and 20-Byte TCP header without any option. CPU clock cycles occupied by processing a single IP datagram with different zone mapping lengths under the protection of CZML-IPSec is listed in Table 3, where m is the length of the HTML object links in Byte, and n is the length of the HTML basic page information in Byte. The CPU cycles occupied by processing a single CZML-IPSec packet is 64.5 per Byte at most. It means that it requires CPU clock frequency at least $C\times64.5/8 = C\times8.225$MHz for a throughout of C Mbps CZML-IPSec connection. The test is under preshared manual keying, without considering key exchanging and key update, and CPU clock cycle is measured on Pentium chip, thus the estimation is not extremely accurate. So,

Table 3. CPU cycles occupied by processing single CZML-IPSec data packet in different zone length (Bytes) cases

m \ *n*	20	40	64	128	256	512	1024
0	3416.6	4678.1	5419.6	8422.3	14548.0	28012.1	53884.9
20	5116.6	6378.1	7119.6	1012 2.3	16248.0	29712.1	55584.9
40	5827.3	7128.0	7624.3	10916.3	16697.9	30576.2	56181.2
64	7164.2	8242.1	9081.7	11959.5	18211.2	31701.8	57714.4
128	10014.5	11121.2	12074.1	15685.8	20994.2	34679.9	60852.0
256	16382.3	18094.9	18813.5	22514.5	28672.4	41354.8	67832.6
512	28464.6	29131.3	29744.9	33636.3	38743.6	52372.1	
1024	52938.0	53672.2	56759.4	59473.8	62624.5		

C×12MHz as the CPU clock estimation is recommended in practice. CPU clock of no less than 1.2GHz is required for 100Mbps CZML-IPSec transmission.

5.3 CZML-IPSec Overhead at Intermediate Nodes

Snooping CZML-IPSec packets at intermediate nodes, includes checking integrity, decryption, and analyses of Zone 1 by modified TCPdump. It is assumed that RC5-CBC + HMAC-MD5 are applied in Zone 1. CPU clock cycles occupied by processing are acquired by overhead test. Fig. 5 gives the test results, where *l* is the length of Zone 1, and the length of HTML object links is 20, 40, 64, 128, 256,512 Bytes.

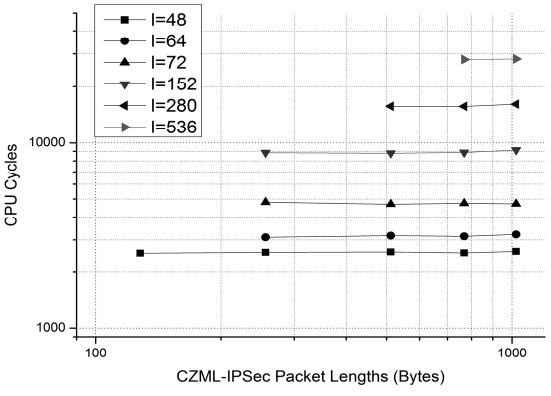

Fig. 5. Overhead measurement results at intermediate node

6 Conclusion

In satellite IP networks, the conflict is serious between end-to-end IPSec mechanism and performance enhancement techniques. In this paper, we have presented a multiple layer IPSec with changeable zone (CZML-IPSec) by designing a novel ESP header structure and constructing a more reasonable composite security association. It enable TCP/IP header and TCP payload deserve efficient protection, meanwhile, TCP header and HTTP object links of variable length can be accessed by licensed TCP PEPs and HTTP accelerating proxies so that the performance of satellite IP networks is improved. The overhead of CZML-IPSec is 6.7% greater than IPSec if Rijndael-CBC + HMAC-MD5-96 are employed. The feasibility is validated through testbed establishment and implementation. Moreover, it indicates how much overhead is added by CZML-IPSec module. It is concluded that CZML-IPSec can provide both network layer security and performance improvement for satellite IP networks. Future work would focus on automatic multiple key exchanges corresponding to CZML-IPSec, now that keys in this paper are preshared and set mutually.

Acknowledgments. This paper is supported by the grants from the National Natural Science Foundation of China [Project No. 60532030].

References

1. Border, J., Kojo, M., Griner, J.: Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, RFC 3135 (June 2001)
2. Balakrishnan, H., Padmanabhan, V., Seshan, S.: A comparison of mechanism for improving TCP performance over wireless links. In: IEEE/ACM Trans. Networking, December 1997, pp. 756–769 (1997)
3. Ehsan, N., Liu, M., Ragland, R.: Evaluation of Performance Enhancing Proxies in Internet over Satellite. Wiley Int'l. J. Commun. Sys. 16, 513–534 (2003)
4. Roy-Chowdhury, A., Baras, J.S., Hadjitheodosiou, M.: Security Issues in Hybrid Networks with A Satellite Component. Wireless Communications 12(6), 50–61 (2005)
5. Kent, S., Atkinson, R.: IP Encapsulating Security Payload (ESP). IETF, RFC 2406 (1998)
6. Bellovin, S.: Transport-friendly ESP (or layer violations for fun and profit). In: The Panel Talk 1999 Network Distributed System Security Symp. (NDSS' 99), February 1999, San Diego CA (1999)
7. Zhang, Y.: A Multilayer IP Security Protocol for TCP Performance Enhancement in Wireless Networks. IEEE Journal on Selected Areas in Communications 22, 767–776 (2004)
8. Ciccarese, G., DeBlasi, M., Patrono, L., Marra, P., Tomasicchio, G.: An IPSEC Aware TCP-PEP for Integrated Mobile Satellite Networks. In: IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications, May 2004, vol. 7, pp. 453–459 (2004)
9. Thompson, I.R., Waller, J.: Performance Enhancing Proxies and Security. In: IEE Seminar on IP over Satellite. The Next Generation: MPLS, DRM VPN and Delivered Services, vol. 2, pp. 1324–1331 (2003)
10. Karir, M., Baras, J.S.: Les: Layered Encryption Security. In: 3rd International Conference on Networking (ICN'04), Guadeloupe (French Caribbean), March 2004, pp. 382–388 (2004)