Analysis of Traffic Data from a Hybrid Satellite-Terrestrial Network

Savio Lau and Ljiljana Trajković
Simon Fraser University
Vancouver, BC, Canada
{saviol, ljilja}@cs.sfu.ca

ABSTRACT

Satellite data networks provide broadband access for areas not served by traditional broadband technologies. In this paper, we describe a collection of traffic data (billing records and *tcpdump* traces) from a satellite Internet service provider in China. We use the billing records to investigate the uploaded and downloaded traffic volume and the aggregate user behavior. We examine daily and weekly cycles and effects of holidays on traffic patterns. Analysis of the *tcpdump* traces indicates that Transmission Control Protocol (TCP) accounts for the majority of data transfers. The analysis also includes the detection of anomalies such as invalid TCP flag combinations, port scans, and anomalies in traffic volume.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks – *Internet*.

General Terms

Measurement.

Keywords

Satellite-terrestrial networks, TCP, traffic measurements.

1. INTRODUCTION

Continuous increase in demand for broadband Internet access has resulted in the increased volume of data traffic and development of new protocols and access technologies. Measurements and analysis of genuine network traffic traces have been used to understand traffic dynamics, characterize traffic and develop new traffic models, and ultimately evaluate network performance.

During the past decade, various traces of wired and wireless terrestrial Internet traffic data have been collected and characterized [1], [2], [3]. Analysis of traffic traces have dealt with identifying characteristics of TCP connections [4] and detecting anomalies [5]. Traffic traces collected from university campuses and research institutions have been made publicly available [6]. However, few traces from commercial satellite networks have been made available to the research community.

In this paper, we analyze patterns and statistical properties of the traffic data collected from a hybrid satellite-terrestrial network operated by ChinaSat, a commercial satellite Internet service provider located in China. We investigate traffic anomalies such as invalid TCP flag combinations, port scans, and anomalies in traffic volume. Detailed statistical analysis of the collected traffic data was reported in [7], where we modeled the TCP connection inter-arrival time and the number of downloaded bytes and performed traffic prediction using the autoregressive integrative moving average (ARIMA) model.

This paper is organized as follows: in Section 2 we describe the DirecPC system. Techniques employed in satellite networks used to improve their performance are presented in Section 3. In Section 4, we describe analysis of billing records. Analysis of *tcpdump* traces and detection of data traffic anomalies are given in Section 5. We conclude with Section 6.

2. DIRECPC SYSTEM

Satellite systems broadcast information over a large geographical area and provide the last mile access for remote sites. DirecPC is an asymmetric satellite network deployed by Hughes Network Systems. It provides television and data services including: DirecTV (a satellite television service), DirecPC (a unidirectional satellite data service), and DirecWay (a new bidirectional satellite data service intended to replace DirecPC). Turbo Internet is the Internet access component of DirecPC. It provides broadband access through a satellite downlink and a return path through a terrestrial dial-up modem. The advertised downlink rate is 400 kbps while a typical rate of a dial-up modem is 33.6 kbps.

ChinaSat provides Internet access through the DirecPC to individual users, businesses, and to over 200 Internet cafés across provinces in China. While we are unable to distinguish between the three groups, the subsequent cluster analysis revealed that 16 users are active 24-hours and 8 business users are active 8-hours daily. The DirecPC system employs geosynchronous satellites, orbiting 35,800 kilometers above the Earth. Hence, the system is designed to overcome long propagation delays (~250 ms) and high bit-error rates (BERs) in the satellite links. DirecPC employs two Transmission Control Protocol (TCP) techniques to improve network performance: TCP splitting and TCP spoofing.

A user's request to browse a website is not sent directly to the destination. Instead, the DirecPC software installed in a satellite user's system adds a "tunneling header" to the request Internet Protocol (IP) packet, redirecting the packet to the Network Operations Center (NOC). At the NOC, the tunneling header is removed and the request is forwarded to the website using a high-speed terrestrial link. The NOC receives the reply from the website and forwards it to the user via the DirecPC satellite link.

Qshine 2007, August 14–17, 2007, Vancouver, BC, Canada.
Copyright 2007 ACM 1-58113-000-0/00/0004…$5.00.

Data paths of the DirecPC system are shown in Figure 1. IP headers and the tunneling header at the user and at the website hosts are shown in Figure 2. Each box indicates an IP source/destination pair. At the client (satellite user), the <Satellite IP, Destination IP> packet is redirected to NOC by embedding the packet into the tunneling header packet <Dial-up IP, NOC IP>.

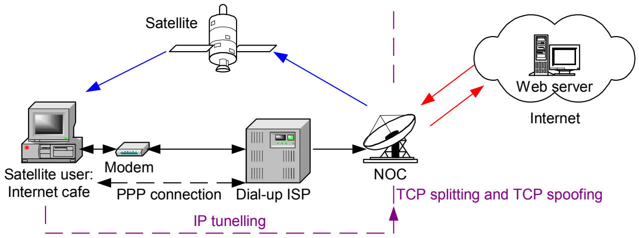


Figure 1. Data paths in the DirecPC system.

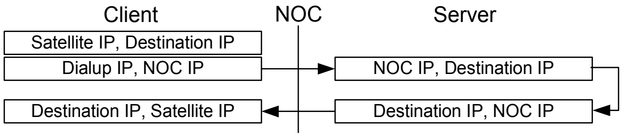


Figure 2. IP headers used in the DirecPC system.

3. SATELLITE NETWORKS: TCP EXTENSIONS

TCP, the most common Internet transport layer protocol [8], was originally not designed for wireless networks. As in the case of wireless networks, satellite networks have two undesirable characteristics: long propagation delays (~250 ms for signals traveling from an earth station to satellite and return) and high BERs. Long propagation delays and large bandwidth result in a large bandwidth-delay product (BDP). BDP indicates the amount of data required to be in transit (unacknowledged) in order to maximize the transfer rate between two connection endpoints. This maximum amount of unacknowledged data is determined by the TCP sliding window. High BERs have a significant impact in networks with long propagation delays. The commonly deployed TCP NewReno may only correct one missing segment per round-trip time by employing its fast retransmit and fast recovery algorithms. Additional missing segments will cause TCP to enter the slow-start phase, resulting in reduced throughput [9], [10].

Various TCP modifications have improved its performance in satellite networks [11] – [13]. The recommended best practices for satellite networks [9] include TCP extensions such as: increasing the TCP's initial congestion window size [14], employing TCP's sliding window scale option [10], using selective acknowledgement option (SACK) [15], and sending path Maximum Transmission Unit (MTU) discovery [16]. Although recommended, these extensions are not all widely deployed.

Performance enhancing proxies (PEPs) [17], a more recent TCP extension, have also been successfully deployed to improve TCP performance in satellite networks. PEPs are techniques employed to improve degraded TCP performance caused by the characteristics of specific link environments. PEPs are not intended for general use because they have an undesirable property to break the TCP end-to-end semantics.

The DirecPC system employs two PEP techniques: TCP splitting and TCP spoofing [18]. An example of these techniques is shown in Figure 3. The TCP connection is split at the NOC (centered vertical line), where it acts as the intermediary between a satellite user (client) and a website (server). The PEP three-way TCP handshake (SYN, SYN/ACK, and ACK) is identical to the handshake in the end-to-end TCP connections. However, subsequent TCP segments from each endpoint are acknowledged by the NOC on behalf of the other endpoint (dashed lines) using a technique known as TCP spoofing. These ACKs are returned to the two endpoints earlier than if end-to-end TCP connection was used. The NOC ignores ACKs transmitted by the two endpoints (dotted lines). TCP spoofing allows the TCP congestion window (*cwnd*) to grow faster, resulting in improved performance.

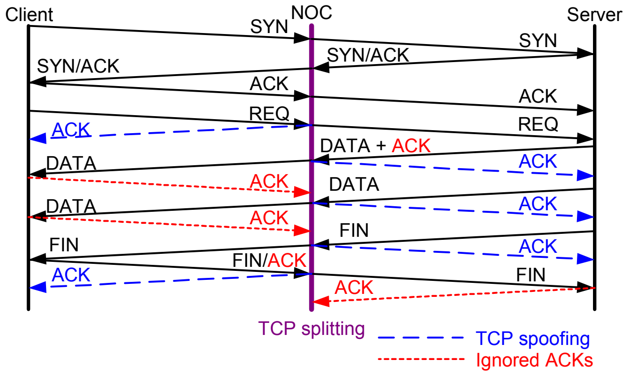


Figure 3. TCP splitting and TCP spoofing.

The PEP technique with TCP splitting and TCP spoofing improves TCP performance. Nevertheless, it imposes considerable memory requirements at the NOC. All segments prematurely acknowledged by the NOC must be kept in local buffers until segments are acknowledged by the endpoints. As a consequence of TCP spoofing, the NOC is also responsible for retransmitting all lost segments.

4. ANALYSIS OF BILLING RECORDS

We have analyzed two months of billing records collected from the DirecPC system: 69 days (1,691 hours) of records during the continuous period from 23:00 on Oct. 31, 2002 to 10:00 on Jan. 10, 2003. The records contain a collection of hourly generated files that consist of connection date, number of downloaded and uploaded packets, volume of downloaded and uploaded bytes, and a hexadecimal ID for each active user during the recorded period. Hence, these billing records capture the hourly network dynamics. Downloaded and uploaded traffic refer to traffic received and traffic sent by satellite users, respectively.

4.1 Volume of Hourly and Daily Traffic

The aggregated downloaded and uploaded hourly and daily traffic data in terms of packet and bytes are shown in Figures 4 – 7. The downloaded traffic (bytes) is larger than the uploaded traffic (bytes) by an order of magnitude. Uploaded number of packets is only slightly higher compared to downloaded number of packets because sent requests are usually followed by a received response. The difference may be attributed to the presence of the User Datagram Protocol (UDP) packets.

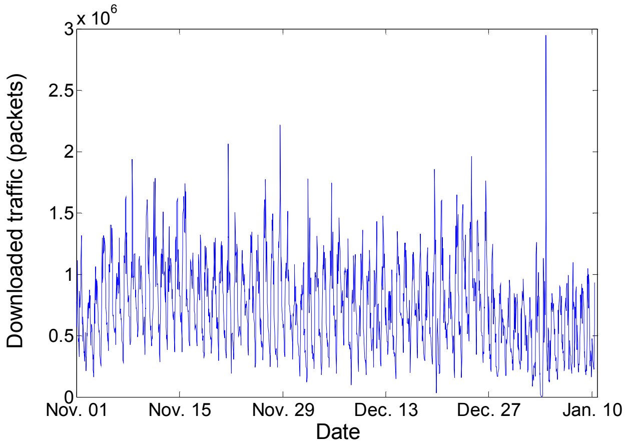
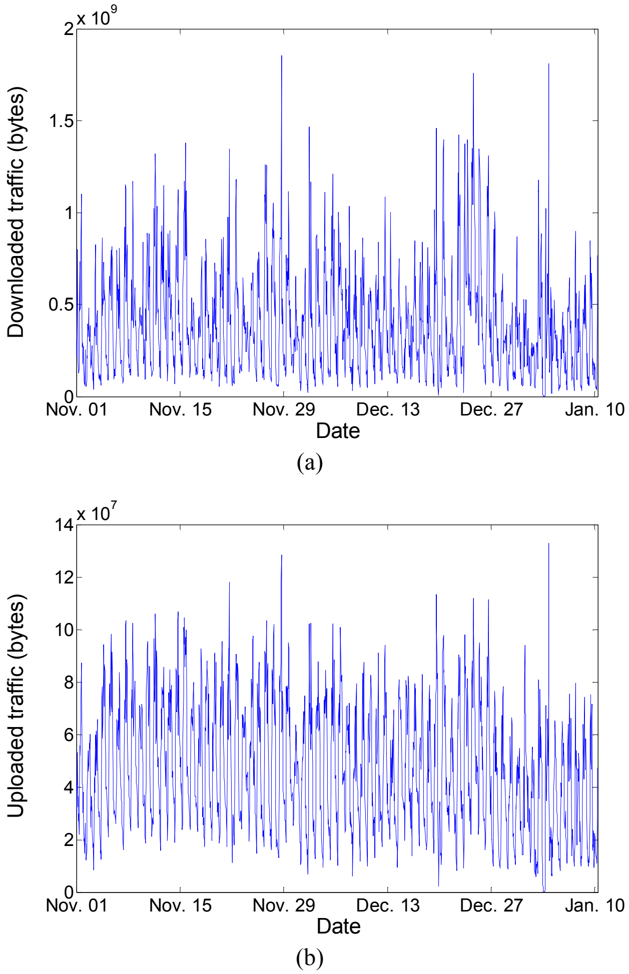


Figure 4. Aggregated traffic of downloaded packets. Uploaded packets have similar trend. Each data point represents the hourly traffic.



(a)

(b)

Figure 5. Aggregated traffic: (a) downloaded bytes and (b) uploaded bytes. Each data point represents the hourly traffic.

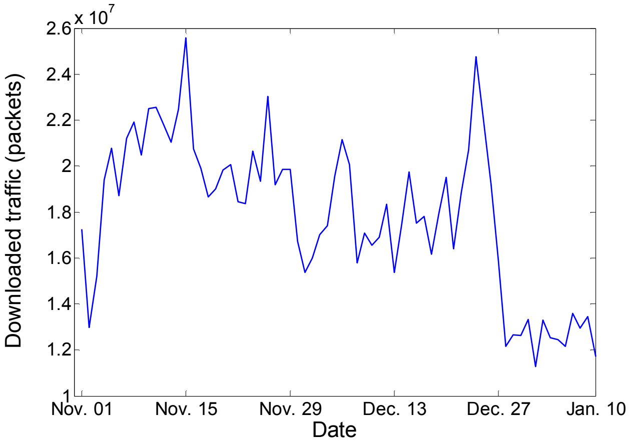
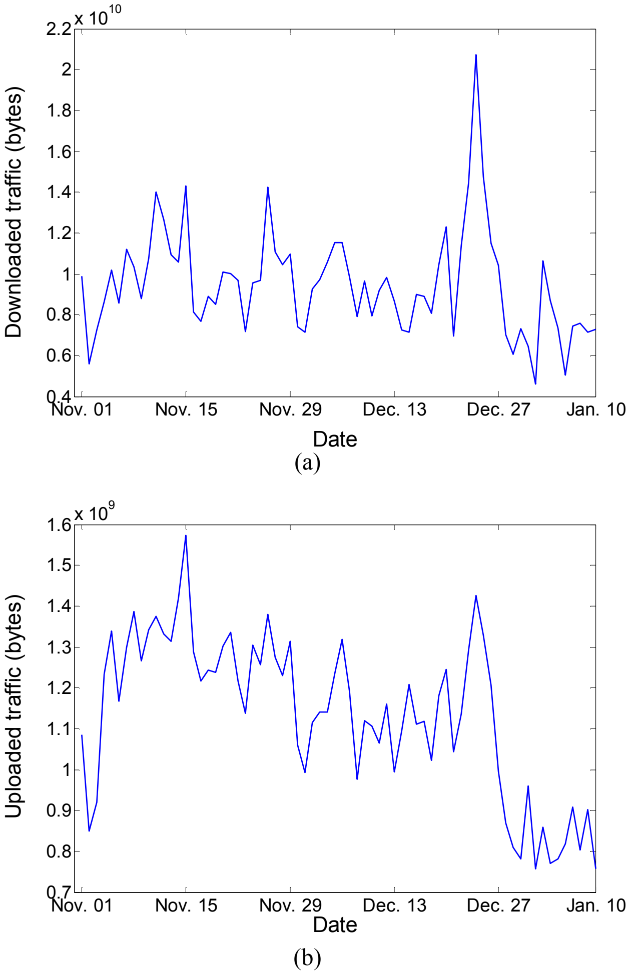


Figure 6. Aggregated traffic of downloaded packets, Uploaded packets have similar trend. Each data point represents the daily traffic.



(a)

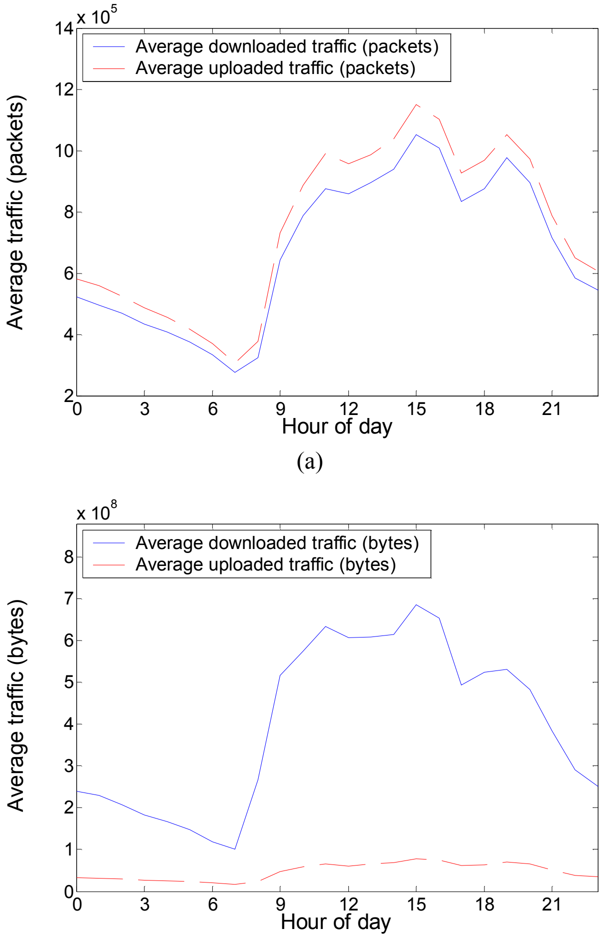
(b)

Figure 7. Aggregated traffic: (a) downloaded bytes and (b) uploaded bytes. Each data point represents the daily traffic.

Trends observed in Figures 4 and 5 exhibit a regular pattern that repeats every 24 hours with an exception of Dec. 24, 2002, when the daily minimum traffic volume is much higher compared to other daily minima. On Jan. 3, 2003, the traffic volume decreased to almost zero, followed by the highest recorded traffic volume, as shown in Figure 4. This change in the traffic pattern was caused by a network outage followed by a recovery. The maximum number of downloaded bytes is recorded on Dec. 24, 2002, as shown in Figure 7, indicating the change in the traffic dynamics during holidays. We also observed a drastic reduction in traffic volume during the extended holiday season between Jan. 1, 2003 and Jan. 10, 2003, as shown in Figures 6 and 7.

4.2 Daily (diurnal) and Weekly Cycles

Daily and weekly cycles were observed by averaging the data traffic for the same hour over all days or over the same day of a week. The daily cycles for (a) downloaded and uploaded packets and (b) downloaded and uploaded bytes are shown in Figure 8. The weekly traffic averages for (a) downloaded and uploaded packets and (b) downloaded and uploaded bytes are shown in Figure 9.

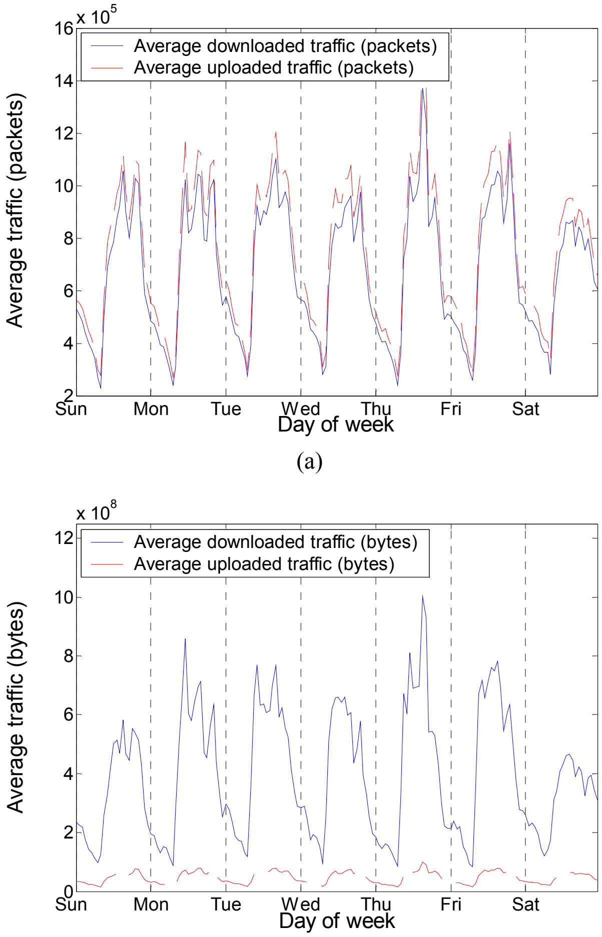


(a)

(b)

Figure 8. Average daily downloaded and uploaded traffic volume in (a) packets and (b) bytes obtained by averaging all recorded values for the same hour.

A daily minimum appears at 7 AM. The data traffic volume then rises rapidly until it reaches the daily maxima at 11 AM, 3 PM, and 7 PM. The traffic volume decreases monotonically from 7 PM until 7 AM. Similar traffic patterns have been reported [19], with the third daily maximum occurring later in the evening (between 9 PM and 10 PM) rather than at 7 PM. As expected, traffic volume on weekends is lower than on working weekdays. The three daily maxima for Wednesdays are not as visible as for other days, as shown in Figure 9 (b). Because both Dec. 24 and Dec. 31, 2002 occur on a Wednesday, this suggests that traffic volume may have different patterns on days immediately prior to holidays.



(a)

(b)

Figure 9. Average weekly downloaded traffic volume in packets (a) and bytes (b) obtained by averaging all recorded values for the same hour on the same weekday.

5. ANALYSIS OF DATA TRAFFIC

The traffic traces were collected at the NOC where traffic from satellite users was rerouted to the NOC by employing the PEP technique. They were collected using the open-source network monitor program *tcpdump* on a Linux PC equipped with a 100 Base-T Ethernet adaptor and a high-resolution (100 µs) timer. The *tcpdump* program was configured to capture the first 68 bytes of each packet to ensure user privacy and to minimize storage requirements while preserving the IP and TCP headers. A port on the primary Cisco router at the NOC located in the Northwest rural area of Beijing, China was used as the network access point for the trace collection. The router provided access to the inbound and outbound packets sent between the hosts using the NOC's 100 Mbps local area network (LAN). A 10 Mbps link connected the NOC to the Internet backbone. The *tcpdump* traffic traces were continuously collected from Dec. 14, 2002 to Jan. 10, 2003. The data were stored in 127 collected files, containing ~63 Gbytes of data.

5.1 Protocols and Applications

The collected traffic traces contain only Internet Protocol (IP) [20] packets because IP is the most widely used network layer protocol. We did not capture traffic from protocols such as the address resolution protocol (ARP) and the reverse address resolution protocol (RARP). The distribution of traffic data by protocols is shown in Table 1. We analyze the activity by TCP port numbers because TCP accounts for majority of the packets. Traffic data in terms of applications, connections, and bytes are shown in Table 2. HTTP/WWW traffic (port 80) is the most widely used TCP application in terms of number of bytes, followed by FTP. Approximately 10% of all connections use unknown ports. We only collected the first 68 bytes of each packet and, hence, we were unable to distinguish between web browsing and peer-to-peer (P2P) applications. However, P2P traffic was not as prevalent in 2003, especially in networks where the uplink is limited by a dial-up connection.

Table 1. Characteristics of traffic data sorted by protocols.

Protocol	Bytes (%)	Packets (%)
TCP	94.50	84.30
UDP	5.06	14.20
ICMP	0.45	1.45
Total	100.00	100.00

Table 2. Characteristics of traffic data sorted by TCP applications.

Applications	Connections (%)	Bytes (%)
WWW (80)	90.00	76.800
FTP-data (20)	0.20	10.700
IRC (194)	0.80	0.008
SMTP (25)	0.10	0.010
POP3 (110)	0.03	0.020
Telnet (23)	0.02	0.002
Others	8.90	12.500
Total	100.00	100.00

There are only few known applications that use a standard UDP port. UDP, an unreliable transport layer protocol, is mainly used for real-time applications such as video streaming and Internet telephony. Many of these applications use random ports. Hence, we cannot apply the same classification technique used for TCP and can only identify the Routing Information Protocol (RIP) packets transmitted on UDP port 520. RIP is used for packet routing between various hosts in a local network. Although we identify a large number of RIP packets, they are not related to the DirecPC traffic in the ChinaSat network. Therefore, we choose not to analyze these packets further.

5.2 TCP Options

TCP extensions such as SACK and the sliding window scale option are requested during the TCP three-way handshake. Hence, we examine the initial two segments (SYN and SYN/ACK) of the TCP three-way handshakes and identify that SACK is widely used in the ChinaSat network. Over 60% of connections support SACK. Less than 5% of connections use the sliding window scale option. The commonly deployed Microsoft Windows OS versions 98 and higher support and enable SACK by default [21]. The sliding window scale option is disabled by default. Thus, the prevalent usage of SACK and the infrequent usage of sliding window scale option in the recorded *tcpdump* traces are caused by the Microsoft Windows TCP implementation. We are unable to determine from the *tcpdump* traces whether other TCP extensions, such as increasing the initial congestion window size and Path MTU discovery, are in use.

5.3 Data Traffic Anomalies

We use the open-source programs *Ethereal/Wireshark* and *tcptrace* to examine the traffic trace at the connection level. Analysis of the *tcpdump* traces reveal data traffic anomalies such as packets with invalid TCP flag combinations, large number of connections closed using TCP reset, port scans, and anomalies in traffic volume.

Packets with Invalid TCP Flag Combinations: TCP SYN, FIN, and RST flags are used to open connections, close connections regularly, and close connections when an error occurs, respectively [8]. These flags cannot be used in combination. Furthermore, the TCP PSH flag cannot be used in combination with RST. Invalid flag combinations may cause TCP/IP implementations to misbehave or fail. They are also used to test TCP/IP robustness [22]. Thus, it is unusual to find packets with combinations of the TCP open/close flags. These packets may be caused by malicious programs or viruses and worms. The number of discovered packets with invalid TCP flag combinations is shown in Table 3. Overall, 0.3% packets with TCP open/close flags have invalid combinations.

Table 3. Packets with various TCP flag combinations. Marked are invalid TCP flag combinations.

TCP flag	Packet count	% of Total
SYN only	19,050,849	48.500
RST only	7,440,418	18.900
FIN only	12,679,619	32.300
* SYN+FIN	408	0.001
* RST+FIN (no PSH)	85,571	0.200
* RST+PSH (no FIN)	18,111	0.050
* RST+FIN+PSH	8,329	0.020
Total number of packets with invalid TCP flag combinations	112,419	0.300
Total packet count	39,283,305	100.000

Large Number of TCP Resets: TCP's usual procedure is to open connections with the SYN flag and to close connections with the FIN flag. However, data from Table 3 indicate that 37% (7,440,418 / (7,440,418 + 12,679,619)) of connections are closed by the RST flag. This is caused by the most commonly used web browser Microsoft Internet Explorer that employs RST instead of

FIN to close connections in order to improve web browsing performance [23].

Port Scans: A significant level of UDP traffic on port 137 originates from the ChinaSat network users. There is also a significant level of outside traffic directed to the ChinaSat network. UDP port 137 is used by the Microsoft NETBEUI protocol, which enables file and printer sharing in a local network of Windows PCs. NETBEUI usually uses UDP port 137 on both endpoints. Thus, traffic from UDP port 137 to other UDP ports or traffic from other UDP ports to UDP port 137 indicates abnormal behavior.

An example of a host in the ChinaSat network (IP address 192.168.2.30) that transmitted packets to Internet hosts from UDP port 137 is shown in Table 4. For a certain destination IP (202.y.y.226), the ChinaSat host transmitted to multiple ports (1025, 1027, 1028, and 1029). This behavior is known as a *port scan* and usually indicates malicious intent. An example of a host external to the ChinaSat network (210.x.x.23) that transmitted packets from UDP port 1035 to ChinaSat hosts at the destination UDP port 137 is shown in Table 5.

Table 4. Port scan originating from the ChinaSat network.

Origin IP:port	Destination IP:port
192.168.2.30:137	195.x.x.42:1026
192.168.2.30:137	202.y.y.226:1026
192.168.2.30:137	218.x.x.238:1025
192.168.2.30:137	202.y.y.226:1025
192.168.2.30:137	202.y.y.226:1027
192.168.2.30:137	202.y.y.226:1028
192.168.2.30:137	202.y.y.226:1029
192.168.2.30:137	202.y.y.242:1026

Table 5. Port scan directed to the ChinaSat network.

Origin IP:port	Destination IP:port
210.x.x.23:1035	192.168.1.121:137
210.x.x.23:1035	192.168.1.63:137
210.x.x.23:1035	192.168.2.11:137
210.x.x.23:1035	192.168.1.250:137
210.x.x.23:1035	192.168.1.25:137
210.x.x.23:1035	192.168.2.79:137
210.x.x.23:1035	192.168.1.52:137
210.x.x.23:1035	192.168.6.191:137

Two Internet worms Bugbear and Opasoft were prevalent at the time the *tcpdump* traces were captured. Both worms use the NETBEUI protocol to propagate to other hosts. Without having the TCP payload recorded, we are unable to determine if these two worms indeed generated the port scans.

Traffic Volume Anomalies: Wavelets decomposition of data traffic has been used to identify anomalies in traffic volume [24]. We record packet count and bytes from the collected *tcpdump* traces and analyze 226,390 seconds of binned traffic data. We decompose the traffic data into 12 levels by employing the *Debauchies 9* mother wavelet using *Matlab* [25]. Each data point at the coarsest level approximately represents the hourly data traffic (2^{12} seconds correspond to approximately one hour), similar to the time scale of the billing data. The approximation of the downloaded packets (a_{12}), the coarsest detail coefficients at level 12 (d_{12}), and the detail coefficients at level 6 (d_6) are shown in Figures 10, 11, and 12, respectively.

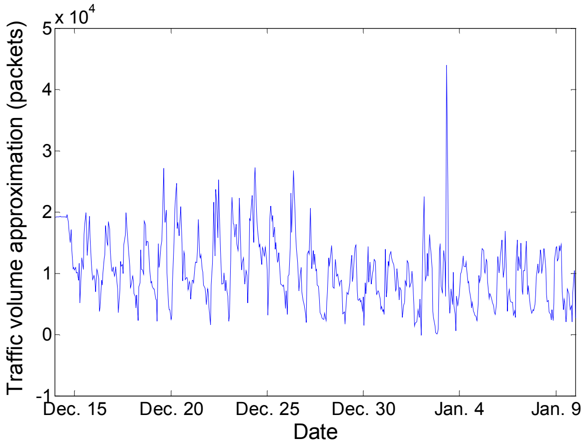


Figure 10. Wavelet approximation of the *tcpdump* trace (downloaded packets) at the coarsest time scale.

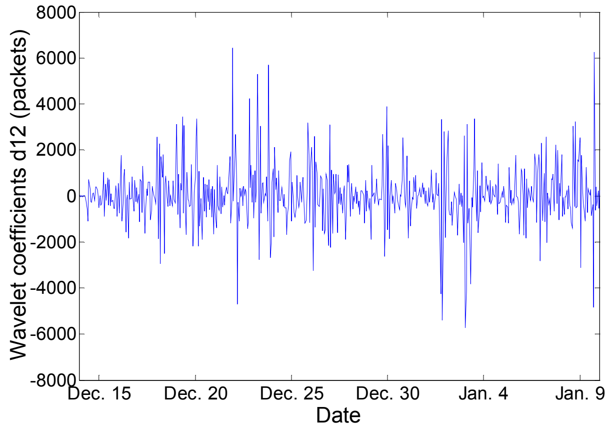


Figure 11. Detail wavelet coefficients d_{12} of the tcpdump trace (downloaded packets) at the coarsest level (time scale equals to 2^{12} seconds).

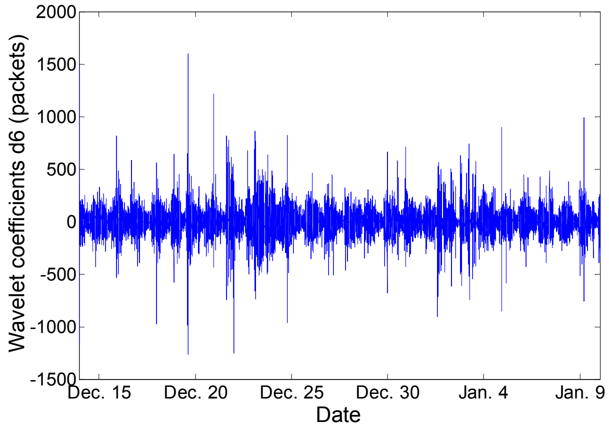


Figure 12. Detail wavelet coefficients d_6 of the *tcpdump* trace (downloaded packets) at level 6 (time scale equals to 2^6 seconds).

Multiple spikes, observed in Figure 11, correspond to large variations in the hourly traffic volume. The spikes are detected by using a moving window of length 20 and by calculating the standard deviation for each window. Anomalies in traffic volume are those data points that are above or below three times the standard deviation (3σ value). An example is the anomaly detected on Jan. 3, 2003. It was due to a network outage and recovery that was also detected in the records of billing data.

Finer detail levels can be used to detect anomalies at various time scales. The wavelet details on the time scale of one minute can be found by examining detail coefficients d_6, shown in Figure 12. The anomaly detected on Dec. 19, 2002 cannot be observed from the coarser level coefficients (such as the d_{12}). The anomaly was caused by port scans that lasted approximately one minute.

6. CONCLUSIONS

In this paper, we described traffic collection in a commercial hybrid satellite-terrestrial network and analyzed the billing records and collected traffic traces. The billing records indicated that the downloaded and uploaded traffic patterns are highly regular, exhibiting both daily and weekly cycles. A daily minimum occurs at 7 AM while three daily maxima occur at 11 AM, 3 PM, and 7 PM. Analysis of *tcpdump* traces shows that the trace is dominated by TCP traffic, with HTTP/WWW packets contributing to the majority of captured data. By examining the TCP SYN packets, we determined that the SACK TCP extension is widely used to improve the TCP performance in satellite networks. We also detected data traffic anomalies, including invalid TCP flag combinations, large number of TCP resets, port scans, and abnormal changes in traffic volume. We have provided plausible explanations for the origin of these anomalies. While the collected traffic data captured only a sample of the satellite network behavior, the analysis presented here may contribute to better performance evaluation of deployed commercial networks.

7. ACKNOWLEDGMENTS

The authors would like to thank Y. Shi and R. Huang, managers of the DirecPC system at ChinaSat, for their help with data collection. We also thank Q. Shao, B. Vujicic, R. Narayanan, and M. Omueti for valuable ideas and suggestions.

8. REFERENCES

[1] S. McCreary, “Trends in wide area IP traffic patterns,” *Proc. 13th ITC Specialist Semin. on Meas. and Modeling of IP Traffic*, Monterey, CA, Sept. 2000, pp. 1–11.

[2] K. Thompson, G. J. Miller, and R. Wilder, “Wide-area Internet traffic patterns and characteristics,” *IEEE Network*, vol. 11, no. 6, pp. 10–23, Nov. 1997.

[3] D. Tang and M. Baker, “Analysis of a metropolitan-area wireless network,” in *Proc. of ACM Mobicom ’99,* pp. 13–23, Seattle, WA, Sept. 1999.

[4] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, “Inferring TCP connection characteristics through passive measurements,” *Proc. INFOCOMM 2004*, Hong Kong, HK, Mar. 2004, pp. 1582–1592.

[5] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” *Proc. ACM SIGCOMM ’04*, Portland, OR, Aug. 2004, pp. 219–230.

[6] The Internet Traffic Archive [Online]. Available: http://ita.ee.lbl.gov/.

[7] Q. Shao and Lj. Trajković, “Measurement and analysis of traffic in a hybrid satellite-terrestrial network,” *Proc. SPECTS 2004*, San Jose, CA, July 2004, pp. 329–336.

[8] J. Postel, Ed., “Transmission control protocol,” *RFC 793*, Sept. 1981.

[9] M. Allman, D. Glover, and L. Sanchez, “Enhancing TCP over satellite channels using standard mechanisms,” *RFC 2488*, Jan. 1999.

[10] V. Jacobson, R. Braden, and D. Borman “TCP extensions for high-performance,” *RFC 1323*, May 1992.

[11] M. Allman et. al., “Ongoing TCP research related to satellites,” *RFC 2760*, Feb. 2000.

[12] S. Oueslati-Boulahia, A. Serhrouchni, S. Tohmé, S. Baier, and M. Berrada, “TCP over satellite links: problems and solutions,” *Telecommun. Syst.*, vol. 13, no. 2–4, pp. 199–212, July 2000.

[13] T. R. Henderson and R. H. Katz, “Transport protocol for Internet-compatible satellite networks,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 2, pp. 326–344, Feb. 1999.

[14] M. Allman, S. Floyd, and C. Partridge, “Increasing TCP’s initial window,” *RFC 2414*, Sept. 1998.

[15] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP selective acknowledgement options,” *RFC 2018*, Oct. 1996.

[16] J. Mogul and S. Deering, “Path MTU discovery,” *RFC 1191*, Nov. 1990.

[17] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, “Performance enhancing proxies intended to mitigate link-related degradations,” *RFC 3135*, June 2001.

[18] J. S. Baras, S. Corson, S. Papademetriou, I. Secka, and N. Suphasindhu, “Fast asymmetric Internet over wireless satellite-terrestrial networks,” *Proc. MILCOM ’97*, Monterey, CA, Nov. 1997, pp. 372–377.

[19] V. Paxson, “Empirically-derived analytic models of wide-area TCP connections,” *IEEE/ACM Trans. on Networking*, vol. 2, no. 4, pp. 316–336, Aug. 1994.

[20] J. Postel, Ed., “Internet protocol,” *RFC 791*, Sept. 1981.

[21] Microsoft Windows 2000 TCP/IP implementation details.. [Online]. Available: http://www.microsoft.com/technet/itsolutions/network/deploy/depovg/tcpip2k.mspx.

[22] J. Postel, “TCP and IP bake off,” *RFC 1025*, Sept. 1987.

[23] M. Arlitt and C. Williamson, “An analysis of TCP reset behavior on the Internet,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 37–44, Jan. 2005.

[24] P. Barford, J. Kline, D. Plonka, and A. Ron. “A signal analysis of network traffic anomalies,” *Proc. ACM SIGCOMM Internet Measurement Workshop 2002*, Marseilles, France, Nov. 2002.

[25] Matlab [Online]. Available: http://www.mathworks.com/products/matlab/