Network Traffic Mining with Open Source Tools (Data Part)

Xiaming Chen
OMNI-Lab, Shanghai Jiao Tong Univ., China
chenxm35@gmail.com

I. INTRODUCTION

Traditional Internet Service Providers (ISPs) today meet the challenge of keeping profit gain against the declining of transportation price per byte. Thus they don't want just to be data haulers to move data from this place to another like before. With the boom of Big Data and Cloud computing technologies, ISPs become more interested and investigate more funds in network traffic mining (NTM) technology, to extract valuable information about their customers from the pipes and sell it to people who need it under some agreement with the customers.

NTM is a technique of the essence for network management (e.g., network problem diagnosis, anomaly behavior detection, network protocol development etc.) and service management (e.g., Quality of Service guarantee, service infrastructure deployment, resource management etc.) [1]. It is a subdomain of data mining regarding to network traffic that is usually integrated with technologies of network forensics, statistical analysis, behavior informatics, and machine learning. Recent years, emerging smart mobile equipment and applications provide affluent data dimensions in network traffic about user's location, preference, sociality, perceived performance and so on, to make it possible to mine *movement patterns, social contact patterns, shopping patterns, perceived performance patterns* about one specific user or a group of users. These patterns can provide references of real users to design marketing and advertisement strategy, decide service deployment schema, and perform network/application optimizations.

This documentation will first introduce the network-traffic-mining platform (NTMP)(especially for mobile traffic mining) in our OMNI-Lab with well-developed open source tools, and second, try to freeze the output data format of the platform to provide a uniform interface to following researchers and development of their analysis tools. As organized, the network-traffic-mining platform is described in Section 2 and data output formats given in Section 3.

II. NETWORK TRAFFIC MINING PLATFORM

NTMP is a passive network traffic measurement platform constructed based on commercial computing hardwires and open source tools. As the mobile case shown in Fig. 1, NTMP is working on the traffic mirrored from the backbone router, where full packets of bi-directional flows are seen on the link. In our scenario, the backbone router serves for the networking of three campuses located at four different geographical places. Totally, around 2000 live Wi-Fi APs are detected in the networks.

In NTMP, it's designed to support multiple miners working concurrently, even when system does not allow parallel sniffing the same network interface by different miners. Tow main components are contained in NTMP: traffic management module and traffic mining module. The former gets the functionality of filtering out targeted flows and replicating them into different interfaces or hosts on which miners run. Traffic Filter (TF) is a rule-based flow selector considering flow source/destination IP/port. Each rule supports a configuration of four-tuple. Rules are matched in the order when they are added to the system, and following rules won't be tested if one rule has been matched. Traffic Replicator (TR) performs to replicate flows generated by the filter in multiple copies and sends them to different network interfaces.

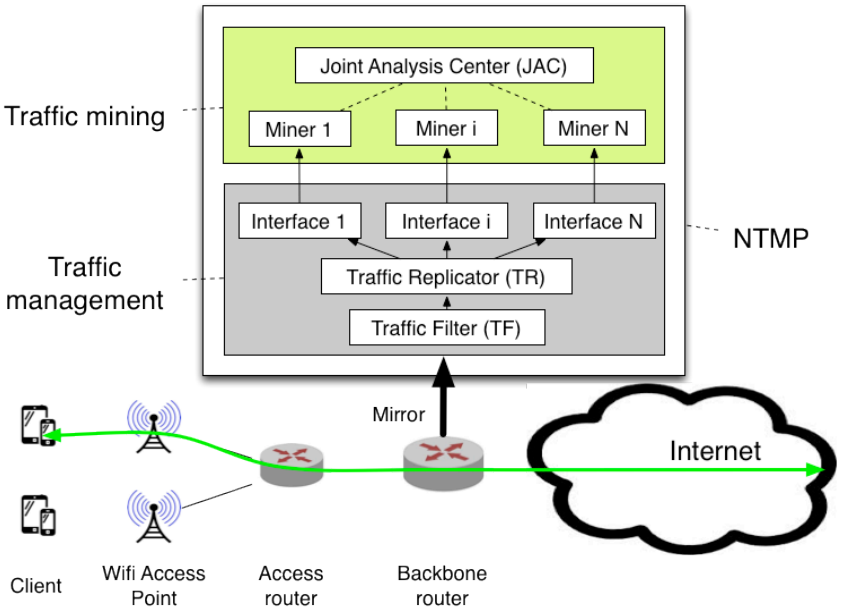

Figure 1 Network-traffic-mining platform

The traffic-mining module runs multiple miners on replicated traffic copies to perform different analysis of the same traffic data. This design obtains two advantages: first, it avoids resource lock on some system where two applications are not allowed to access the same network interface concurrently; second, different miners can work on both standalone host or distributed cluster, which empower the scalability of the platform. Another part of this module is the Joint Analysis Center (JAC). In JAC, various analyses can be performed based on the outputs of different miners. Some services may need only the output of specific miner while others acquire multiple results of different miners and perform joint analysis. JAC is a predefined sub-module representing future functionalities to develop based on miners' outputs.

In this documentation, we describe three types of open source tools (miners) working on live network traffic: Justniffer [2], Tstat [3], and pcapDPI [4]. For the first two of them, we refer the reader to their websites. The third is a simple C program based on nDPI [5] that is a forked branch of original openDPI; the latter was announced to be unavailable when ipoque was acquired by Rohde & Schwarz [6]. Overall, three types of information are obtained finally [7]: *HTTP information, TCP statistics and traffic classification label*. In next part, we will describe the data formats generated by them and freeze them to support future development of tools in JAC.

III. DATA OUTPUT FORMAT DESCRIPTIONS

A. Format of HTTP information

Justniffer generates the HTTP information where we consider de facto and popular web and proxy ports when capturing, including: 80, 8080, 3128, 8081, 9080, 8000, 8001. Each row in the log corresponds to a different request/response pair (Note, not a tcp connection) and each column is associated to a specific feature. Totally 38 features are recorded and "N/A" present when feature is not available. All features numbered in order they appear in the log file are listed in Table 1.

Table 1 features of HTTP information

#	Field name	Description
Overall information of connection		
1	source_ip	Source IP address
2	source_port	Source TCP port
3	dest_ip	Destination IP address

4	dest_port	Destination TCP port
5	connection	Connection persistence indicator: unique: the req/rsp is the unique in the tcp connection start: the req/rsp is the first in the tcp connection last : the req/rsp is the last in the tcp connection continue : the req/rsp is the middle in the tcp connection
6	connection_timestamp	The timestamp in seconds initializing the tcp connection
7	close_timestamp	The timestamp in seconds close the tcp conenction
8	connection_time	Elapsed time (3-way HS) for establishing the tcp connection
9	idle_time_0	Elapsed time form when the connection is established and before the request is started
10	request_timestamp	The timestamp in seconds of the first packet of request
11	request_time	Elapsed time form when the first packet and the last packet of request
12	response_timestamp	The timestamp in seconds of the first packet of response
13	response_time_begin	Elapsed time form when the last packet of request and the first packet of response
14	response_time_end	Elapsed time form when the first packet and the last packet of response
15	idle_time_1	Elapsed time form when the last response is finished and before new request is started or the tcp connection is closed
16	request_size	The size of request (including request header length)
17	response_size	The size of response (including response header length)
Request header fields		
18	request_method	The request method (e.g., GET, PUT, HEAD)
19	request_url	The request URL
20	request_protocol	The request protocol (e.g., HTTP/1.0, HTTP/1.1)
21	request_ host	The request HOST header value
22	request_ user_agent	The request User-Agent header value
23	request _referer	The request Referer header value
24	request _connection	The request Connection header value
25	request _keep_alive	The request Keep-Alive header value
Response header fields		
26	response_protocol	The response protocol
27	response_code	The response code (e.g., 200, 301, 404)
28	response _server	The response Server header value
29	response _content_length	The response Content-Length header value
30	response _content_type	The response Content-Type header value
31	response _content_encoding	The response Content-Encoding header value
32	response _etag	The response header ETag value
33	response _cache_control	The response header Cache-Control value
34	response _last_modified	The response header Last-Modified value
35	response _age	The response header Age value
36	response _expires	The response header Expires value
37	response _connection	The response header Connection value
38	response _keep_alive	The response header Keep-Alive value

B. Format of TCP statistics

Tstat generates the TCP statistics where each row corresponds to a different flow and each column is associated to a specific measure. When it is useful, the columns are grouped according to C2S (Client-to-Server) and S2C (Server-to-Client) traffic directions. The generated logs are:

log_tcp_complete, log_tcp_nocomplete:

report every TCP connection that has been tracked by Tstat. A TCP connection is identified when the first SYN segment is observed, and is ended when either:

- the FIN/ACK or RST segments are observer;
- no data packet has been observed (from both sides) for a default timeout of 10s after the thress-way handshake or 5min after the last data packet

Tstat discards all the connections for which the three-way handshake is not properly seen. Then, in case a connection is correctly closed it is stored in log_tcp_complete, otherwise in log_tcp_nocomplete. For detailed description, please refer to official documentation at [8].

log_udp_complete:
reports every tracked UDP flow pair. An UDP flow pair is identified when the first UDP segment is observed for a UDP socket pair, and is ended when no packet has been observed (from both sides) for 10s after the first packet or 3min after the last data packet. For detailed description, please refer to official documentation at [9].

Besides, Tstat also reports statistics for RTP and RTCP flows (see log_mm_complete), skype flows (see log_skype_complete), MSN, Yahoo! Messenger, chat based on XMPP flows (see log_chat_complete and log_chat_messages), video flows (see log_video_complete) and streaming flows (see log_streaming complete).

C. Format of traffic classification output

pcapDPI generates statistics about traffic classification where each row corresponds to a different flow and each column is associated to a specific feature. Table 2 lists all features recorded and Table 3 gives the list of application protocols supported by nDPI.

Table 2 features of traffic classification

#	Field name	Description
1	Source_ip	The source IP address
2	Source_port	The source TCP port
3	Dest_ip	The destination IP address
4	Dest_port	The destination TCP port
5	L4_protocol	The protocol field in TCP header (e.g., TCP, UDP, SNMP)
6	Detect_protocol	The detected application protocol names
7	Packets	Total packets in both direction of a flow
8	Bytes	Total bytes in both direction of a flow

Table 3 list of application protocols

#	Protocol	#	Protocol	#	Protocol
0	UNKNOWN	56	SHOUTCast	112	LDAP
1	FTP	57	SopCast	113	MapleStory
2	POP	58	TVAnts	114	msSQL
3	SMTP	59	TVUplayer	115	PPTP
4	IMAP	60	VeohTV	116	WARCRAFT3
5	DNS	61	QQLive	117	World of Kung Fu
6	IPP	62	Thunder/ Webthunder	118	MEEBO
7	HTTP	63	Soulseek	119	FaceBook
8	MDNS	64	GaduGadu	120	Twitter
9	NTP	65	IRC	121	DropBox
10	NETBIOS	66	Popo	122	Gmail
11	NFS	67	Jabber	123	Google Maps
12	SSDP	68	MSN	124	YouTube
13	BGP	69	Oscar	125	Skype
14	SNMP	70	Yahoo	126	Google
15	XDMCP	71	Battlefield	127	DCE RPC
16	SMB	72	Quake	128	NetFlow_IPFIX
17	SYSLOG	73	VRRP	129	sFlow
18	DHCP	74	Steam	130	HTTP Connect (SSL over HTTP)

19	PostgreSQL	75	Halflife2	131	HTTP Proxy
20	MySQL	76	World of Warcraft	132	Netflix
21	TDS	77	Telnet	133	Citrix
22	DirectDownloadLink	78	STUN	134	CitrixOnline /GotoMeeting
23	I23V5	79	IPSEC	135	Apple (iMessage, FaceTime)
24	AppleJuice	80	GRE	136	Webex
25	DirectConnect	81	ICMP	137	WhatsApp
26	Socrates	82	IGMP	138	Apple iCloud
27	WinMX	83	EGP	139	Viber
28	VMware	84	SCTP	140	Apple iTunes
29	PANDO	85	OSPF	141	Radius
30	Filetopia	86	IP in IP	142	WindowsUpdate
31	iMESH	87	RTP	143	TeamViewer
32	Kontiki	88	RDP	144	Tuenti
33	OpenFT	89	VNC	145	LotusNotes
34	Kazaa/Fasttrack	90	PCAnywhere	146	SAP
35	Gnutella	91	SSL	147	GTP
36	eDonkey	92	SSH	148	UPnP
37	Bittorrent	93	USENET	149	LLMNR
38	OFF	94	MGCP	150	RemoteScan
39	AVI	95	IAX	151	Spotify
40	Flash	96	TFTP	152	H323
41	OGG	97	AFP	153	OpenVPN
42	MPEG	98	StealthNet	154	NOE
43	QuickTime	99	Aimini	155	CiscoVPN
44	RealMedia	100	SIP	156	TeamSpeak
45	Windowsmedia	101	Truphone	157	Tor
46	MMS	102	ICMPv6	158	CiscoSkinny
47	XBOX	103	DHCPv6	159	RTCP
48	QQ	104	Armagetron	160	RSYNC
49	MOVE	105	CrossFire	161	Oracle
50	RTSP	106	Dofus	162	Corba
51	Feidian	107	Fiesta	163	UbuntuONE
52	Icecast	108	Florensia		
53	PPLive	109	Guildwars		
54	PPStream	110	HTTP Application Activesync		
55	Zattoo	111	Kerberos		

REFERENCES

[1] Netowrk management: http://en.wikipedia.org/wiki/Network_management
[2] Homepage of Justniffer: http://justniffer.sourceforge.net/
[3] Homepage of Tstat: http://tstat.tlc.polito.it/index.shtml
[4] Homepage of pcapDPI: https://github.com/caesar0301/pcapDPI
[5] Homepage of nDPI: http://www.ntop.org/products/ndpi/
[6] “Bye bye OpenDPI”: http://lastsummer.de/bye-bye-opendpi/
[7] Homepage of PcapEx: https://github.com/caesar0301/PcapEx
[8] TCP measures of Tstat: http://tstat.tlc.polito.it/measure.shtml#log_tcp_complete
[9] UDP measures of Tstat: http://tstat.tlc.polito.it/measure.shtml#log_udp_complete