Wireshark を利用した専攻共通計算機室ネットワークトラブル対応

[○]鬼頭良彦^{A)}、藤原冨未治^{A)}

A) 工学系技術支援室 情報通信技術系

概要

現在担当している業務の中に専攻共通計算機室の管理がある。共通計算機室は専攻関連の学生、教職員が 自由に利用できるWindowsパソコンと専攻のメールやWeb等のサーバがセキュリティの関連でローカルの同 ーネットワークで運用されている。

その環境下でおきるトラブルでハード関連やログが残るものについては対応が容易であるが、ネットワー クの遅延等のログを調査しても原因が特定できないといった場合には対応に時間がかかることがあった。

そのような場合にフリーソフトの Wireshark を利用してパケットキャプチャーを行い、ネットワークの状況 を観察することによって原因究明の手掛かりとなることがあるのでその報告をする。

1 はじめに

専攻共通計算機室は Windows Server 2003 と Windows Server 2008 のサーバで管理を行い、クライアントと しては Windows7 が 20 台、WindowsXP が 4 台、Windows2000 が 5 台で学生実験やレポート作成用に自由に利 用できる環境となっている。ネットワーク的には全てローカル内で運用されており、同じネットワークに専 攻のメールサーバ、Web サーバ等も配置されている。その為、ネットワーク内でおきたトラブルについては メール等専攻全体に影響を及ぼすので早期の原因究明が必要となる。

2 Wireshark とは

フリーのネットワークアナライザでパソコンベースで簡単に利用ができ、ネットワークに流れるパケット のチェック、またデータを保存しておけば後で詳細分析も可能である。そのためノートパソコンにインスト ールしておけば様々な場所でネットワークの状況が確認できる。最新バージョンは1.8.5 で次のサイトからダ ウンロードが可能である。

Wireshark Web $\sim - \vec{v}$: http://www.wireshark.org/download.html

3 Wireshark の使用方法

パケットキャプチャを行う場合、ネットワーク全体 を観察するか、個別機器の状況を観察するかで別途器 具が必要となるが、Wireshark の使用方法は変わらな い。ネットワーク全体の場合は HUB に Wireshark を インストールしたパソコンを接続することで観察可 能であり、サーバ等の個別機器のアクセス状況を観察 したい場合は図1のようにパケットのミラーリング 可能なスイッチング HUB を利用する。我々のところ では CenterCOM FS808TP V1 という携帯に便利な



図1. 個別機器のパケット観察

HUB を利用している。

3.1 パケットのキャプチャ方法

Wireshark を起動すると図2の起動画面が表示される。

キャプチャを開始するには起動画面上部メニューの Capture から Options を選択する。Option メニューで図3に表示の項目 にチェックをいれて Start ボタンを押すとキャプチャが開始さ れる。チェック項目の説明は次の通りである。

- Capture all in promiscuous mode: NIC に届いたフレームを全 てキャプチャする
- Update list of packets in real time: キャプチャしたパケットを リアルタイム表示する。
- Automatic scrolling in live capture: パケット一覧表示を自動的 にスクロールする。
- Hide capture info dialog: 別画面で表示されるキャプチャ情報 画面を非表示にする。
- Enable transport name resolution: プロトコルのポート番号をプロトコル名に変換する。

また、長時間の観察が必要な場合は Capture File の設定を行う ことによって保存ファイルの分割を行うことができる。

図4にキャプチャが開始された時の画面を示す。画面上部が パケット一覧部でパケットデータがリアルタイムで表示され、 自動スクロールされる。画面中央がパケット詳細部、画面下部 がパケットデータ部になる。ある程度キャプチャを行ったとこ ろで停止させ、パケット一覧部、詳細部でトラブルのチェックを行う。



図2. Wireshark 起動画面





📶 Inte	el(R) 8257	8DM Gigabit Ne	twork Connec	tion: ¥Devi	e¥NPF_{743	30E200	-9A2C-40A	6-B7C2-	C7E04AA5	5E027}	[Wiresha	ark 1.8.0	(SVN		x
<u>File</u>	<u>E</u> dit <u>V</u> iew	<u>Go</u> <u>C</u> apture	<u>A</u> nalyze <u>S</u> ta	atistics Tele	phon <u>y T</u> ook	ls <u>I</u> nte	rnals <u>H</u> elp								
		🕷 🖻 🗖	* 2 8	0, 4	🔷 🥥 😚	₽		€. 0.	0	X	1 🖪 🐝				
Filter: Expression Clear Apply Save															
No.	Time		Source		Destination		Protocol	Length	n Info						*
315 316 317	17:14:0 17:14:0 17:14:0	9.347364000 9.347599000 9.347984000	192.168. 124.83.2 124.83.2	1.21 48.251 48.251	124.83.248 192.168.1. 192.168.1	8.123 .21	TCP HTTP TCP	54 27:	4 outlaw 1 Contir 0 http >	vs > ht nuation	tp [RST or nor s-rbt-i	r] Seq= n-HTTP	=2 Win=0 traffic) Len=0 : Seq=218	AC
318 319	17:14:0 17:14:0	9.348030000 9.348043000	192.168. 124.83.2	1.21 48.251	124.83.248 192.168.1	8.251	ТСР НТТР	54 27:	4 opalis 1 Contir	-rbt-i nuation	pc > ht or nor	tp [RS	traffic	Seq=2 A	ck=
320 321	17:14:0 17:14:0	9.348482000 9.348514000	124.83.2 192.168.	48.251 1.21	192.168.1.	.21 8.251	TCP TCP	6) 54	0 http > 4 opalis	⊳ perma s-rbt-i	<mark>bit-cs</mark> pc > ht	[FIN, tp [RS	ACK] Seq	eq=218 Ac =2 Win=0	k=2 📖 Len
322	17:14:0	9.348524000	192.168.	1.21	124.83.248	8,251	тср	54	4 permah	oit-cs	> http	[RST.	ACK] SE	≏α=2 Ack=	218
B Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits) on interface 0 E thernet II, Src: 78:2b:cb: (78:2b:cb:), Dst: 00:0a:79: (00:0a:79:)) B Destination: 00:0a:79:93:40:d3 (00:0a:79:) B Source: 78:2b:cb:8e:e4:9d (78:2b:cb:) Type: IP (0x0800) □ Internet Protocol Version 4, Src: 192.168.1.21 (192.168.1.21), Dst: 133.6. (133.6.))									* E						
Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport)) Total Length: 106 Identification: 0x0eb2 (3762) Elags: 0x00 Fragment offset: 0															
0000 0010 0020 0030 0040 0050	00 0a 00 6a 09 73 06 70 02 01 02 01	79 93 40 d3 De b2 00 00 ed 10 00 a1 75 62 6c 69 D0 30 33 30 D5 01 05 00	78 2b cb 80 11 1b 00 56 d0 63 a0 3f 0f 06 0b 30 0f 06	8e e4 9d 9a c0 a8 dc 30 4c 02 02 02 2b 06 01 0b 2b 06	08 00 45 01 15 85 02 01 00 44 02 01 02 01 19 01 02 01	00 06 04 00 03 19	y.@.x+ .jv .public. 030 030	(0L. ?D .+	. E.						• III •
💛 Frai	me (frame)), 120 bytes		P Profi	a: Default	_				_	_				

図4. パケットキャプチャ画面

3.2 キャプチャデータの分析方法

分析方法としては分析目的によって手法が違ってくるが、我々のところでは主にトラブル究明のヒントと して利用するため、図4の上部メニューの Statistics から Flow Graph を選択し、データ全体をラダー表示させ て通信の流れをチェックしている。図5に Flow Graph 画面を示す。左側に時間、中央部に送受信の方向、右 側にコメントが掲載されている。この画面から注目すべきプロトコルや Mac アドレス等があれば図4のメニ ューの下にある Filter でデータの絞り込みを行い、再度 Flow Graph でチェックを行い、最後に図4中央のパ ケット詳細部で詳細データのチェックを行う方法をとっている。



図5. Flow Graph 画面

4 トラブル事例紹介

次に専攻共通計算機室でおきたネットワークトラブルの事例を二つ照会する。

4.1 事例1 (ブロードキャストストーム)

専攻のメールやWebへのアクセスができなくなったとの連絡があり、各サーバを調べると稼働はしており、 ログにはサーバ間の通信がタイムアウトとなりマウントが切れたとの記録があった。サーバ室のネットワー ク機器を調べたが故障が見つからず原因が不明であった。そこでWiresharkでネットワーク上のパケットを調 べると図6のように同じパケットが大量に流れており、ブロードキャストストームの状態が確認できた。そ こでクライアント室を調べ、異常点滅をしている HUB を停止させるとブロードキャストストームが収まった。 原因は HUB の故障と思われたが、調べるとパソコンに接続されていたネットワークケーブルが何らかの理由 で外され、HUB に差し戻されていたためブロードキャストストームがおきたと思われる。

130211.pcapng [Wireshark 1.8.0 ((SVN Rev 43431 from /trunk-1.8)]	AL & # 1								
Efe Edit View Go Capture Analyze Statistics Telephony Tools Internals Help										
	। 🔍 🔶 🗢 🗿 🖥 🔳 🗐 🗐 🔍 Q	0, 🗉 👹 🖾 🥵 % 😫								
Filter: Expression Clear Apply Save										
No. Time Sou	rce Destination Protocol Le	ingth Info	*							
44216 11:03:32.035083000 193 44217 11:03:32.035089000 193	2.168.100.130 192.168.100.255 UDP 2.168.100.130 192.168.100.255 UDP	74 Source port: 49580 Destin 74 Source port: 49580 Destin	nation port: 22936 nation port: 22936							
44218 11:03:32.035096000 19 4219 11:03:32.036198000 19 44220 11:03:32.036210000 19	2.168.100.130 192.168.100.255 UDP 2.168.100.130 192.168.100.255 UDP 2.168.100.130 192.168.100.255 UDP	74 Source port: 49580 Destin 74 Source port: 49580 Destin 74 Source port: 49580 Destin	nation port: 22936 nation port: 22936							
14221 11:03:32.036216000 193 14222 11:03:32.036222000 193	2.168.100.130 192.168.100.255 UDP 2.168.100.130 192.168.100.255 UDP	74 Source port: 49580 Destin 74 Source port: 49580 Destin	nation port: 22936 nation port: 22936							
44223 11:03:32.036353000 19 44224 11:03:32.036363000 19 44225 11:03:32.036363000 19	2.168.100.130 192.168.100.255 UDP 2.168.100.130 192.168.100.255 UDP	74 Source port: 49580 Destin 74 Source port: 49580 Destin 74 Source port: 49580 Destin	nation port: 22936 nation port: 22936							
44226 11:03:32.036376000 19	2.168.100.130 192.168.100.255 UDP	74 Source port: 49580 Destin	hation port: 22936							
Frame 1: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0 Ethernet 11, Src: Toshiba (00:1c:7e:), pst: Assistek (c8:60:00) Internet Protocol Version 4, Src: 192.168.100.130 (192.168.100.130), pst: 133.6. (133.6.) User Datagram Protocol, Src Port: 61825 (61825), pst Port: sxuptp (19540) Data (88 bytes))										
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	7e a7 54 24 08 00 45 00	£. 								
○ ぎ File: "F:\技術部\発表会・技報資料\2	2012 Profie: Default		.d							

図6. ブロードキャストストーム例

4.2 事例2 (Web サーバへのアクセス調査)

専攻Webサーバはグローバル側にルータを入れてWebサーバはローカルネットワークで運用している。そのルータより同一建物内の同一機器からの定期的アクセスログが排出されたが、ルータの機能として簡易的なログしか排出されずプロトコルのような詳細データは残っていない。そこで状況確認の為、図1で示したミラーリング用HUBを使用して1日中キャプチャを行いパケットの分析を行った。分析の際に今回はアクセス機器が特定されているので、そのMacアドレスをFilterで絞り込んだ結果、図7のようにARPの問い合わせが10分間隔程度で定期的に行われていると判明した。

📶 pordo-121121_00022_20121122130732 [Wireshark 1.8.0 (SVN Rev 43431 from /trunk-1.8)]	×							
Ele Edit View Go Capture Analyze Statistics Telephony Tools Internals Help								
E = = = = = = = = = = = = = = = = = = =								
Filter: eth.addr == 00:1b:21:								
No. Time Source Destination Protocol Length Info 626 13:08:55.014923000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 5831 13:19:43.567132000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 12159 13:29:34.477367000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 1231 13:40:05.654056000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 124178 13:50:11.174968000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 24778 13:50:11.174968000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6. 2478 13:50:11.174968000 IntelCor_ Broadcast ARP 60 Who has 133.6. ? Tell 133.6.								
(III) 	Þ							
<pre> i Frame 626: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0 i Ethernet II, Src: IntelCor(00:1b:21:, Dst: Broadcast (ff:ff:ff:ff:ff) Address: Broadcast (ff:ff:ff:ff:ff) Address: Broadcast (ff:ff:ff:ff:ff) Address: IntelCor</pre>								
0000 ff ff ff ff ff ff 00 1b 21 08 06 60 01								
⊘ ∑ Target IP address (arp.dst.proto_ipv4), 4 Profile: Default	đ							

図7. APR の問い合わせ

5 まとめ

ネットワークトラブルの対応はサーバログのチェックで原因が判明する場合が多いが、ログはサーバ等で 起きたトラブルの記録のためネットワーク全体に異常が起きた場合には判断が難しい。その場合にネットワ ークの状況を観察することによって原因が判明する場合があるので、ノートパソコン等にインストールした Wireshark でネットワーク全体か、機器単体のトラブルなのかの切り分けすることによって原因追求が早まる 場合もあると思われる。

Wireshark は現在専攻共通計算機室のトラブル時のみに利用しているが機能としては様々な分析が行える ため、Snort 等の監視ソフトとの併用でセキュリティチェックにも利用が可能と思われる。

参考文献

- [1] 竹下 恵, "パケットキャプチャ入門-LAN アナライザ Wireshark 活用術", リックテレコム, 2007.2.15
- [2] 竹下 恵, "パケットキャプチャ実践技術-Wireshark によろパケット解析応用編", リックテレコム, 2009.2.6
- [3] Chris Sanders 著,高橋基信・宮本久仁男監訳,"実践パケット解析-Wireshark を使ったトラブルシューティング第2判",オライリー・ジャパン,2012.11.21