AQM, TFRC, LEDBAT, uTP

Roman Dunaytsev

roman.dunaytsev@spbgut.ru

AQM

- RFC 2309 – Recommendations on Queue Management and Congestion Avoidance in the Internet
- “Internet meltdown” or “congestion collapse” in the mid of 1980
- The original fix for Internet meltdown was provided by Van Jacobson in 1986
  - TCP connections should “back off” during congestion

AQM (cont'd)

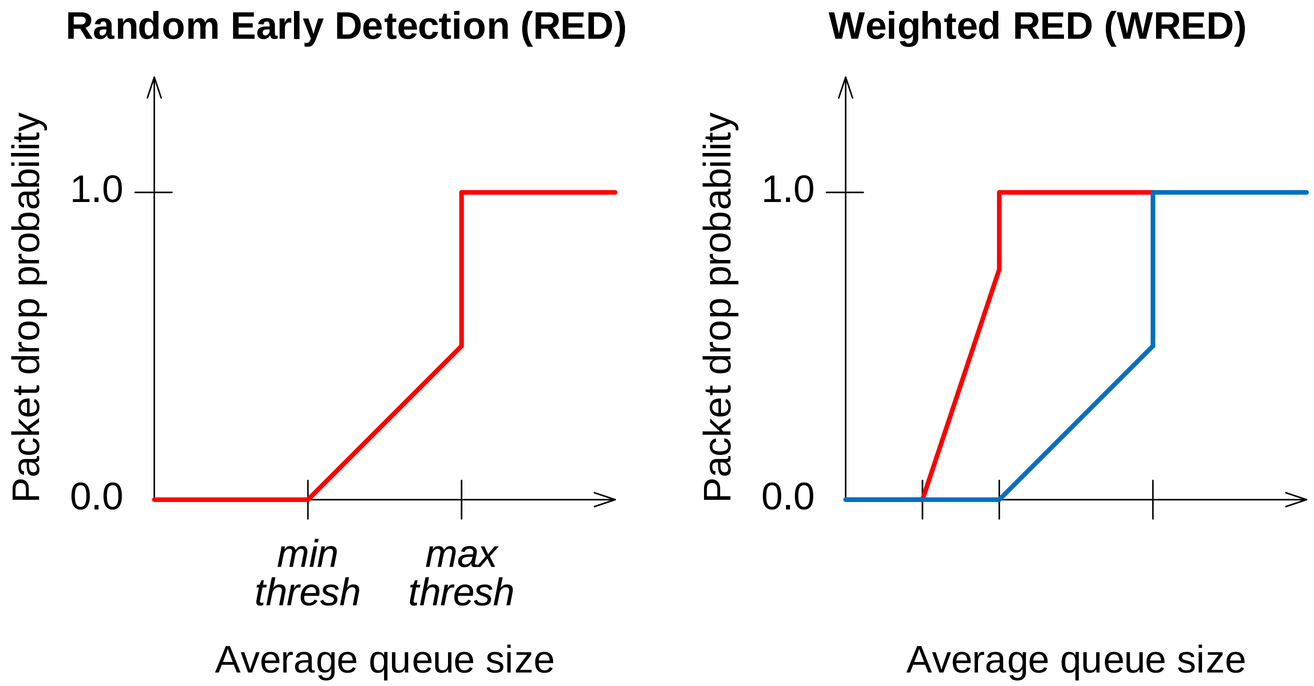
- The need for **Active Queue Management (AQM)**
  - Lock-out
  - Full queues

AQM (cont'd)

- AQM advantages for responsive flows:
  - Reduce number of packets dropped in routers
  - Provide lower-delay interactive service
  - Avoid lock-out behavior

AQM (cont'd)

- 2 main parts of **Random Early Detection (RED)**
  - Estimation of the average queue size
  - Packet drop decision



TFRC

- RFC 5348: TCP Friendly Rate Control (TFRC): Protocol Specification

- **TCP-Friendly Rate Control (TFRC)** is a congestion control mechanism designed for unicast flows competing with TCP traffic

- TFRC is not a protocol!

TFRC (cont’d)

- The need for TFRC:
  - Be reasonably fair when competing for bandwidth with TCP traffic
  - A much lower variation of throughput over time compared with TCP

- The penalty of having smoother throughput than TCP while competing fairly for bandwidth is that TFRC responds slower than TCP to changes in available bandwidth
  - Thus TFRC should only be used when the application has a requirement for smooth throughput

TFRC (cont’d)

- The throughput equation recommend for TFRC is a slightly simplified version of the throughput equation for TCP Reno (the PFTK model)

$$X = \frac{s}{R \cdot \sqrt{2 \cdot b \cdot p/3} + (t_{RTO} \cdot (3 \cdot \sqrt{3 \cdot b \cdot p/8} \cdot p \cdot (1+32 \cdot p^2)))}$$

LEDBAT

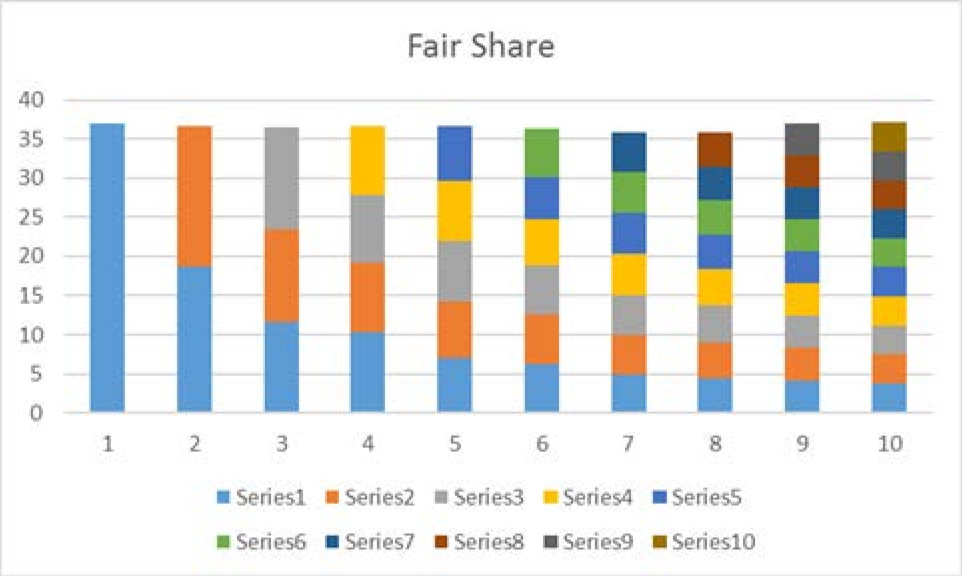
- RFC 6817 – Low Extra Delay Background Transport (LEDBAT)

- **Low Extra Delay Background Transport (LEDBAT)** is a delay-based congestion control algorithm

- LEDBAT is designed for use by background bulk-transfer applications to be no more aggressive than standard TCP congestion control

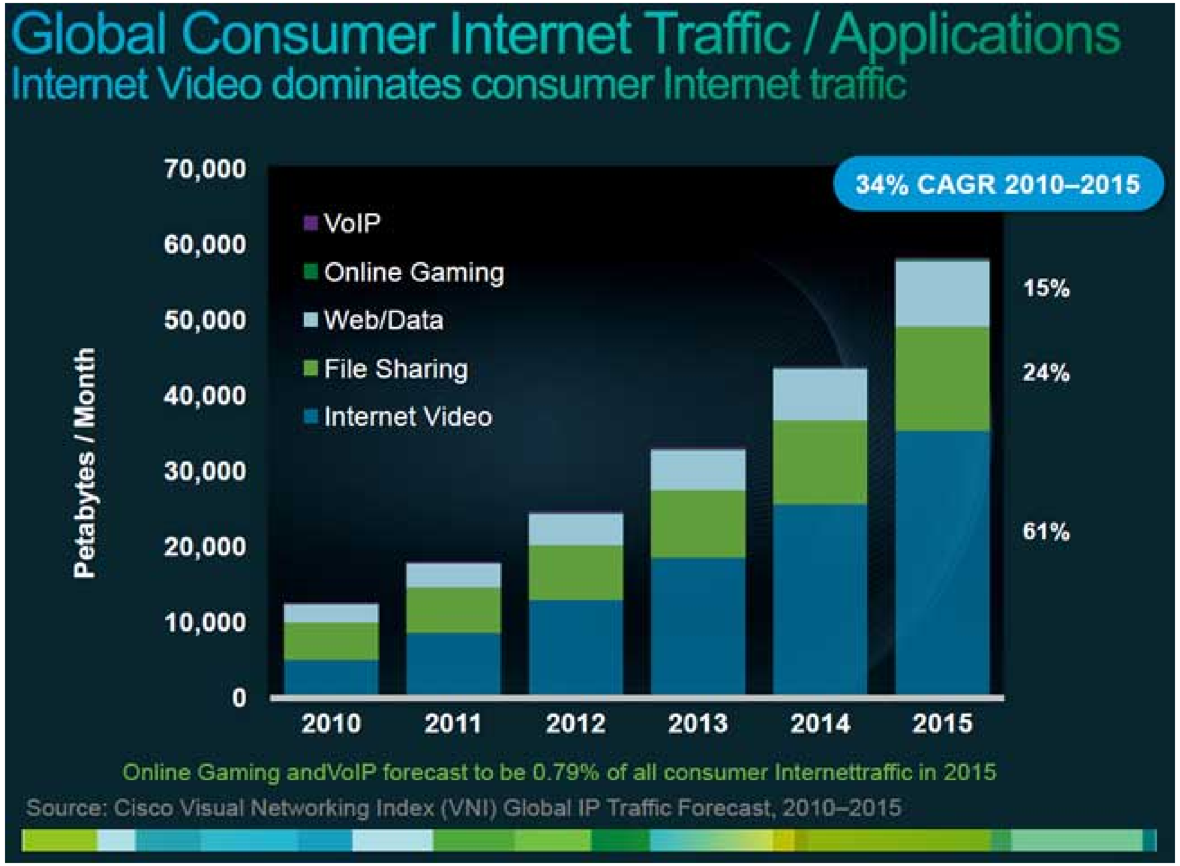
LEDBAT (cont'd)

- The need for LEDBAT
  - TCP seeks to share bandwidth at a bottleneck link equitably among flows competing at the bottleneck
  - However, not all applications seek an equitable share of network throughput



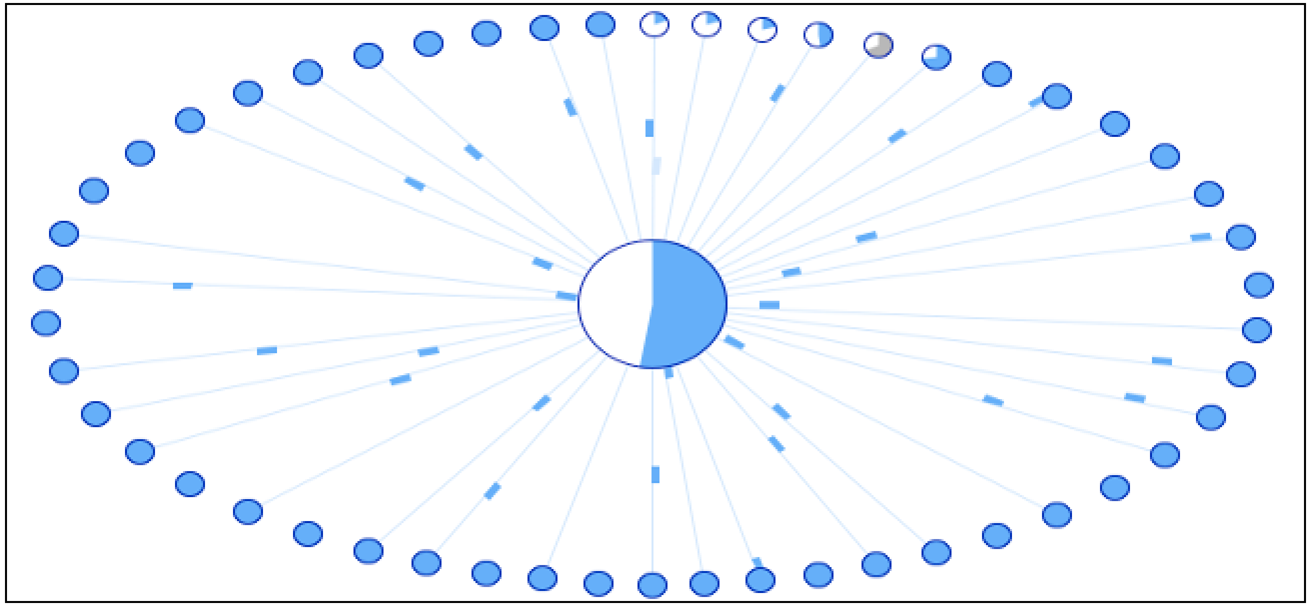
LEDBAT (cont'd)

- 2 types of applications:
  - Background (non-real-time)
  - Foreground (real-time)



LEDBAT (cont’d)

- **Standard TCP** implementations (TCP Reno, TCP NewReno, etc.) may be too aggressive for use with background applications



LEDBAT (cont’d)

- **Non-standard TCP** implementations (TCP Vegas, TCP FAST, etc.) are generally designed to achieve more, not less throughput than standard TCP, and often outperform TCP under particular network settings

- LEDBAT is designed to be no more aggressive than TCP
  - It is a “scavenger” congestion control mechanism
  - Less-than-Best-Effort (LBE)

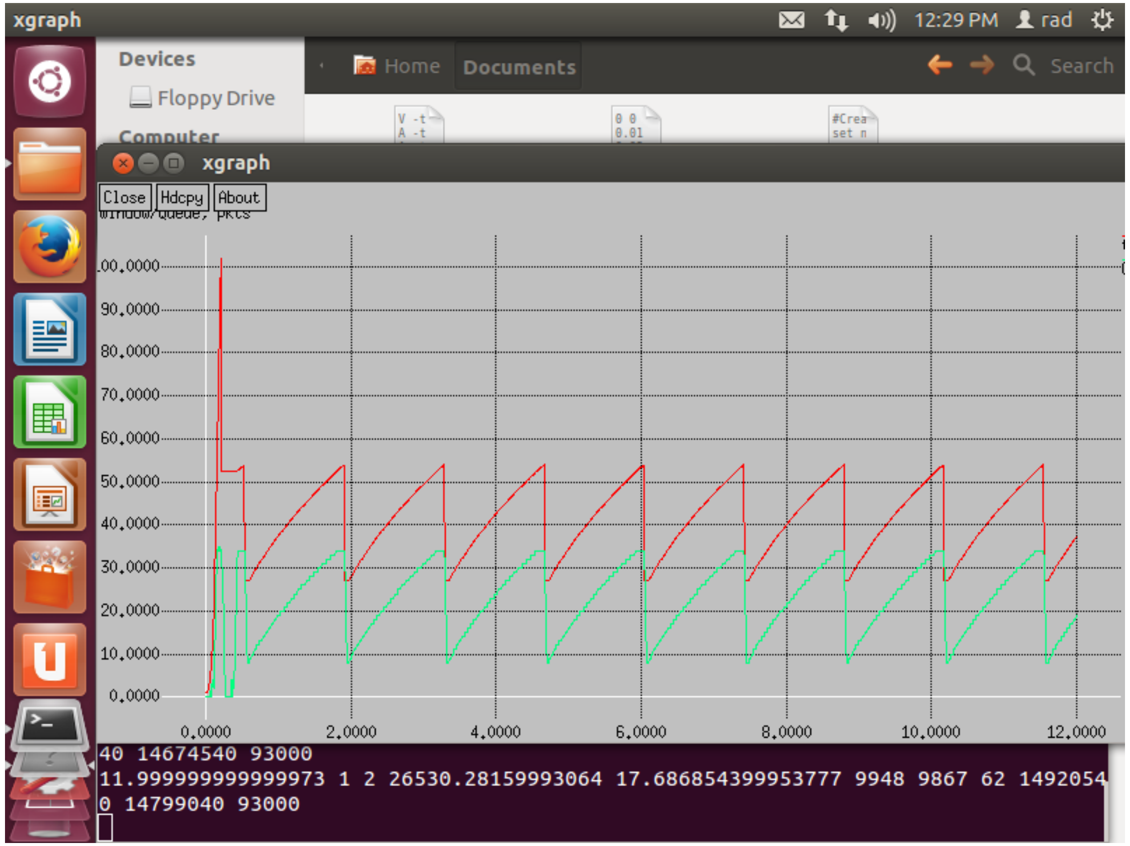
LEDBAT (cont’d)

- LEDBAT design goals:
  - To utilize end-to-end available bandwidth and to maintain low queueing delay when no other traffic is present
  - To add limited queuing delay to that induced by concurrent flows
  - To yield quickly to standard TCP flows that share the same bottleneck link

- LEDBAT can be used as:
  - A part of a transport protocol (UDP, RTP)
  - A part of an application (P2P, software updates)

LEDBAT (cont'd)

- Standard TCP sender increases its congestion window (cwnd) until a loss occurs



LEDBAT (cont’d)

- LEDBAT employs one-way delay measurements to estimate queueing delay
- When the estimated queueing delay is less than a predetermined target, LEDBAT infers that the network is not yet congested and increases its sending rate to utilize any spare capacity in the network
- When the estimated queueing delay becomes greater than the predetermined target, LEDBAT decreases its sending rate as a response to potential congestion in the network

LEDBAT (cont'd)

- Sender-side operation on ACK

```
current_delay = acknowledgement.delay
base_delay = min(base_delay, current_delay)
queuing_delay = current_delay - base_delay
off_target = (TARGET - queuing_delay) / TARGET
cwnd += GAIN * off_target * bytes_newly_acked * MSS / cwnd
```

uTP

- BEP 29 – uTorrent Transport Protocol
  - Also known as **Micro Transport Protocol** or **µTP**
  - BEP = BitTorrent Enhancement Proposal

- **uTorrent Transport Protocol (uTP)** design goals:
  - To not disrupt internet connections
  - Still utilize the unused bandwidth fully

- The problem is that DSL and cable modems typically have large buffers

uTP (cont'd)

- Traditional solution:
  - Cap the upload rate of the BitTorrent client to 80% of the up-link speed
  - 80% leaves some head room for interactive traffic

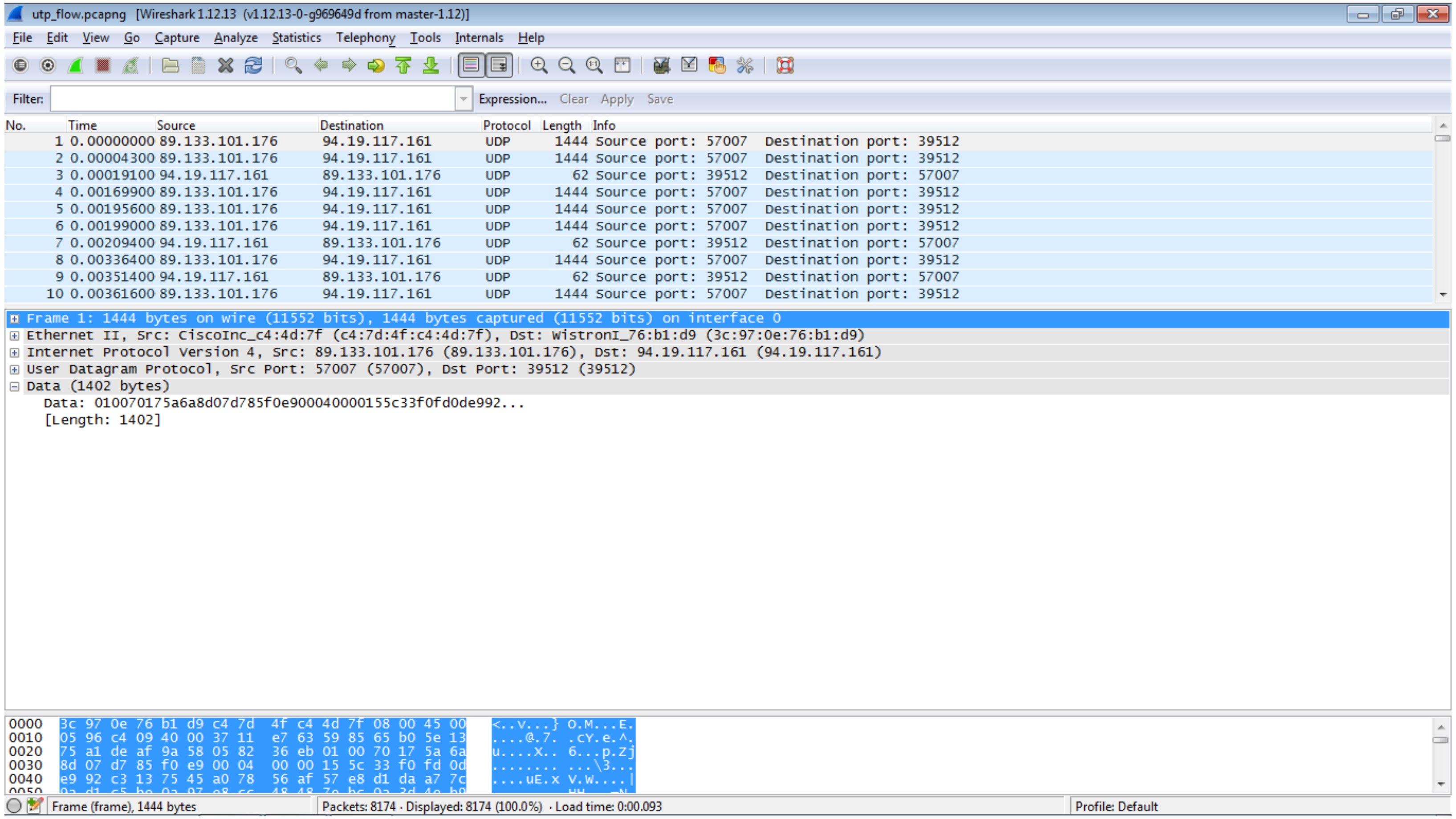
- Drawbacks with this solution:
  - The user needs to configure his/her BitTorrent client
  - The user needs to know his/her internet connection's upload speed
  - The headroom of 20% wastes bandwidth

uTP (cont'd)

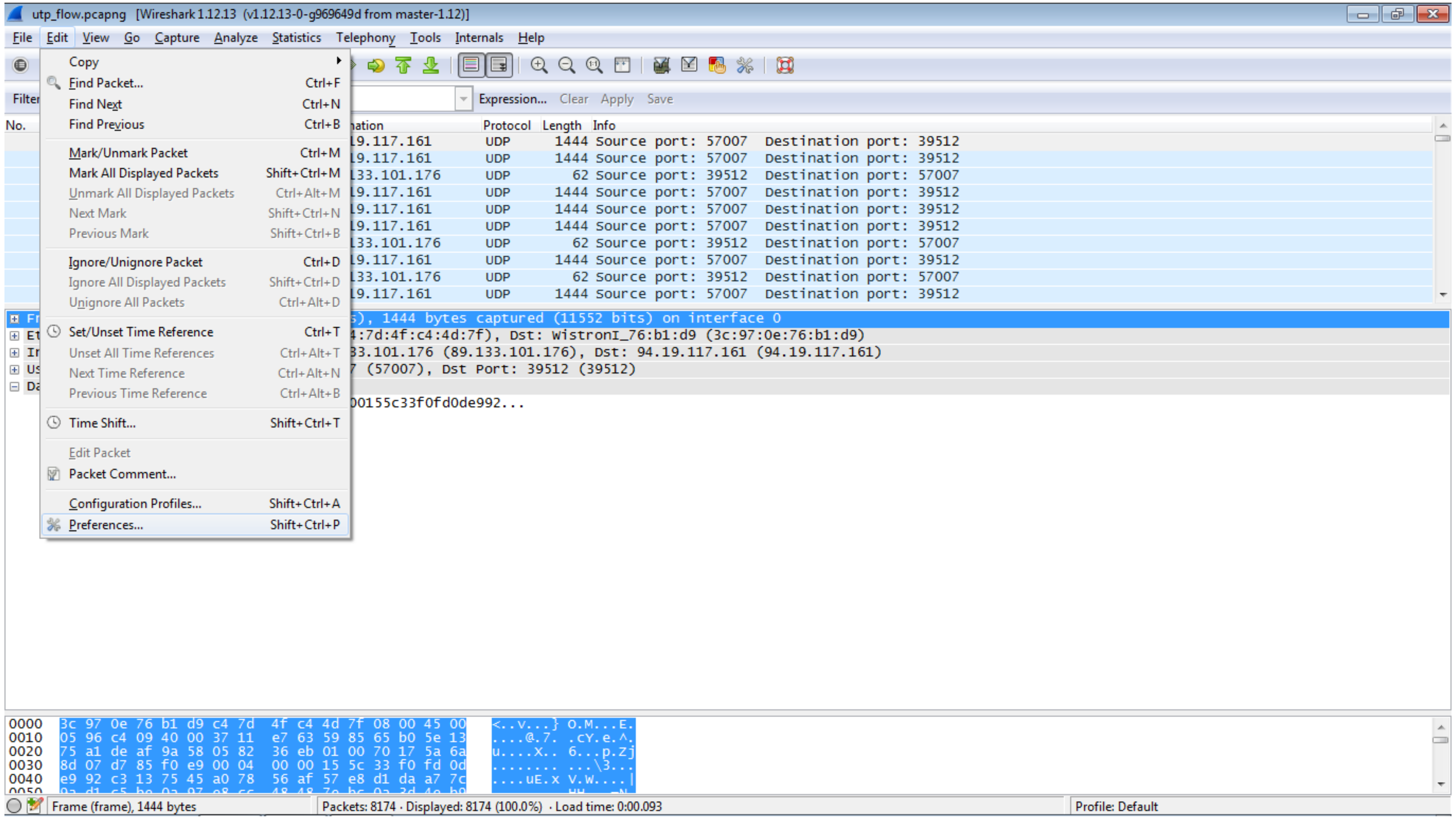
- uTP is a transport protocol layered on top of UDP

```
0       4       8               16              24              32
+-------+-------+---------------+---------------+---------------+
| type  | ver   | extension     | connection_id                 |
+-------+-------+---------------+---------------+---------------+
| timestamp_microseconds                                        |
+---------------+---------------+---------------+---------------+
| timestamp_difference_microseconds                             |
+---------------+---------------+---------------+---------------+
| wnd_size                                                      |
+---------------+---------------+---------------+---------------+
| seq_nr                        | ack_nr                        |
+---------------+---------------+---------------+---------------+
```

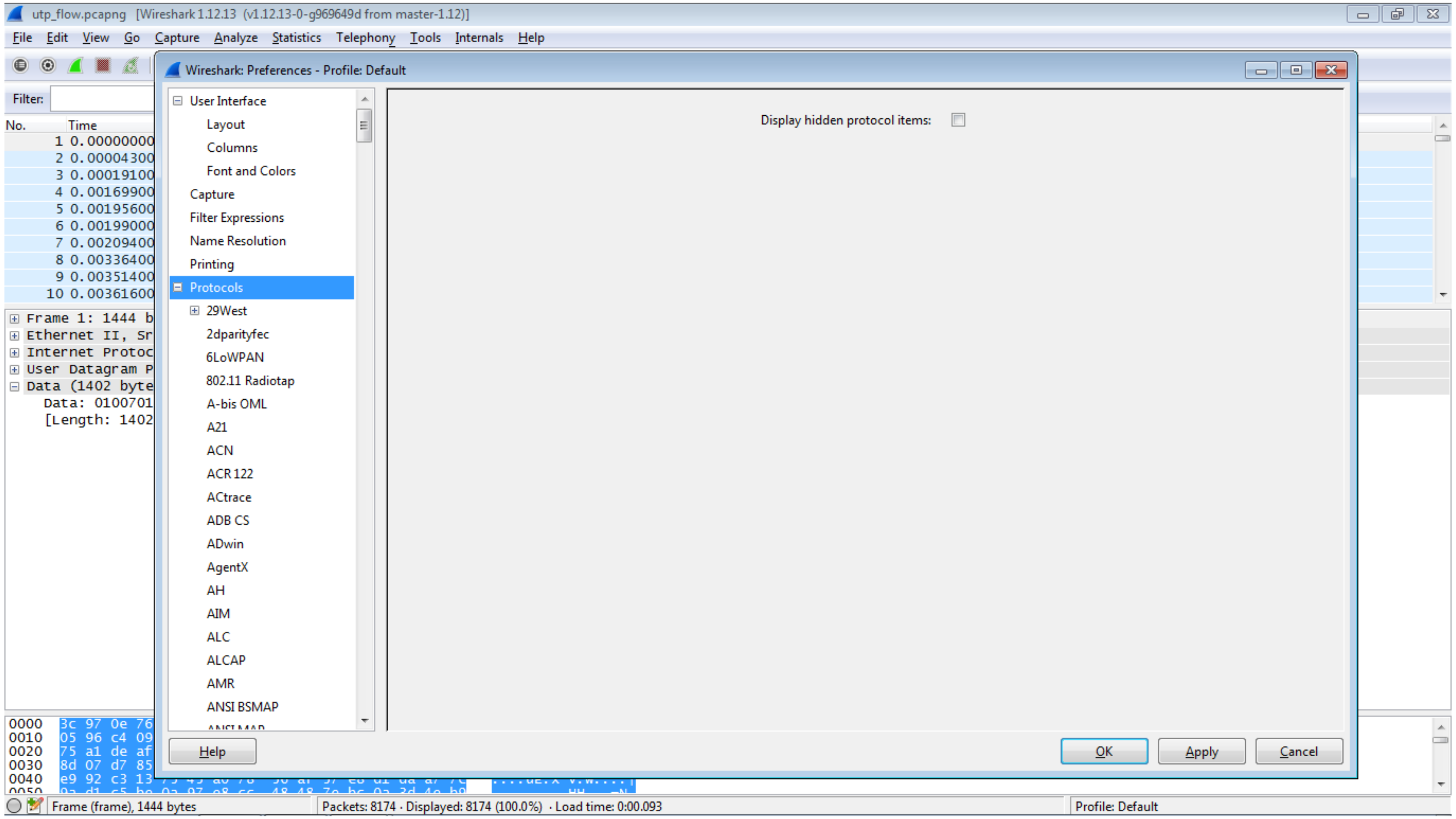
uTP (cont'd)


utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.00000000 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
2 0.00004300 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
3 0.00019100 94.19.117.161 89.133.101.176 UDP 62 Source port: 39512 Destination port: 57007
4 0.00169900 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
5 0.00195600 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
6 0.00199000 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
7 0.00209400 94.19.117.161 89.133.101.176 UDP 62 Source port: 39512 Destination port: 57007
8 0.00336400 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
9 0.00351400 94.19.117.161 89.133.101.176 UDP 62 Source port: 39512 Destination port: 57007
10 0.00361600 89.133.101.176 94.19.117.161 UDP 1444 Source port: 57007 Destination port: 39512
Frame 1: 1444 bytes on wire (11552 bits), 1444 bytes captured (11552 bits) on interface 0
Ethernet II, Src: CiscoInc_c4:4d:7f (c4:7d:4f:c4:4d:7f), Dst: WistronI_76:b1:d9 (3c:97:0e:76:b1:d9)
Internet Protocol Version 4, Src: 89.133.101.176 (89.133.101.176), Dst: 94.19.117.161 (94.19.117.161)
User Datagram Protocol, Src Port: 57007 (57007), Dst Port: 39512 (39512)
Data (1402 bytes)
Data: 010070175a6a8d07d785f0e900040000155c33f0fd0de992...
[Length: 1402]
0000 3c 97 0e 76 b1 d9 c4 7d 4f c4 4d 7f 08 00 45 00
0010 05 96 c4 09 40 00 37 11 e7 63 59 85 65 b0 5e 13
0020 75 a1 de af 9a 58 05 82 36 eb 01 00 70 17 5a 6a
0030 8d 07 d7 85 f0 e9 00 04 00 00 15 5c 33 f0 fd 0d
0040 e9 92 c3 13 75 45 a0 78 56 af 57 e8 d1 da a7 7c
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.093
Profile: Default

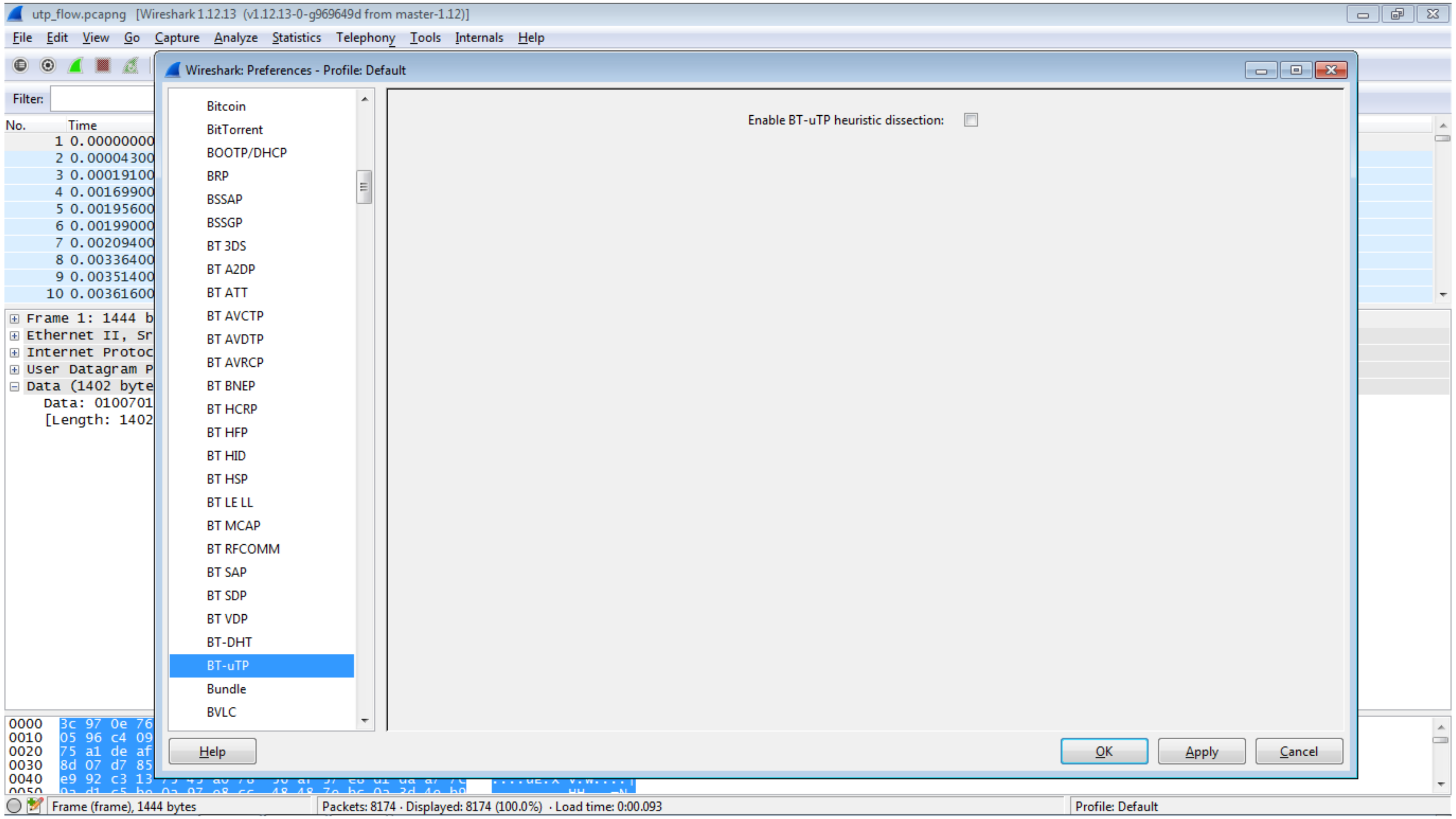
uTP (cont'd)


utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet Ctrl+M
Mark All Displayed Packets Shift+Ctrl+M
Unmark All Displayed Packets Ctrl+Alt+M
Next Mark Shift+Ctrl+N
Previous Mark Shift+Ctrl+B
Ignore/Unignore Packet Ctrl+D
Ignore All Displayed Packets Shift+Ctrl+D
Unignore All Packets Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Shift+Ctrl+T
Edit Packet
Packet Comment...
Configuration Profiles... Shift+Ctrl+A
Preferences... Shift+Ctrl+P
Expression... Clear Apply Save
Protocol Length Info
UDP 1444 Source port: 57007 Destination port: 39512
UDP 62 Source port: 39512 Destination port: 57007
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.093
Profile: Default

uTP (cont'd)

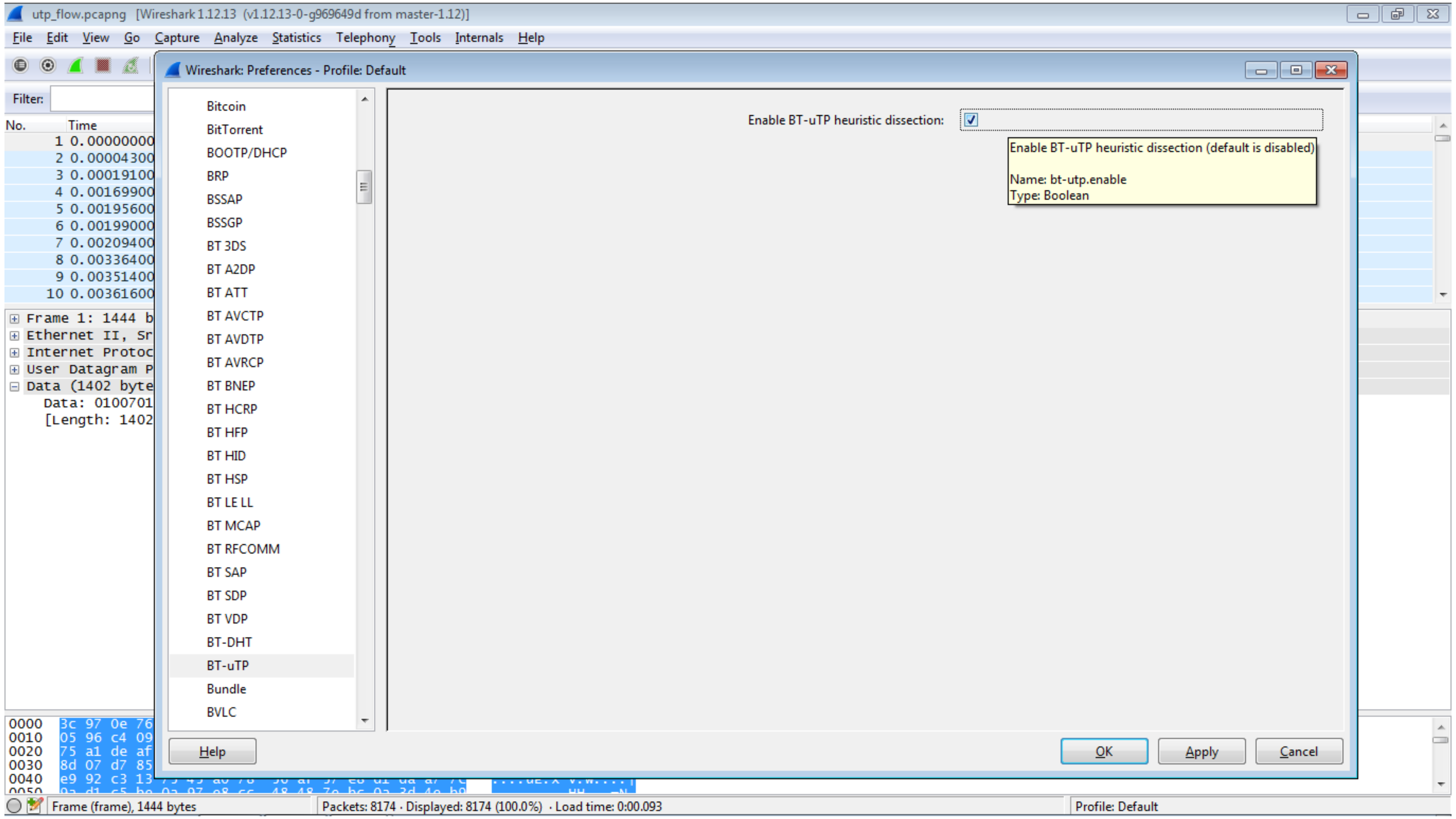

utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
No. Time
Frame 1: 1444 b
Ethernet II, Sr
Internet Protoc
User Datagram P
Data (1402 byte
Data: 0100701
[Length: 1402
Wireshark: Preferences - Profile: Default
User Interface
Layout
Columns
Font and Colors
Capture
Filter Expressions
Name Resolution
Printing
Protocols
29West
2dparityfec
6LoWPAN
802.11 Radiotap
A-bis OML
A21
ACN
ACR 122
ACtrace
ADB CS
ADwin
AgentX
AH
AIM
ALC
ALCAP
AMR
ANSI BSMAP
Display hidden protocol items:
Help
OK
Apply
Cancel
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.093
Profile: Default

uTP (cont'd)



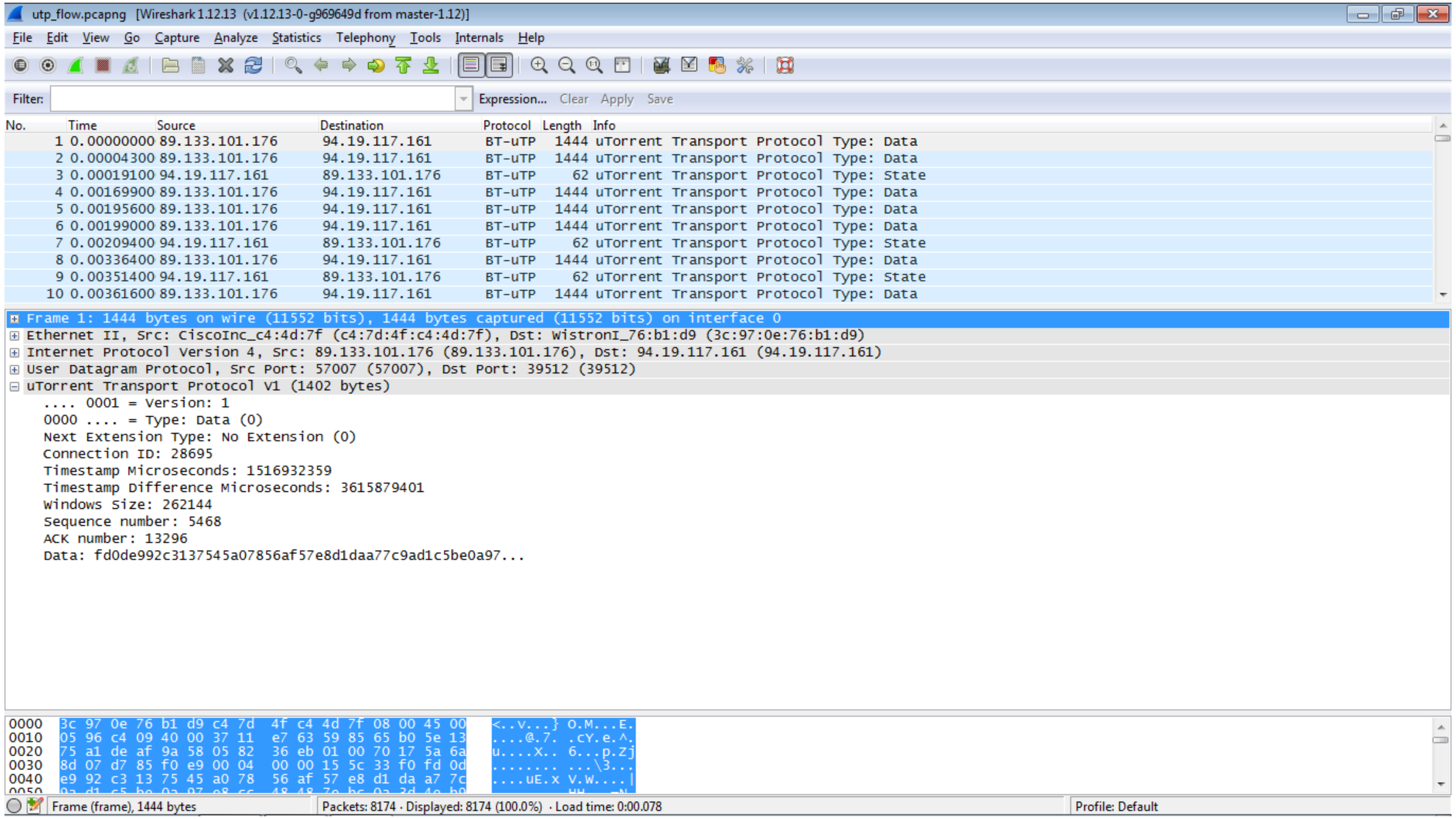
utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Wireshark: Preferences - Profile: Default
Filter:
No. Time
Bitcoin
BitTorrent
BOOTP/DHCP
BRP
BSSAP
BSSGP
BT 3DS
BT A2DP
BT ATT
BT AVCTP
BT AVDTP
BT AVRCP
BT BNEP
BT HCRP
BT HFP
BT HID
BT HSP
BT LE LL
BT MCAP
BT RFCOMM
BT SAP
BT SDP
BT VDP
BT-DHT
BT-uTP
Bundle
BVLC
Enable BT-uTP heuristic dissection:
Help
OK
Apply
Cancel
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.093
Profile: Default

uTP (cont'd)



utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Wireshark: Preferences - Profile: Default
Bitcoin
BitTorrent
BOOTP/DHCP
BRP
BSSAP
BSSGP
BT 3DS
BT A2DP
BT ATT
BT AVCTP
BT AVDTP
BT AVRCP
BT BNEP
BT HCRP
BT HFP
BT HID
BT HSP
BT LE LL
BT MCAP
BT RFCOMM
BT SAP
BT SDP
BT VDP
BT-DHT
BT-uTP
Bundle
BVLC
Enable BT-uTP heuristic dissection:
Enable BT-uTP heuristic dissection (default is disabled)
Name: bt-utp.enable
Type: Boolean
Help
OK
Apply
Cancel
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.093
Profile: Default

uTP (cont’d)


utp_flow.pcapng [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.00000000 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
2 0.00004300 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
3 0.00019100 94.19.117.161 89.133.101.176 BT-uTP 62 uTorrent Transport Protocol Type: State
4 0.00169900 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
5 0.00195600 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
6 0.00199000 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
7 0.00209400 94.19.117.161 89.133.101.176 BT-uTP 62 uTorrent Transport Protocol Type: State
8 0.00336400 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
9 0.00351400 94.19.117.161 89.133.101.176 BT-uTP 62 uTorrent Transport Protocol Type: State
10 0.00361600 89.133.101.176 94.19.117.161 BT-uTP 1444 uTorrent Transport Protocol Type: Data
Frame 1: 1444 bytes on wire (11552 bits), 1444 bytes captured (11552 bits) on interface 0
Ethernet II, Src: CiscoInc_c4:4d:7f (c4:7d:4f:c4:4d:7f), Dst: WistronI_76:b1:d9 (3c:97:0e:76:b1:d9)
Internet Protocol Version 4, Src: 89.133.101.176 (89.133.101.176), Dst: 94.19.117.161 (94.19.117.161)
User Datagram Protocol, Src Port: 57007 (57007), Dst Port: 39512 (39512)
uTorrent Transport Protocol V1 (1402 bytes)
.... 0001 = Version: 1
0000 .... = Type: Data (0)
Next Extension Type: No Extension (0)
Connection ID: 28695
Timestamp Microseconds: 1516932359
Timestamp Difference Microseconds: 3615879401
Windows Size: 262144
Sequence number: 5468
ACK number: 13296
Data: fd0de992c3137545a07856af57e8d1daa77c9ad1c5be0a97...
0000 3c 97 0e 76 b1 d9 c4 7d 4f c4 4d 7f 08 00 45 00
0010 05 96 c4 09 40 00 37 11 e7 63 59 85 65 b0 5e 13
0020 75 a1 de af 9a 58 05 82 36 eb 01 00 70 17 5a 6a
0030 8d 07 d7 85 f0 e9 00 04 00 00 15 5c 33 f0 fd 0d
0040 e9 92 c3 13 75 45 a0 78 56 af 57 e8 d1 da a7 7c
Frame (frame), 1444 bytes
Packets: 8174 · Displayed: 8174 (100.0%) · Load time: 0:00.078
Profile: Default

uTP (cont'd)

- LEDBAT vs. TCP