BitTorrent-Client

- Dokumentacja –

Autorzy:

Patryk Borowicz

Karolina Ciupalska

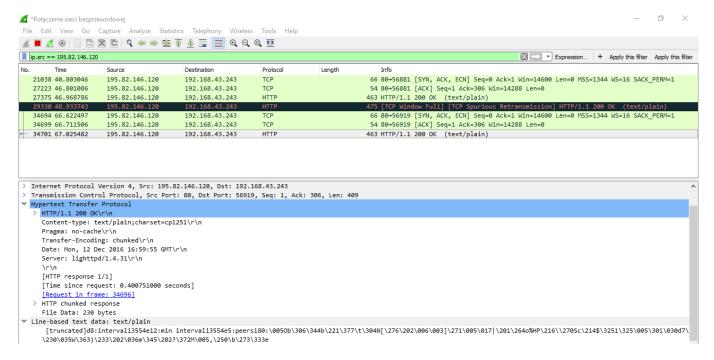
Katarzyna Pagórska

Radosław Weresa

Spis Treści

- 1. Podstawowa sesja w programie WireShark
- 2. Architektura aplikacji
- 3. Cele Projektu
- 4. Wytyczne GUI
- 5. Kluczowe pojęcia
- 6. Funkcje aplikacji
- 7. Wykorzystywane protokoły
- 8. Harmonogram prac

1. Podstawowa sesja w programie WireShark



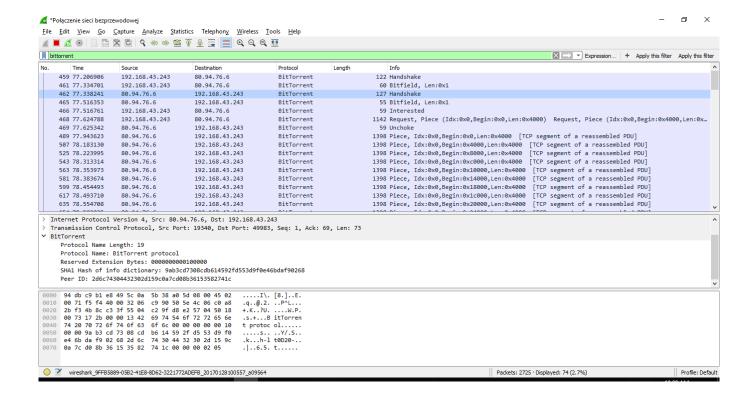
Odpowiedź trackera HTTP zawiera informacje na temat adresów IP peerów posiadających dany torrent oraz dane na temat możliwego czasu odpytywania trackera.

Odpowiedź trackera:

d8:intervali3554e12:min

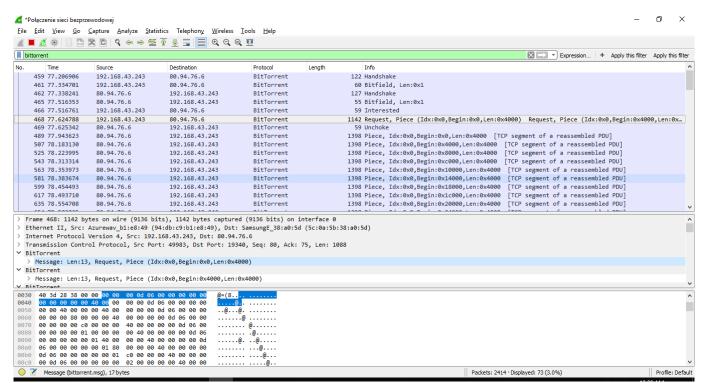
 $intervali3554e5:peers180:\\0050b\\306\\344b\\221\\377\\t\\304N[\\276\\202\\006\\003]\\271\\005\\017\\|\\201\\204\\325R>\\274\\006\\303z\\326KR\\0\\230\\035W\\363)\\233\\202\\036m\\345\\202?\\372M\\005,\\250\\b\\273\\333e$

Kolorem zielonym jest oznaczona lista adresów IP i portów peerów dla danego torrenta.



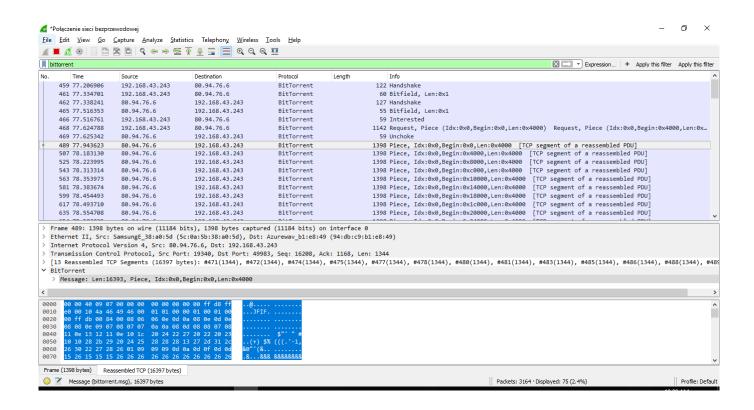
Operacja uwierzytelnienia – wysłanie wiadomości HandShake.

Wiadomość zawiera informacje takie jak: nazwa protokołu, Id Klienta, hash metadanych pliku torrent.



Wysłanie zapytania do peera o daną część torrenta.

Odebranie części od peera.



2. Architektura aplikacji

Hierarchy For All Packages

Package Hierarchies:

- bencode,
- bencode.types,
- bittorrent

Class Hierarchy

- o java.lang.Object
 - o java.util.AbstractCollection<E> (implements java.util.Collection<E>)
 - java.util.AbstractList<E> (implements java.util.List<E>)
 - java.util.ArrayList<E> (implements java.lang.Cloneable, java.util.List<E>, java.util.RandomAccess, java.io.Serializable)
 - o bencode.types.<u>BList</u> (implements bencode.<u>BElement</u>)
 - java.util.AbstractMap<K,V> (implements java.util.Map<K,V>)
 - java.util.HashMap<K,V> (implements java.lang.Cloneable, java.util.Map<K,V>, java.io.Serializable)
 - bencode.types.<u>BMap</u> (implements bencode.<u>BElement</u>)
 - o bencode.types.<u>BNumber</u> (implements bencode.<u>BElement</u>)
 - o bencode.BReader
 - bencode.types.BString (implements bencode.BElement)
 - bencode.types.BStringComparator (implements java.util.Comparator<T>)
 - bittorrent.Client
 - o bittorrent. ClientIdentifier
 - java.awt.Component (implements java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable)
 - java.awt.Container
 - o javax.swing.JComponent (implements java.io.Serializable)
 - javax.swing.JPanel (implements javax.accessibility.Accessible)
 - o bittorrent.PiecesPanel

- o java.awt.Window (implements javax.accessibility.Accessible)
 - o java.awt.Frame (implements java.awt.MenuContainer)
 - javax.swing.JFrame (implements javax.accessibility.Accessible, javax.swing.RootPaneContainer, javax.swing.WindowConstants)
 - o bittorrent.BitfieldTest
 - bittorrent.<u>MainWindow</u>
- o bittorrent. <u>Handshake</u>
- o bittorrent.<u>HTTPResponse</u>
- o bittorrent.<u>Peer</u>
- o bittorrent. PeerRequest
- o bittorrent. PeerResponse
- bittorrent.<u>Piece</u> (implements java.lang.Comparable<T>)
- o bittorrent.<u>RequestedBlock</u>
- o bittorrent. Share Service
- o bittorrent. Torrent
- o bittorrent. TorrentItem
- o bittorrent.<u>TrackerBase</u>
 - o bittorrent.<u>TrackerHTTP</u>
 - o bittorrent.<u>TrackerUDP</u>
- bittorrent.<u>TrackerUDP</u>
- o bittorrent. UDPRequest
- o bittorrent. <u>UDPResponse</u>

Interface Hierarchy

o bencode.<u>BElement</u>

Enum Hierarchy

- o java.lang.Object
 - o java.lang.Enum<E> (implements java.lang.Comparable<T>, java.io.Serializable)
 - o bittorrent.<u>Client.LOAD_RESULT</u>
 - o bittorrent. Peer. Status
 - o bittorrent. <u>PeerResponse. MessageType</u>
 - o bittorrent.<u>TrackerUDP .Status</u>
 - o bittorrent. <u>UDPResponse. Action</u>

3. Cele Projektu

- Odczytanie metadanych z plików torrent
- Równoległe pobieranie wielu plików
- Wstrzymywanie i wznawianie pobierania plików

4. Wytyczne GUI

- Aktualna szybkość pobierania/wysyłania plików
- Pozostały czas do zakończenia pobierania
- Wartość procentowa postępu pobrania
- Informacje o fragmentach pobieranego pliku (ilość i dostępność)
- Dane peer adres IP, szybkość pobierania/wysyłania, numer portu i nazwa klienta BitTorrent

5. Kluczowe pojęcia

Peer – użytkownik pobierający i udostępniający plik

Tracker – serwer, który przekazuje informacje (adresy IP) o pobierających i udostępniających pliki użytkownikach

Metadane – dane o plikach i folderach, przeznaczonych do udostępniania

6. Funkcje aplikacji

- Wczytywanie i dekodowanie pliku torrent
- Wymiana informacji z trackerem odnośnie listy użytkowników, którzy posiadają dany plik
- Komunikacja z innymi użytkownikami (peerami)
- Śledzenie postępu pobierania plików

7. Wykorzystywane protokoły

7.1 Bencoding

- Łańcuch znaków <liczba znaków>:<wyrażenie>
- Liczba całkowita i<liczba>e
- Lista l<elementy listy>e
- Słownik d<elementy słownika>e

Przykład: d8:announce22:http://bt.t-ru.org/anne

7.2 Tracker HTTP/HTTPS Protocol

http://bt.tru.org/ann?info_hash=%C9%8AT%C3%13%A4B%CDZ%8 5x%B1%29e%5BX_%05%C6%7F&peer_id=ABCDEFGHIJ KLMNOPQRST&port=6882&uploaded=0&downloaded= 0&left=0&numwant=50&event=started

7.3 Peer wire protocol (TCP)

Peer wire protocol – umożliwia przesyłanie fragmentów plików. Klient BitTorrent musi dostarczyć informację do każdego połączenia ze zdalnym peerem.

Statusy:

Choked – oznacza, że żadne zapytania nie są uwzględnianie dopóki klient nie będzie odblokowany (unchoked).

(Nie można pobierać żadnych fragmentów pliku).

Interested – peer jest zainteresowany fragmentami, które posiada klient, ale będzie wysyłać zapytanie o nie, dopiero kiedy klient je odblokuje (unchoked).

8. Harmonogram prac

<u>Tydzień 1</u> – zapoznanie się ze specyfikacją tematu, zrozumienie celu projektu

<u>Tydzień 2-3</u> – dobór narzędzi, wstępne projektowanie architektury

<u>Tydzień 4-5</u> - programowanie podstawowych części programu

<u>Tydzień 6</u> – prezentacja postępu prac

<u>Tydzień 7-końca</u> – część programistyczna aplikacji