NBP

NBP-NETWORK

浅探百度网盘下载逻辑

NBP-NETWORK

01 基本准备

02 网页端 VS 客户端

03 SVIP VS PoorMan

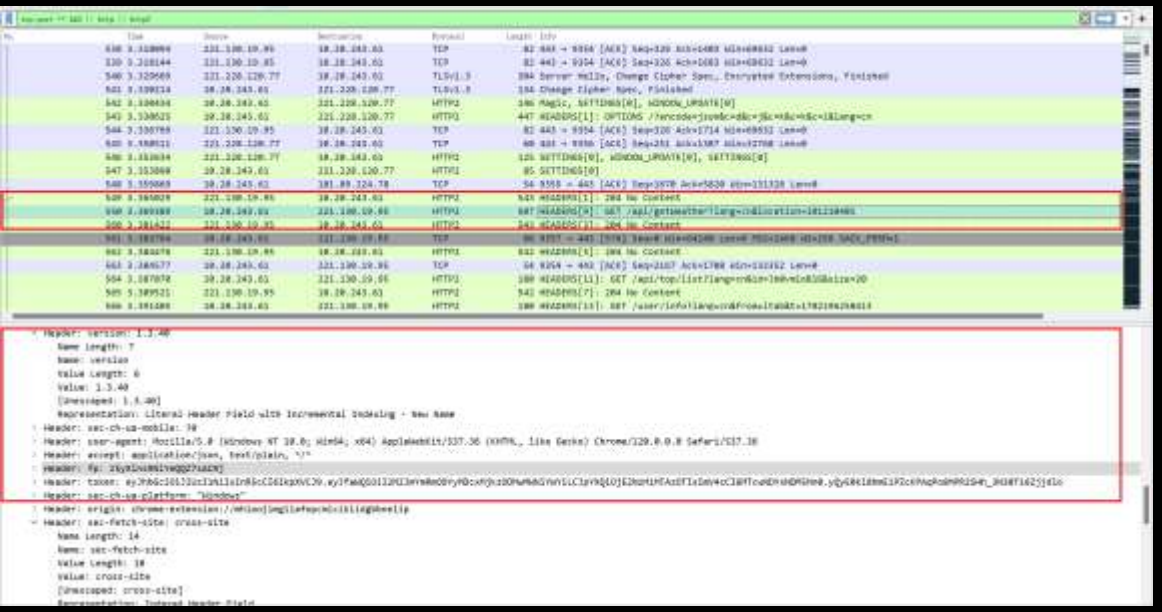
04 买一送二 -> RESTful框架 双栈下载

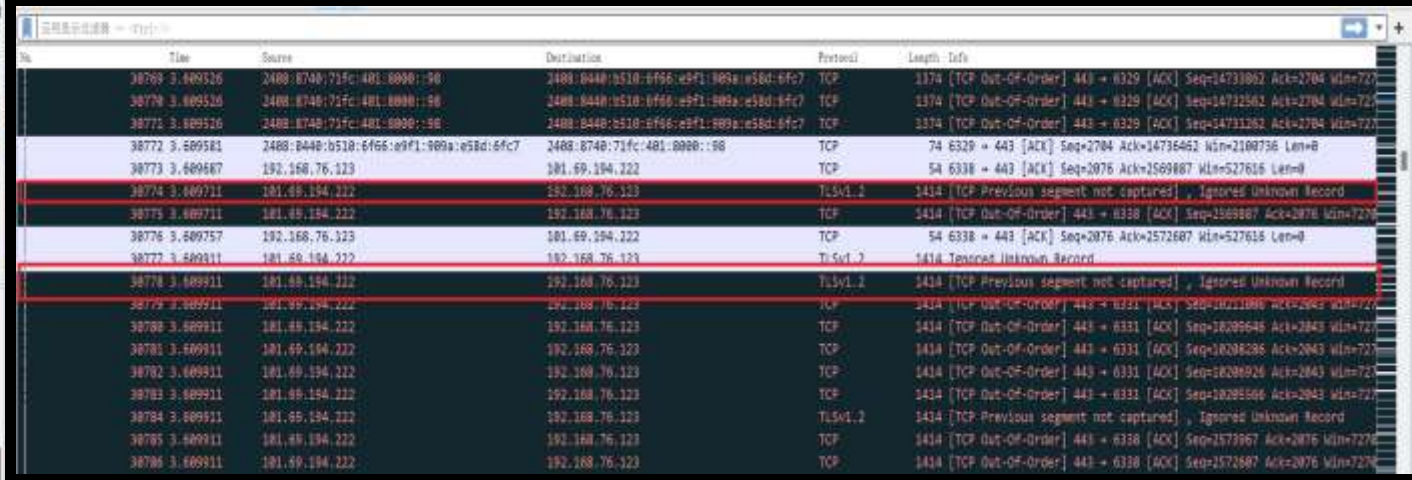
NBP-NETWORK

01 基本准备

优雅使用wireshark:
https://zhuanlan.zhihu.com/p/366693777

对PC端抓包:
https://www.cnblogs.com/gancuimian/p/14010628.html





NBP-NETWORK

01 基本准备

优雅使用wireshark:
https://zhuanlan.zhihu.com/p/36669377

对PC端抓包:
https://www.cnblogs.com/gancuimian/p/14010628.html

Wireshark

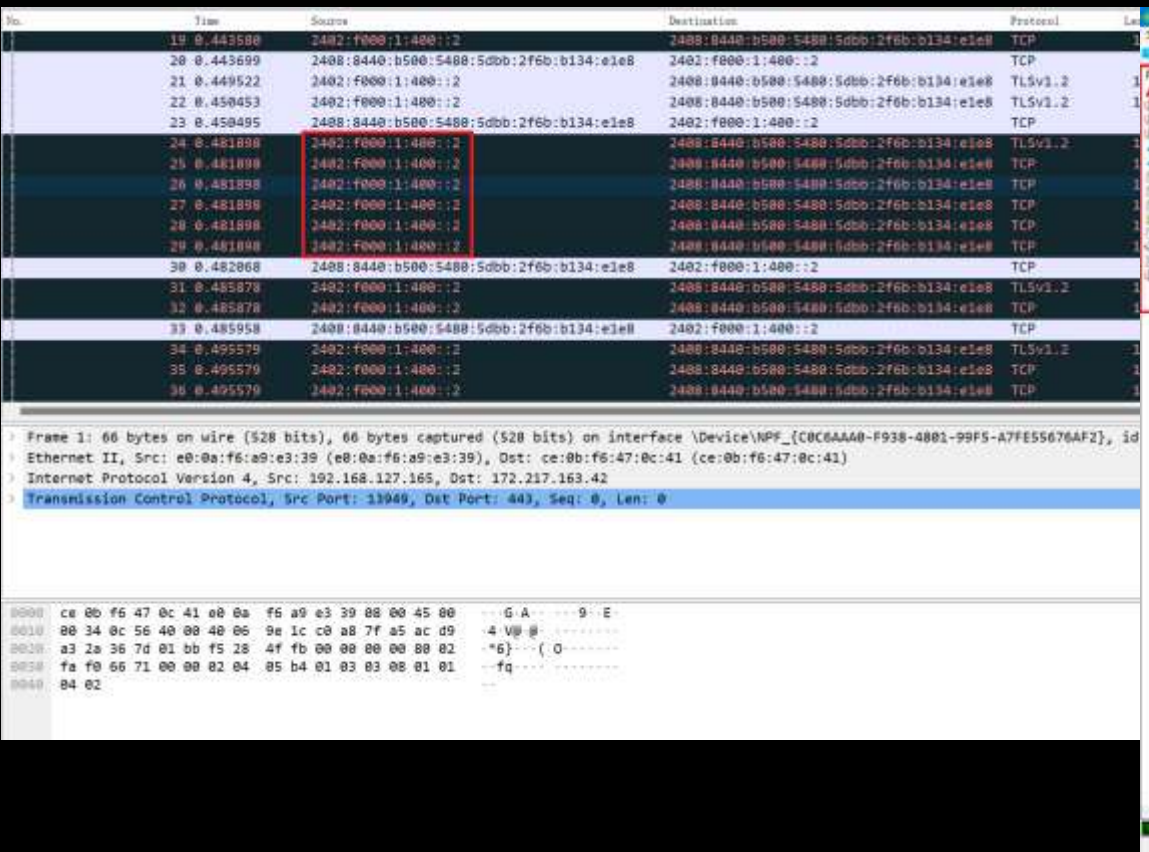
Fiddler+proxifier

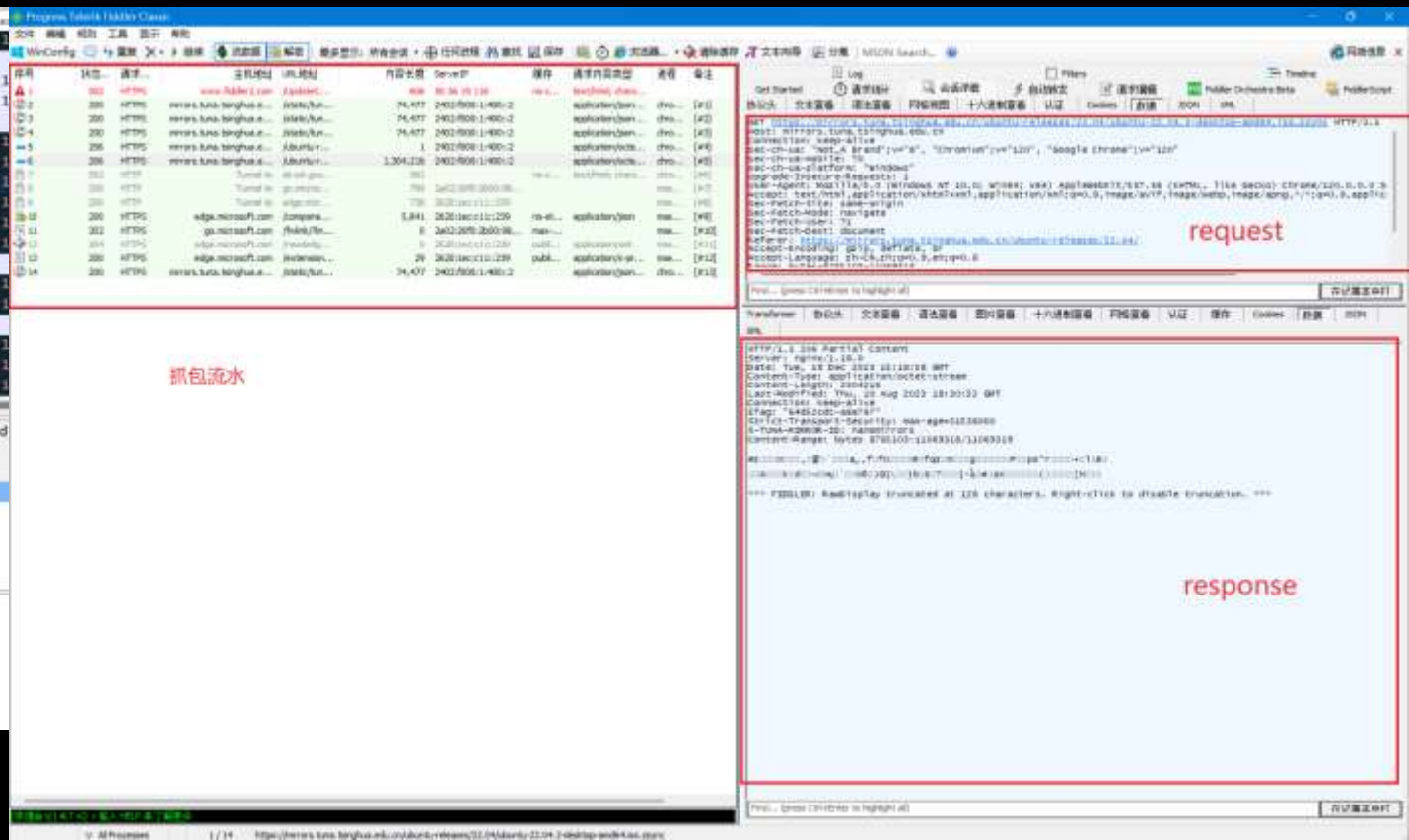
NBP-NETWORK

01 基本准备

优雅使用wireshark:
https://zhuanlan.zhihu.com/p/36669377

对PC端抓包:
https://www.cnblogs.com/gancuimian/p/14010628.html





NBP-NETWORK

02 网页端 VS 客户端

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

03 从百度客户端登录并下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

12对比，观察百度网盘（PanBD）网页端和一般下载网站的下载区别

13对比，观察PanBD网页端和客户端的下载区别

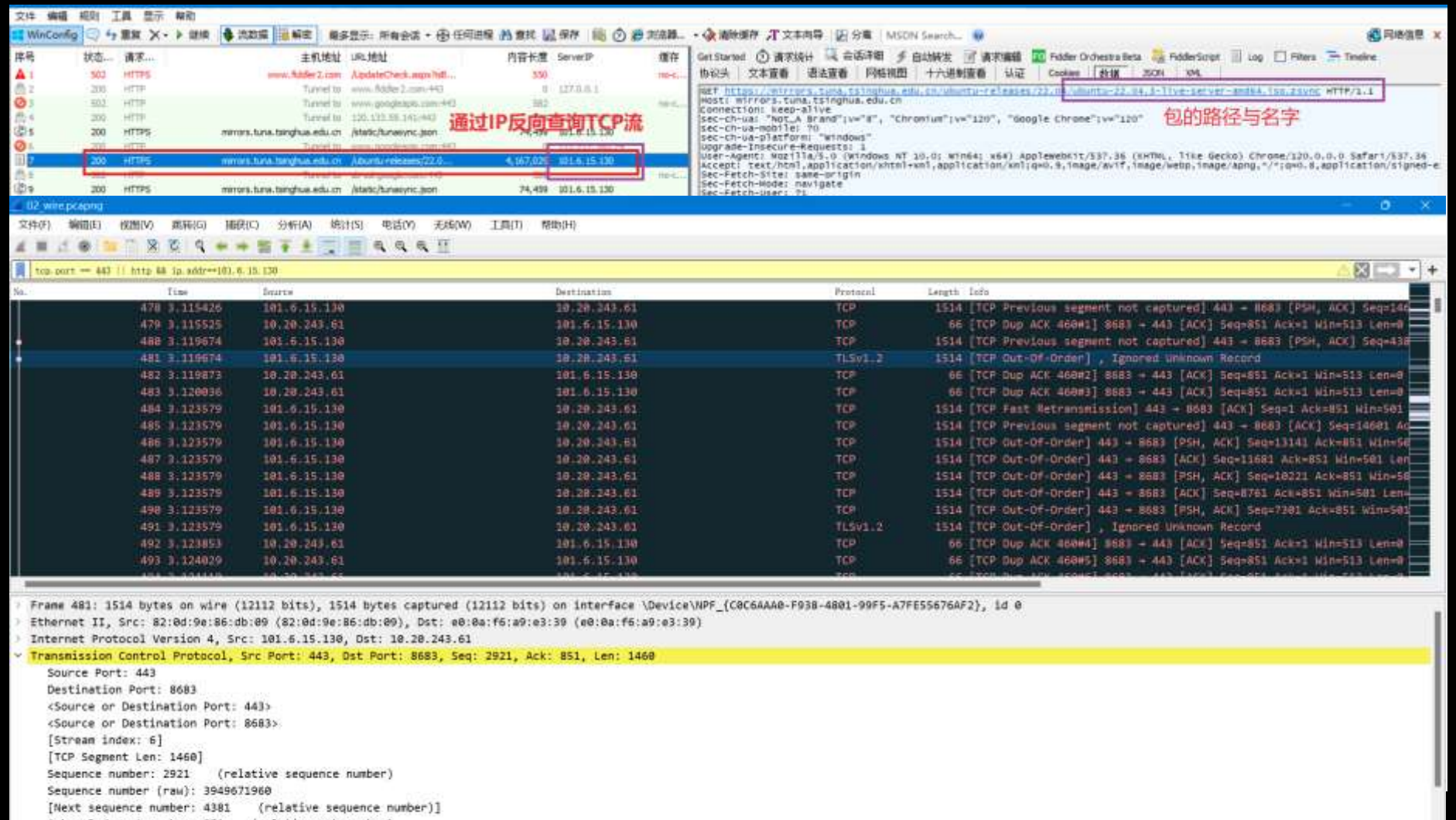
NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

TCP下载证明：
以包02为例
找到下载的包的报文，
通过IP反向观察wireshark数据流



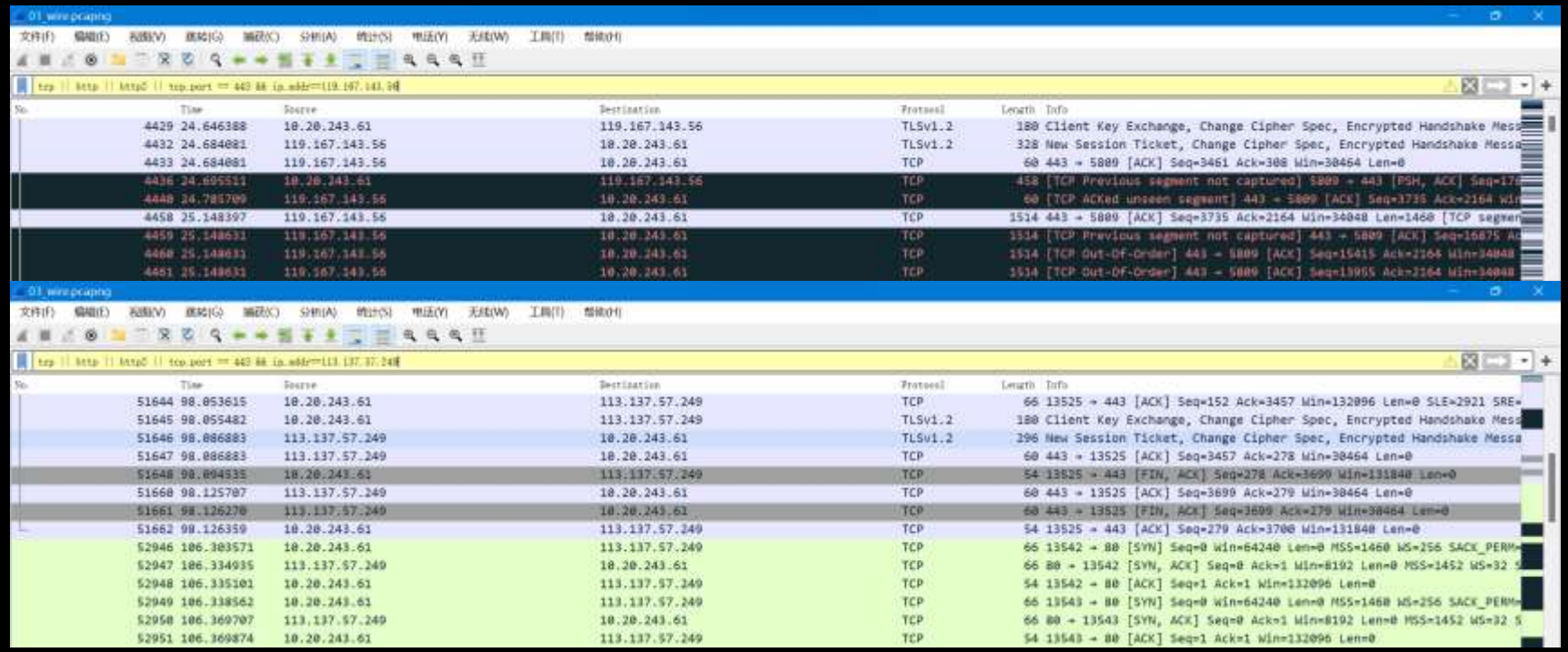
NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

TCP下载证明：
对01包和03包做相同处理，
结果一致
证毕



NBP-NETWORK

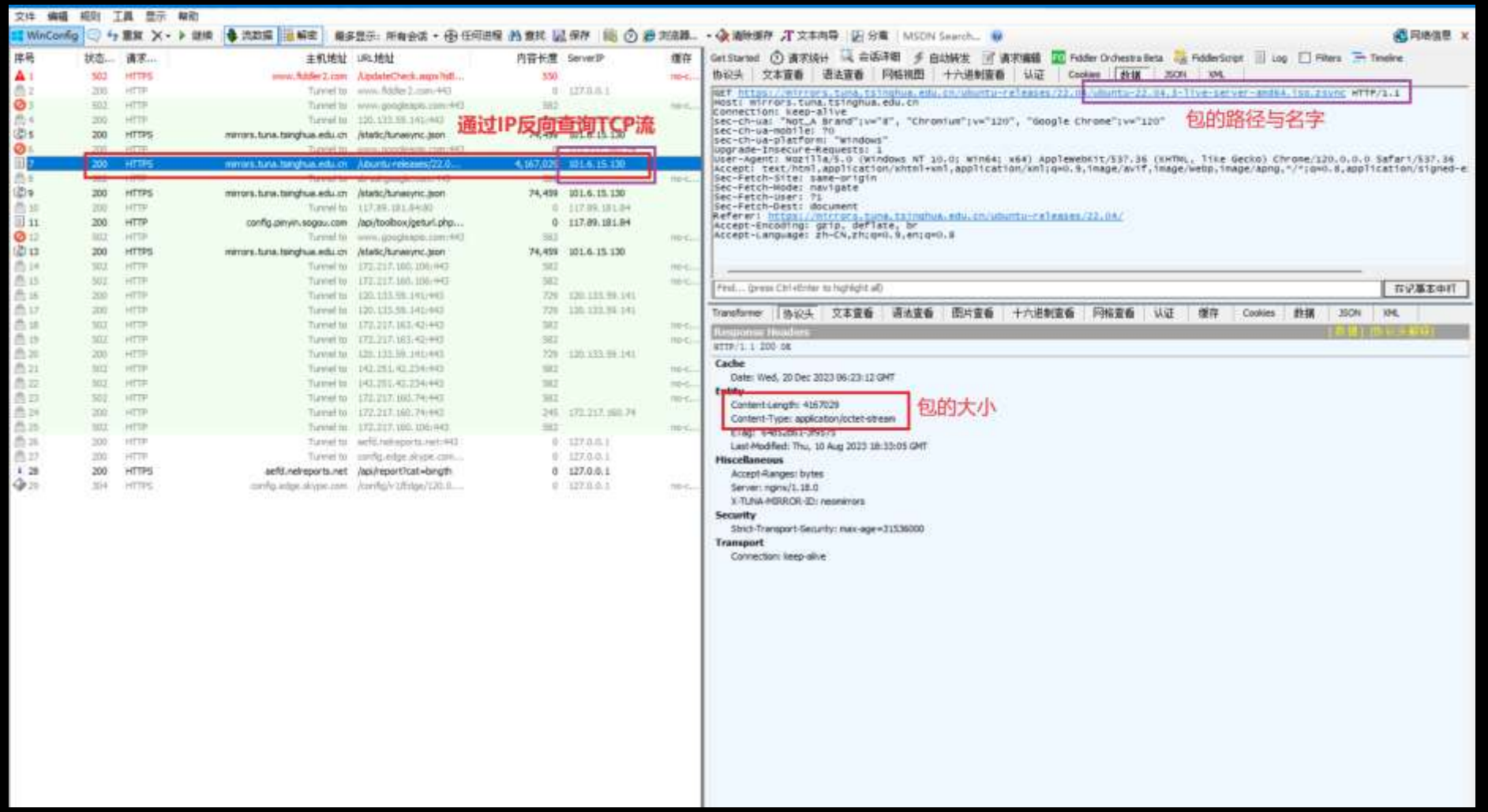
01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

先看一般网站的下载，可以发现来往报文
http首部相对简洁，自定义字段并不多，也
没有cookies或者request URL
自定义字段在图中给出了说明
当然，这只是个例网站，仍有以偏概全之嫌



NBP-NETWORK

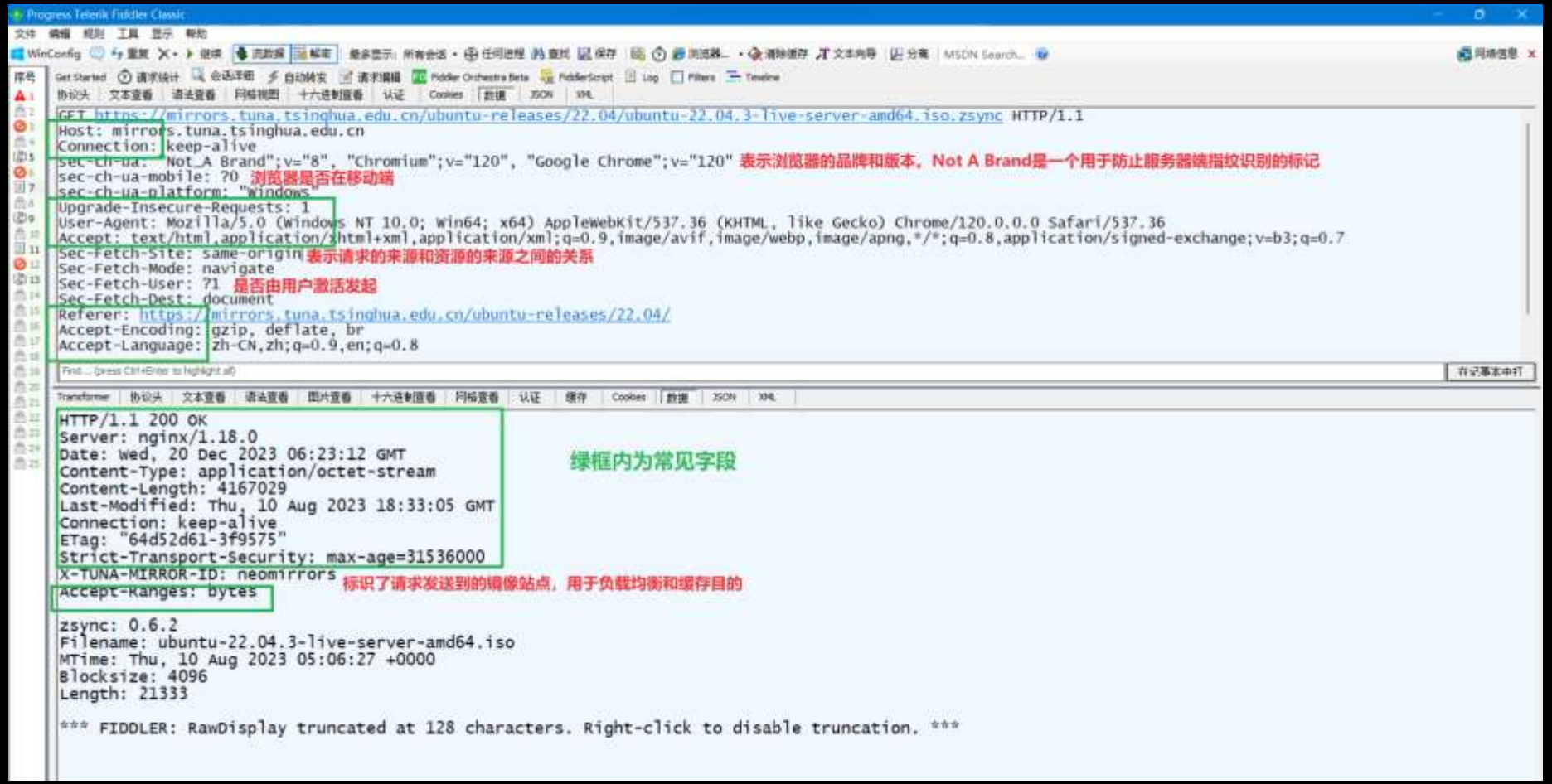
01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

先看一般网站的下载，可以发现来往报文http首部相对简洁，自定义字段并不多，也没有cookies或者request URL
自定义字段在图中给出了说明
当然，这只是个例网站，仍有以偏概全之嫌



NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

再看PanBD，我们发现复杂度激增，连接请求逻辑和自定义首部数量与一般下载网站截然不同
我们将从下载服务器开始，尽可能的梳理出网页端下载时的服务器请求链与特殊字段

从右图可见，下载服务器并不是pan.baidu.com, 他们似乎都是baidupcs.com的子域名服务器，命名也都有迹可循

同时，不像一般网站的下载，网盘并不是在点击下载时直接跳转到该服务器，我们也看见了如d.pcs.baidu.com/mbd.baidu.com等等服务器，他们的作用也是我们感兴趣的地方

下面， 我们以d.pcs.baidu.com这一服务器为例，带大家从http首部剖析其作用

NBP-NETWORK

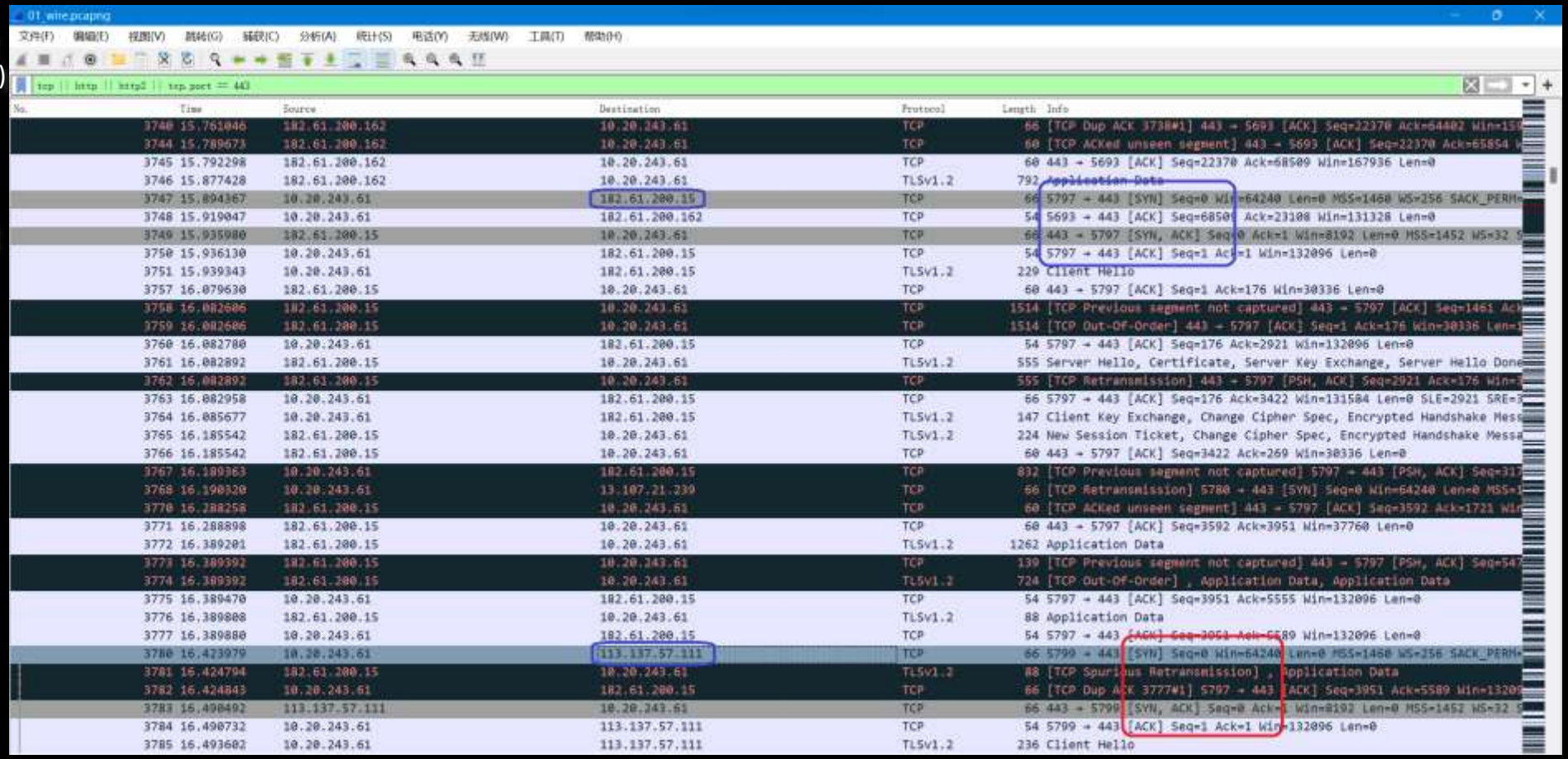
01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

让我们先回到wireshark的抓包流，将IP锁定为下载服务器的IP（这里指xafj-ct11服务器）可以看见下方 113段IP为下载服务器。而在此之前紧跟着的是IP182.61.200.15的tcp握手与数据传输。图中没有显示的是在于182交流完毕后主机马上就利用DNS寻找113服务器，说明d.pcs.baidu.com可能是一个响应传输请求的服务器
让我们回到fiddler去实锤



NBP-NETWORK

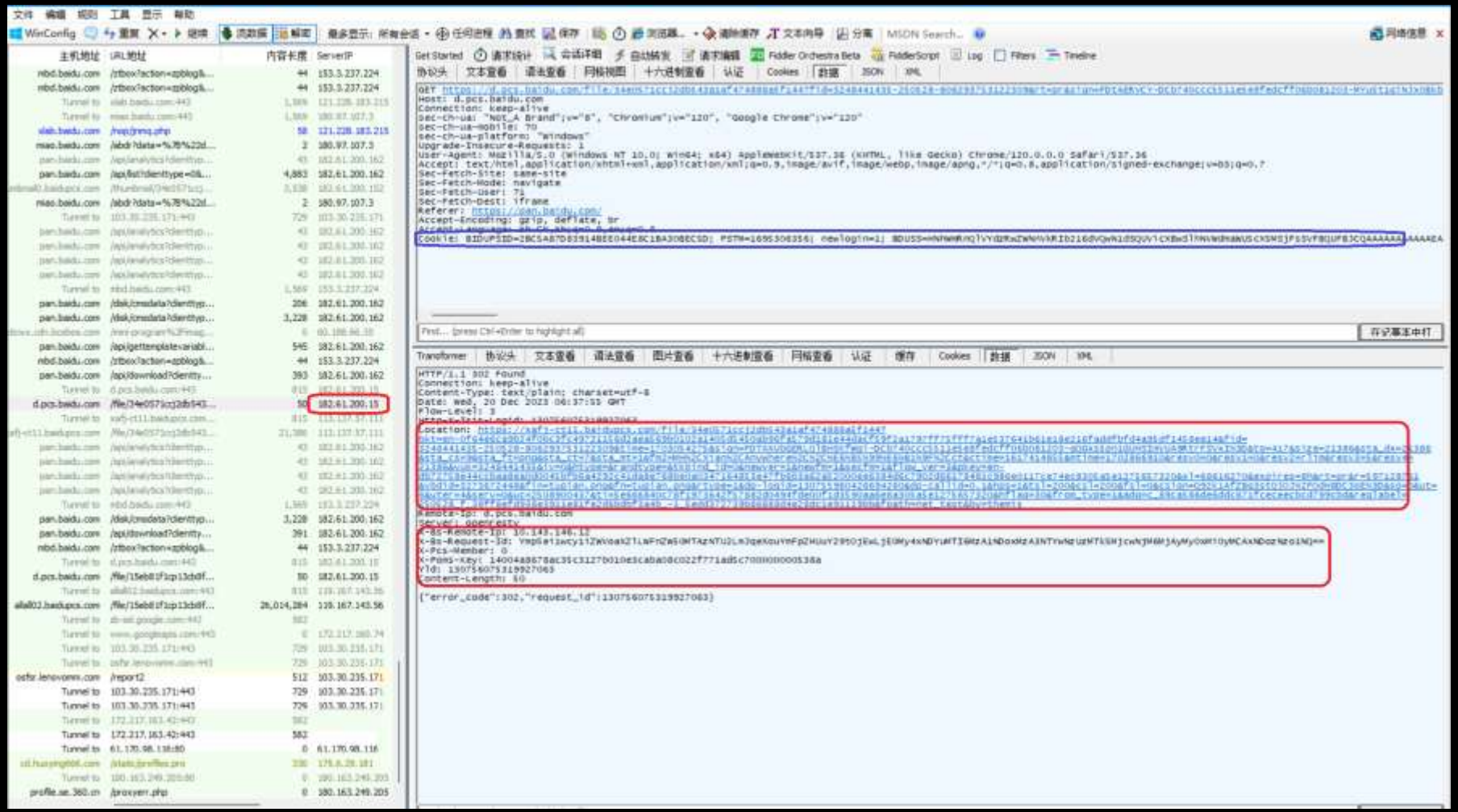
01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

182段IP证实刚刚看见的tcp握手确实是d.pcs.baidu.com
再看应答与请求
应答中的Location字段纯纯显眼包，仔细一看开头正是xafj下载服务器，这说明就是这个服务器给host提供了下载的地址。同时下面也有一些自定义的X字段，推测是认证信息，此处不予证明。
而请求报文看起来则是风平浪静，与一般网站相差无几。但仔细看它超出屏幕的cookie就知道事情不简单



NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

02 网页端 VS 客户端

一般下载网站 VS PanBD网页端：

上方的requestURL传递了文件信息，最瞩目的就是VIP字段了
下方的cookie则传递了用户的认证信息。从很多很长的ID中能感觉出PanBD的安全性还是不错的

研究到这，我们进一步明确了方向，即
网页端和客户端在登录、运行、下载文件时与百度的各类服务器的交互逻辑链有什么异同，与下载时携带的http首部有什么区别

NBP-NETWORK

02 网页端 VS 客户端

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 从清华镜像网站下载 4167029 字节 ubuntu-22.04.3-live-server-amd64.iso.zsync

PanBD网页端：

故接下去我们按照分析d.pcs.baidu.com的逻辑将网页端主要（出现频率高）的子域名服务器作了分析
因时长有限，我们直接给出结论
其中
请求服务器用于处理所有的动作请求并给出下一指令应答
推送服务器用于建立全双工通信以实时获取数据交互
略缩图服务器用于预览图片，子域名中的X代表数字
静态资源服务器负责存储框架类web前端文件
埋点服务器在特定动作埋伏，获取用户特征画像以优化运营策略
文件传输服务器分为专线和通用，每次连接的服务器为host发送请求时的相对最优解。专线服务器的前四位’aabb’代表地区和用途，’cu/cm/ct’代表三大运营商，XX代表number
通用服务器一般以allall或者 地区+all为域名，所在地服务器规模一般较大

具体的结构框图会在本节最后给出

‘根’服务器:pan.baidu.com/pcs.baidu.com/baidupcs.com

百度账号登录服务器:passport.baidu.com

账号数据云服务器:pcsdata.Baidu.com

请求服务器:d.pcs.baidu.com

推送服务器:webpush.pan.baidu.com //websocket 全双工通信

略缩图服务器:thumbnailX.baidupcs.com

静态资源服务器:bdstatic.com

埋点服务器 mbd.baidu.com Mobile Big Data / hm.baidu.com hao123 monitor

文件传输服务器

专线服务器 aabb-cu/ct/cmXX.baidupcs.com
通用服务器：allall/地点all01.baidupcs.com

NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

03 从百度客户端登录并下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 网页端 VS 客户端

PanBD客户端：

同理（省略一万字），我们整理出客户端从登录到下载所接触的服务器，并与网页端对比

右图中黑色为增添服务器，白色为保留服务器，灰色为不可见服务器

其中，host会规律性的向更新服务器发送用户状态，包括签名，传输沟道，VIP信息等
与P2P服务器的连接出现在登录初始化时，可能用于所谓“高速下载”？
插件服务器提供内核、基础库、升级程序等插件，封装了web端的一些功能服务器

我们会发现相比于网页端，客户端在下载服务器上并无太大区别，而功能服务器似乎更加集中密闭，从某一程度上确实提高了安全性。

找到了交互服务器的区别，接着再聚焦于下载时的字段有无明显区别

‘根’服务器
更新服务器:update.pan.baidu.com
P2P连接服务器:aa.t.bcsp2p.baidu.com
百度账号登录服务器
账号数据云服务器
请求服务器
推送服务器
略缩图服务器
静态资源服务器
埋点服务器
插件服务器:wppkg.baidupcs.com
文件传输服务器

NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

03 从百度客户端登录并下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

02 网页端 VS 客户端

PanBD 网页端 VS PanBD客户端：

下面我们对两者的来往报文与requestURL作对比
先看报文：

```
GET https://xafj-ct11.baidupcs.com/file/34e057icc12db543a1af474888a5f1447?bkt=en-0f64e6ca9b24f0bc3fc49721156d2aea569b0102a1405d5450ab96fa579d181e44dacf59f2a1797
Host: xafj-ct11.baidupcs.com
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Google Chrome";v="120"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Referer: https://pan.baidu.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
```

```
HTTP/1.1 200 OK
Date: Wed, 20 Dec 2023 06:37:55 GMT
Content-Type: image/png
Connection: keep-alive
Content-Disposition: attachment;filename="tupian.png"
Content-Length: 21386
Cache-Control: max-age=259200
ETag: 4a8678ac35c3127b010e3ca8a08c022f
x-bs-client-id: HTIyLjIyNy4yNTMuMTC3
x-bs-file-size: 21386
x-bs-request-id: MTAuODAuMTE2LjE1MDg0MTc0OjE2MDc1NTk2MDQyNjY5NDI2MDoyMDIzLTEyLTIwIDE0OjM3OjU1
x-bs-meta-crc32: 1998247367
Content-MD5: 4a8678ac35c3127b010e3ca8a08c022f
superfile: 0
Accept-Ranges: bytes
Last-Modified: Sat, 03 Apr 2021 01:53:29 GMT
Server: PCMS/CloudUI 1.0

PNG

IHDR  sRGB  gAMA
  pHYs  IDATx^
*** FIDDLER: RawDisplay truncated at 128 characters. Right-click to disable truncation. ***
```

```
GET http://xafj-ct10.baidupcs.com/file/34e057icc12db543a1af474888a5f1447?bkt=en-0f64e6ca9b24f0bc3fc49721156d2aea569b0102a1405d5450ab96fa579d181e44dacf59f2a1797f
Host: xafj-ct10.baidupcs.com
Connection: Keep-Alive
Content-Length: 0
Accept: */*
Accept-Language: zh-CN
Range: bytes=0-21385
User-Agent: netdisk;P2SP;3.0.0.236
Accept-Encoding: identity
```

```
HTTP/1.1 206 Partial Content
Date: Wed, 20 Dec 2023 11:52:11 GMT
Content-Type: image/png
Connection: keep-alive
Content-Disposition: attachment;filename="tupian.png"
Content-Length: 21386
Content-Range: bytes 0-21385/21386
Cache-Control: max-age=259200
ETag: 4a8678ac35c3127b010e3ca8a08c022f
x-bs-client-id: HTIyLjIyNy4yNTMuMTC3
x-bs-file-size: 21386
x-bs-request-id: MTAuMzkuMTAxLjM4OjMwMDQ0MTM1ODE3MTQ4MTUwMTA4OTk5OjIwMjMtMTItMDhjAgMTk6NTI6MTE=
x-bs-meta-crc32: 1998247367
Content-MD5: 4a8678ac35c3127b010e3ca8a08c022f
superfile: 0
Accept-Ranges: bytes
Last-Modified: Sat, 03 Apr 2021 01:53:29 GMT
Server: PCMS/CloudUI 1.0

PNG

IHDR  sRGB  gAMA
  pHYs  IDATx^
*** FIDDLER: RawDisplay truncated at 128 characters. Right-click to disable truncation. ***
```

上方左右两图分别是网页端的来往报文，下方为客户端
报文上二者无明显区别，除了前者的agent为浏览器，而后者netdisk
再试图分析requestURL首部

NBP-NETWORK

01 从网页端下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

03 从百度客户端登录并下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

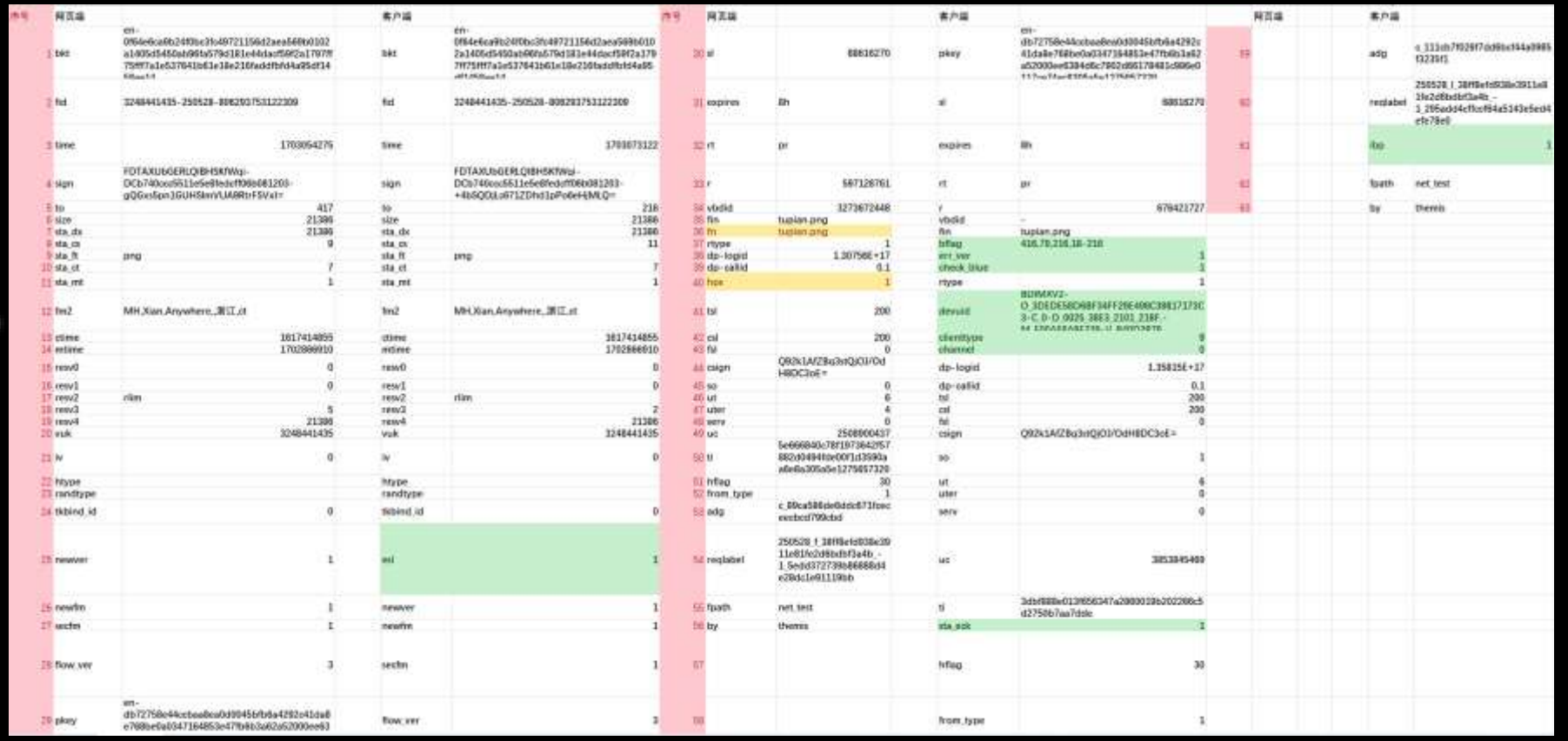
02 网页端 VS 客户端

PanBD 网页端 VS PanBD客户端：

字段对比如下，
有颜色的字段代表该端的特殊字段

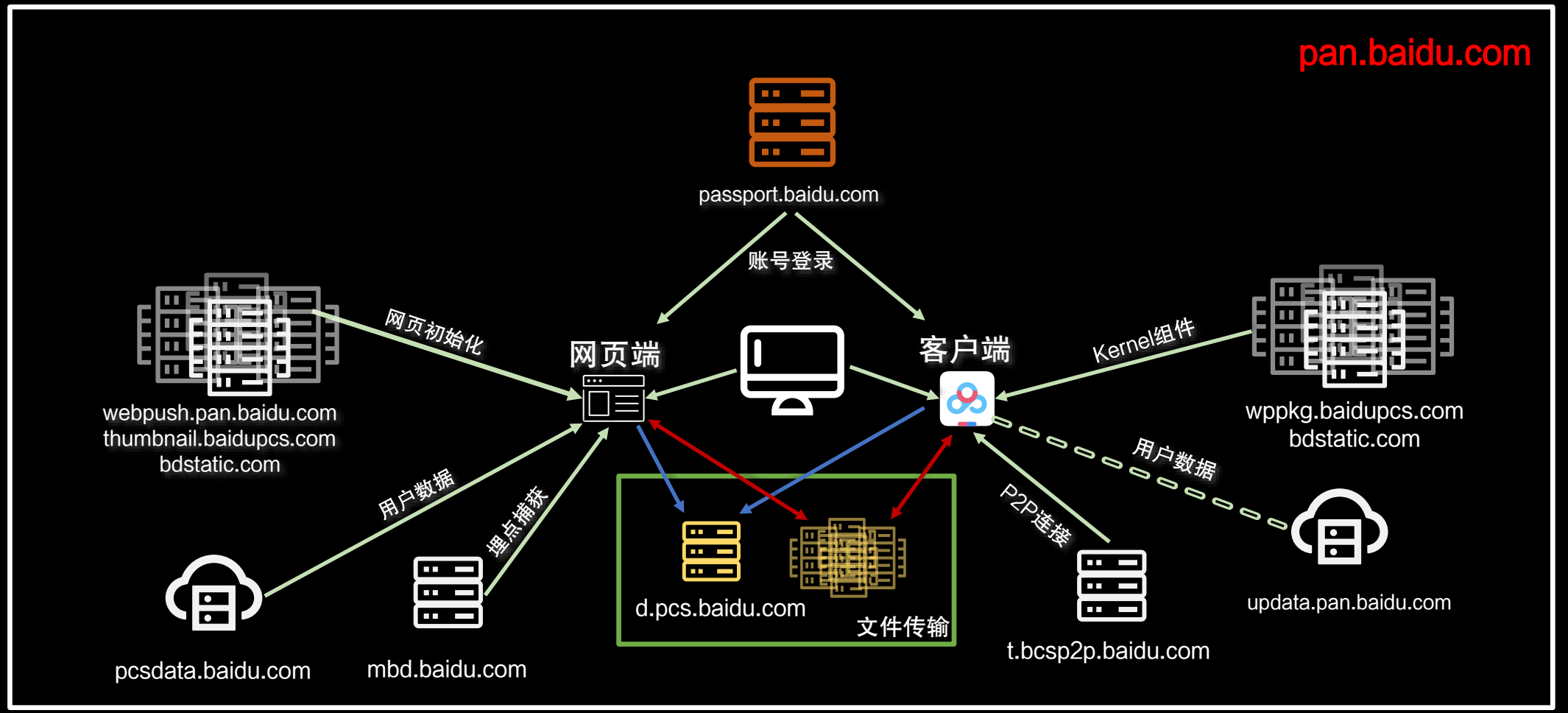
因能力有限，我们无法透过字段实锤结论，
只能做出猜测：

总体上二者的requestURL首部数量相差不大且都远远大于纯粹下载网址，再者rt字段都为pr，应为private，故推断下载时二者的逻辑本质上是没有差别的
网页端因通过browser这一媒介，故可能部分用于认证的字段可以略去



02 网页端 VS 客户端

总结框图



NBP-NETWORK

优雅提速直链获取网站: 94speed. com

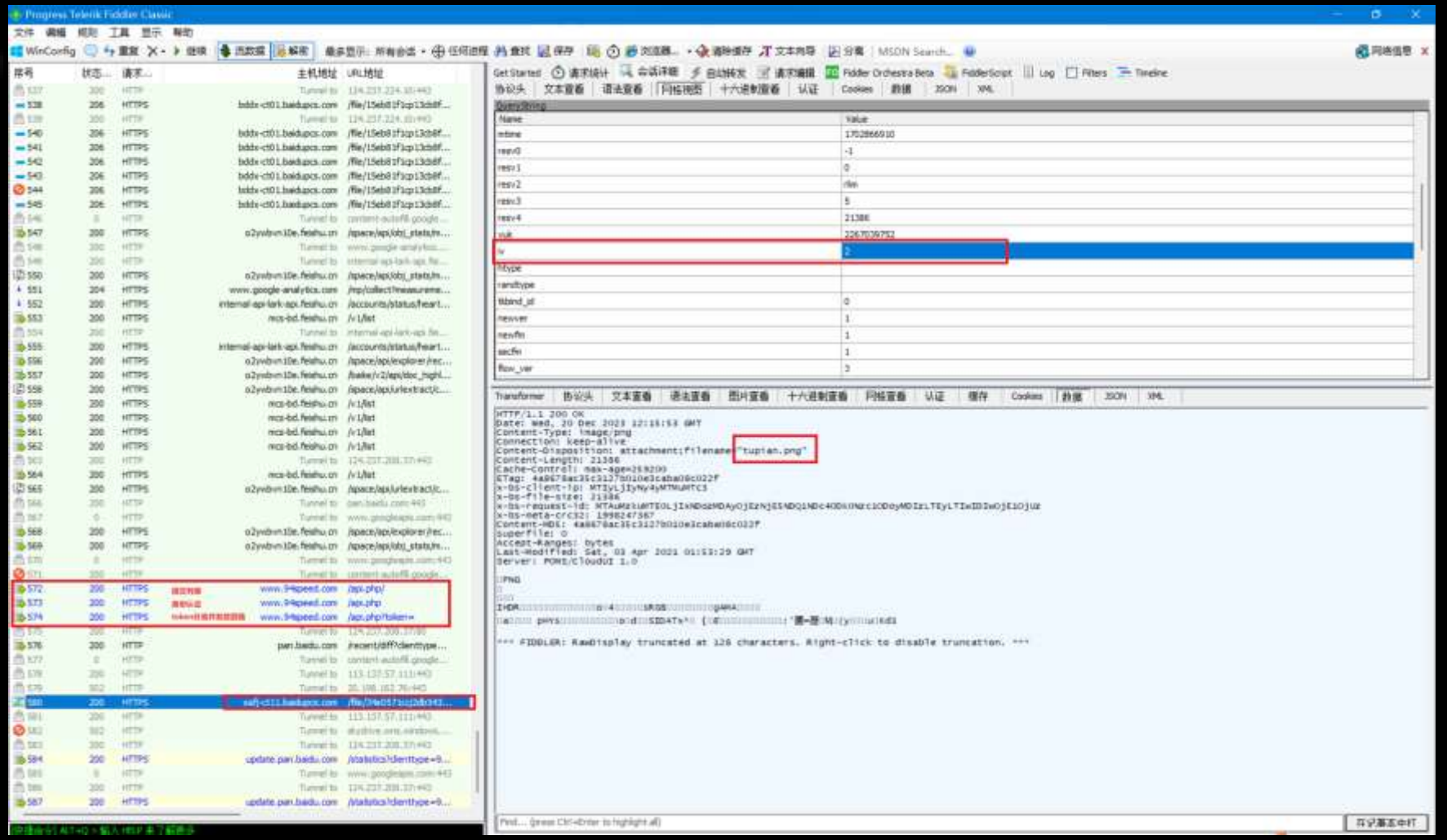
03 SVIP VS PoorMan

04 用第三方工具下载 21386 字节 tupian.png 和 26014284 字节 testppt.pptx

这里我们采用拿他人API接口获取SVIP直链的方法
可以看见原本的d.pcs.baidu.com变成了三个api的请求，分别为数据列表提交、身份认证和token验证

普通用户和SVIP用户所用的服务器是一样的
从requestURL的角度，除了iv字段从0变成2，还有一些随机数的不同以外几乎没有差别。

而后，我们开始怀疑tcp的窗口限制



NBP-NETWORK

03 SVIP VS PoorMan

但事实上，两者的实际窗口大小相差无几，怀疑不成立

接着我们想到了线程的控制，并对比了网页端、客户端和SVIP的区别，有了一些发现：
先看网页端，网页端用浏览器下载只有单线程，难怪恁慢

```
Window size value: 1040   SVIP
[Calculated window size: 33280]
[Window size scaling factor: 32]
Checksum: 0xd846 [unverified]
```

```
Window size value: 952
[Calculated window size: 30464]
[Window size scaling factor: 32]
Checksum: 0xb2aa [unverified]
```

序号	状态...	请求...	主机地址	URL地址	内容长度	ServerIP
290	200	HTTPS	pan.baidu.com	/api/download?clientty...	391	182.61.200.162
291	200	HTTPS	mbd.baidu.com	/ztbox?action=zpblog&...	44	153.3.237.224
292	200	HTTP	Tunnel to	d.pcs.baidu.com:443	815	182.61.200.15
293	302	HTTPS	d.pcs.baidu.com	/file/15eb81f1cp13cb8f...	50	182.61.200.15
294	200	HTTP	Tunnel to	allall02.baidupcs.com:443	815	119.167.143.56
295	200	HTTP	Tunnel to	110.242.69.174:80	0	110.242.69.174
296	200	HTTP	pan.baidu.com	/api/filediff?block_list=...	467	110.242.69.174
297	200	HTTPS	allall02.baidupcs.com	/file/15eb81f1cp13cb8f...	26,014,284	119.167.143.56

NBP-NETWORK

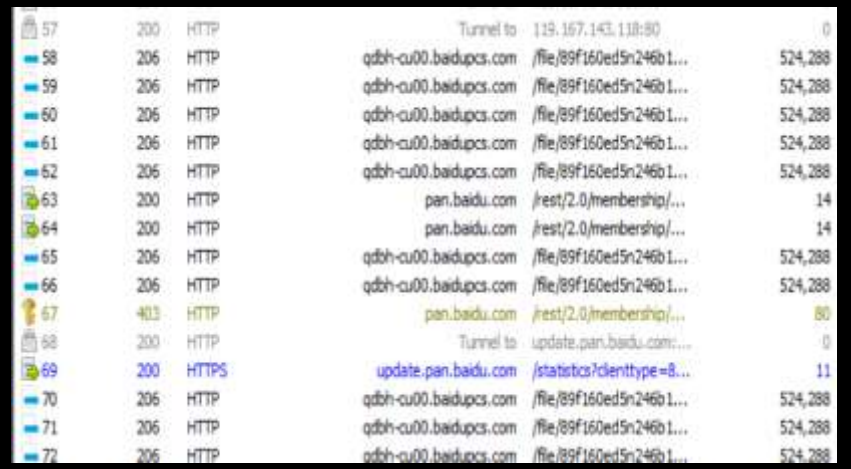
优雅提速直链获取网站: 94speed. com

03 SVIP VS PoorMan

左图为小文件客户端下载，速度在5M/s左右，已经是飞速了
可以看见网盘对线程数量并没有作过多的限制，而是对**每个线程的下载上限**作出约束，像左图都是2097152封顶
小文件封顶影响不显著，但一旦下载较大文件就要命了，上限掉到原来的1/4，意味着与服务器连接所花费的时间激增，自然速度就下来了

再看SVIP，在截图所在时间的网速下并没有触碰到上限，这样能让网络自己调整到最高效的下载节奏上
即不限速

综上，我们得出结论，网盘是通过对每一线程的上限作出限制而达成限速的目的~



NBP-NETWORK

04 买一送二

我们在研究的过程中了解了RESTful框架统一接口，并在抓包中发现了IPv6、IPv4的双栈传输，本欲详细说明，怎奈时间有限，只能贴出网址了T.T

优雅提速直链获取网站: 94speed. com

什么是RESTful API:
https: //zhuanlan. zhihu. com/p/334809573

什么是双栈结构:
https: //zhuanlan. zhihu. com/p/569641486

05 总结

基于wireshark和fiddler+proxifier两个平台，我们抓取了网页端和客户端、普通用户和SVIP用户的下载流程链，作出对比分析，最终得到了下载逻辑框图和限速原因，并给出了科学合法有效的提速方法，浅显地解决了自己的问题。
当然，我们仍未清晰的弄明一些请求字段的含义与作用，这是我们此次研究的遗憾。
最后，这里是NBP-宁波小学队，希望我们的调查能给你一些收获~