Yongwe Jean-Luc

Dataset exploration

Introduction :

Over the las two years the video traffic has exploded , in a sense that more people are watching more videos using multiple devices and wants all of them in high definition and high quality.

The traditional broadcasters are faced with a challenge to which their only response thus far is to extend their infrastrucutre and find themselves spending more money (in term of millions dollars) on the bandwith which economicly is not a sustainable solution.

At this rate the traditional scheme of delivery client to server has became prone to congestion and may soon be unscalable anymore.

Streamroot , in order to help their clients and partners to better handle that issue by using an well known technology but often overlooked which is the peer-to-peer delivery system that allows the broadcasters to offload their bandwith from the server by connecting the people watching the same stream at the same time (allowing the users to download the video from each other in the same area) .

Given a dataset of around 500K observations where each point is defined by :

Variable name	***Variable type***
streaming id (the video content identification)	*integer*
ISP name (Internet Service Provider name)	*string*
browser name	*strng*
connection state(connected or not to the backend)	*boolean*
p2p (peer-to-peer)	*float*
cdn (Client Delivery Network)	*float*

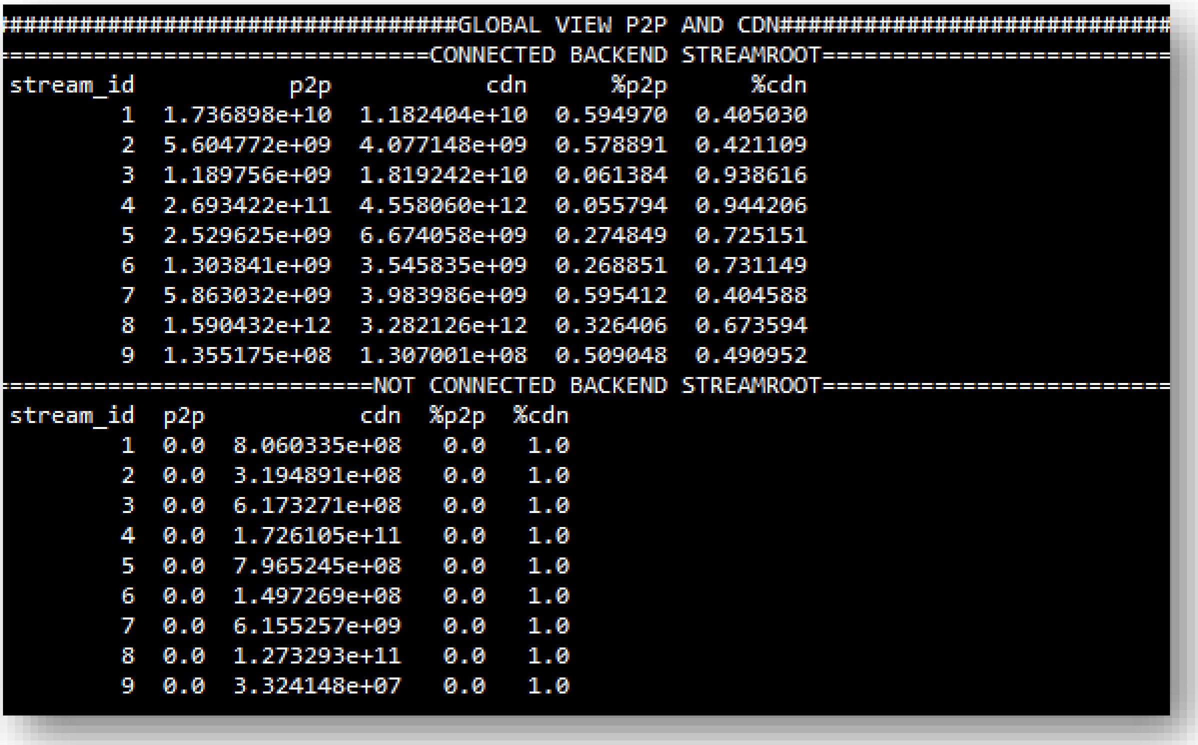
The all goal of this little project was to explore this dataset under different angles.

The choosen technology was Python because it is a very versatile and modular langage that allows the user to either simply manage and explore his datasets (preliminaries statistics and graphical vizualisation) and even pull off some more exotics insights by using advanced statistics.

The results are going to be reported by a quick description of the angle approach , followed by the graphical vizualisation

First angle :

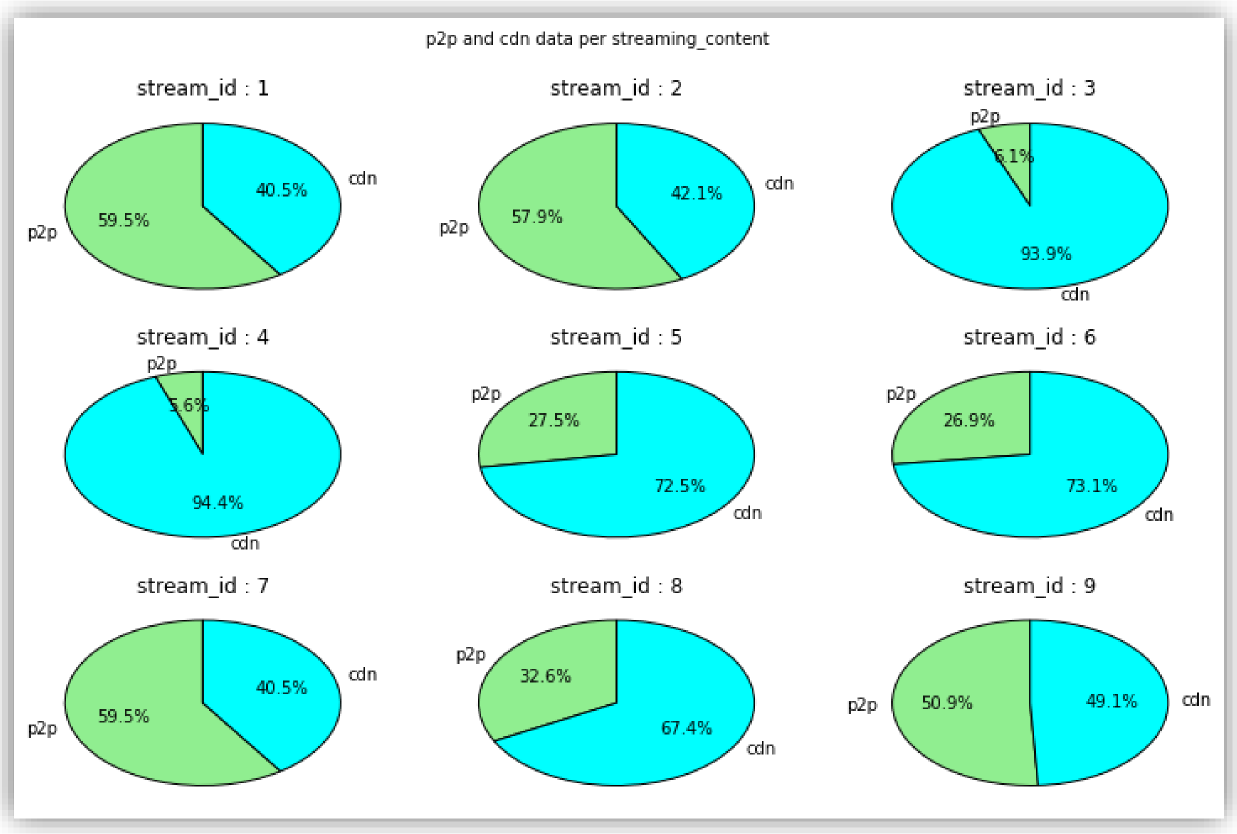
summarize (additioning) the amount of data exchange by peer-to-peer and cdn coming from the streaming content approach divided by the fact of being connected to the streamroot backend or not , regardless the ISP name or the browser.



```
################################GLOBAL VIEW P2P AND CDN##############################
==============================CONNECTED BACKEND STREAMROOT==========================
stream_id           p2p           cdn      %p2p      %cdn
        1  1.736898e+10  1.182404e+10  0.594970  0.405030
        2  5.604772e+09  4.077148e+09  0.578891  0.421109
        3  1.189756e+09  1.819242e+10  0.061384  0.938616
        4  2.693422e+11  4.558060e+12  0.055794  0.944206
        5  2.529625e+09  6.674058e+09  0.274849  0.725151
        6  1.303841e+09  3.545835e+09  0.268851  0.731149
        7  5.863032e+09  3.983986e+09  0.595412  0.404588
        8  1.590432e+12  3.282126e+12  0.326406  0.673594
        9  1.355175e+08  1.307001e+08  0.509048  0.490952
==========================NOT CONNECTED BACKEND STREAMROOT==========================
stream_id  p2p           cdn  %p2p  %cdn
        1  0.0  8.060335e+08   0.0   1.0
        2  0.0  3.194891e+08   0.0   1.0
        3  0.0  6.173271e+08   0.0   1.0
        4  0.0  1.726105e+11   0.0   1.0
        5  0.0  7.965245e+08   0.0   1.0
        6  0.0  1.497269e+08   0.0   1.0
        7  0.0  6.155257e+09   0.0   1.0
        8  0.0  1.273293e+11   0.0   1.0
        9  0.0  3.324148e+07   0.0   1.0
```

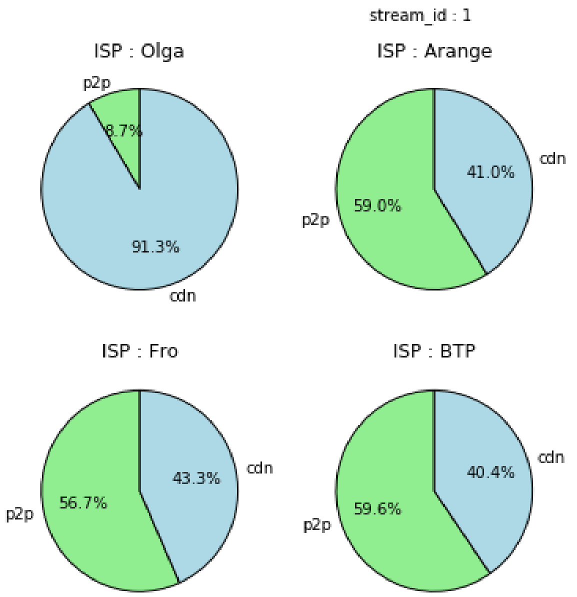
As we can see not being connected to the backend Streamroot left us with easily predictable information, from this moment on we will focus on the « connected » part.

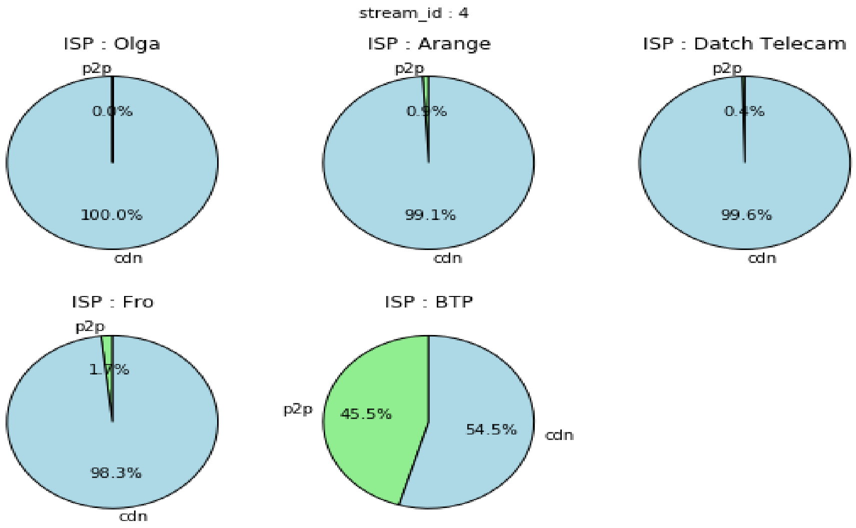
(cf. the next graph)

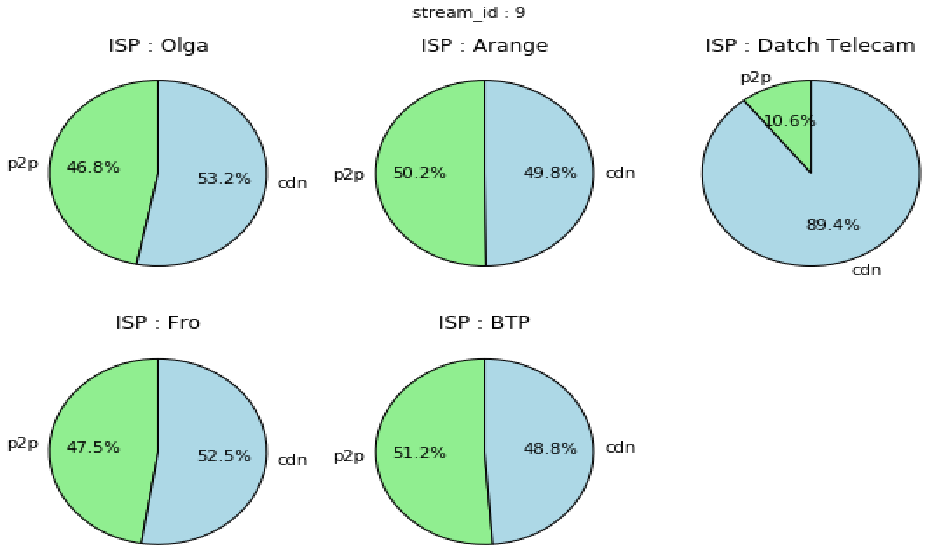


A first look at the graph seems to suggests that some streaming content are more likely to be downloaded on peer-to-peer than others but overall the cdn data are more called upon than peer-to-peer ones.

With a closer look :





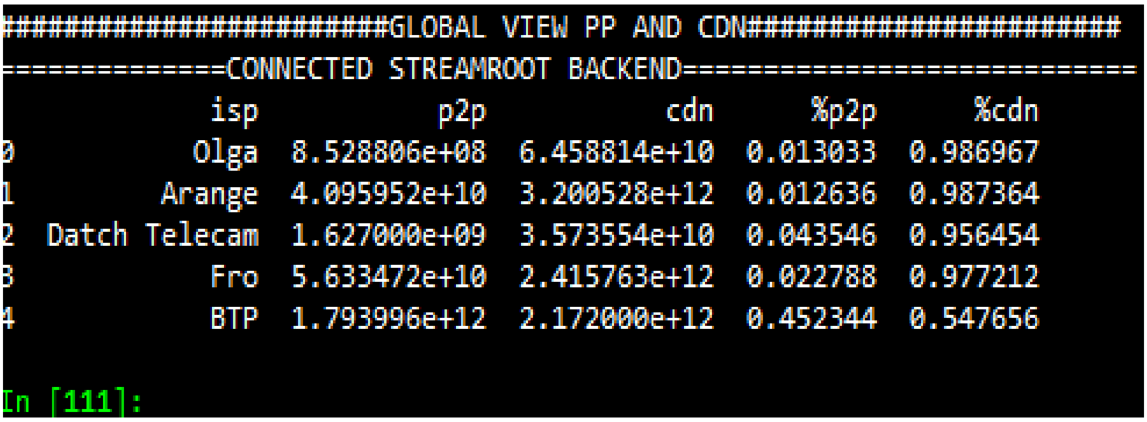


Those graphs confirm our firsts impressions and might give us a hint about the fact that some content may be shorter than others or the loading from some ISP is better off cdn than others (users might privilege peer-to-peer according the case)..

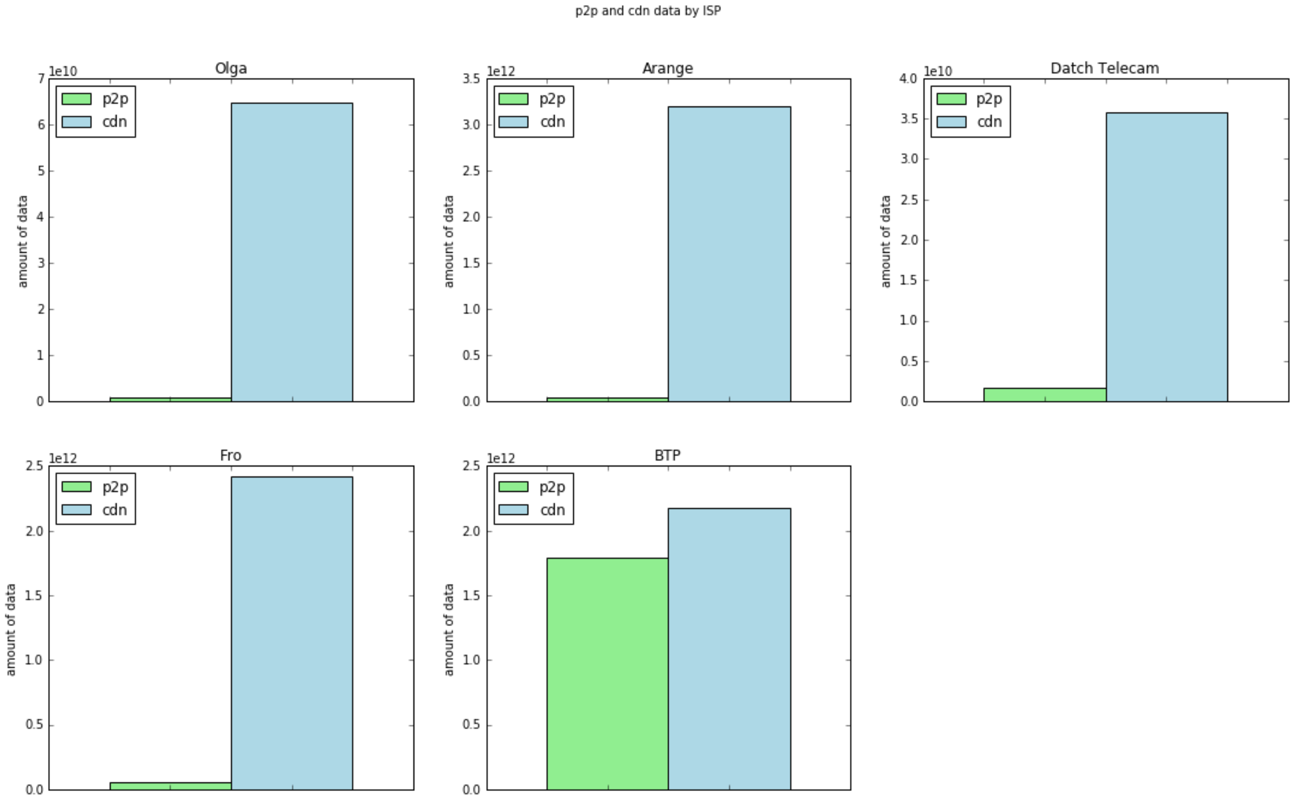
By paying attention you will realized that no matters the streaming content the ISP named 'BTP' is the one that has quite often the most peer-to-peer data exchanged (offloaded servers).

Second angle:

summarize (additioning) amount of data exchange by the users by ISP name regardless the browser name or the streaming content.

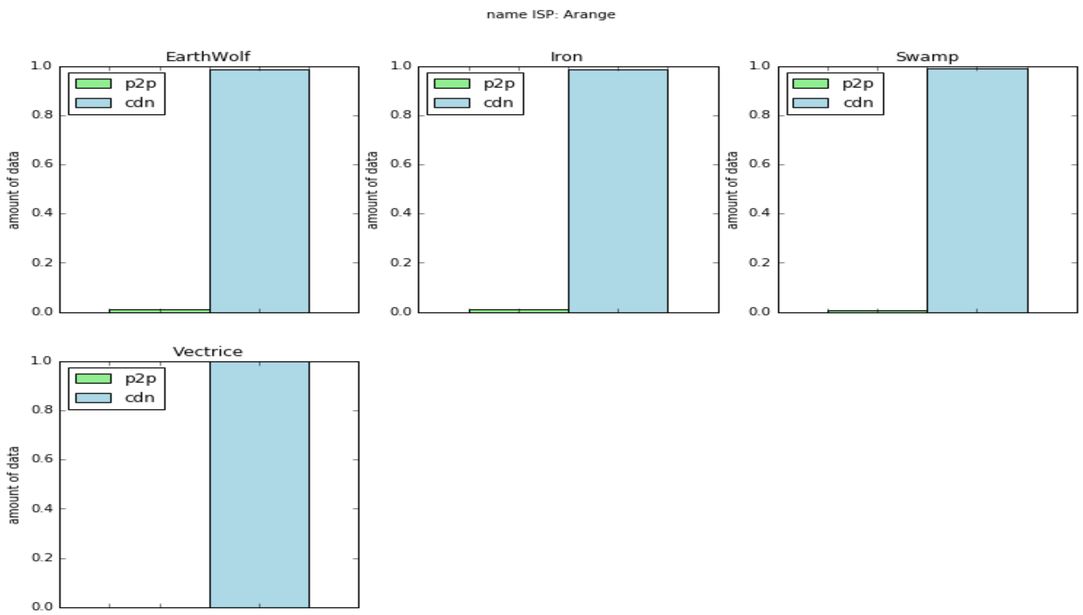


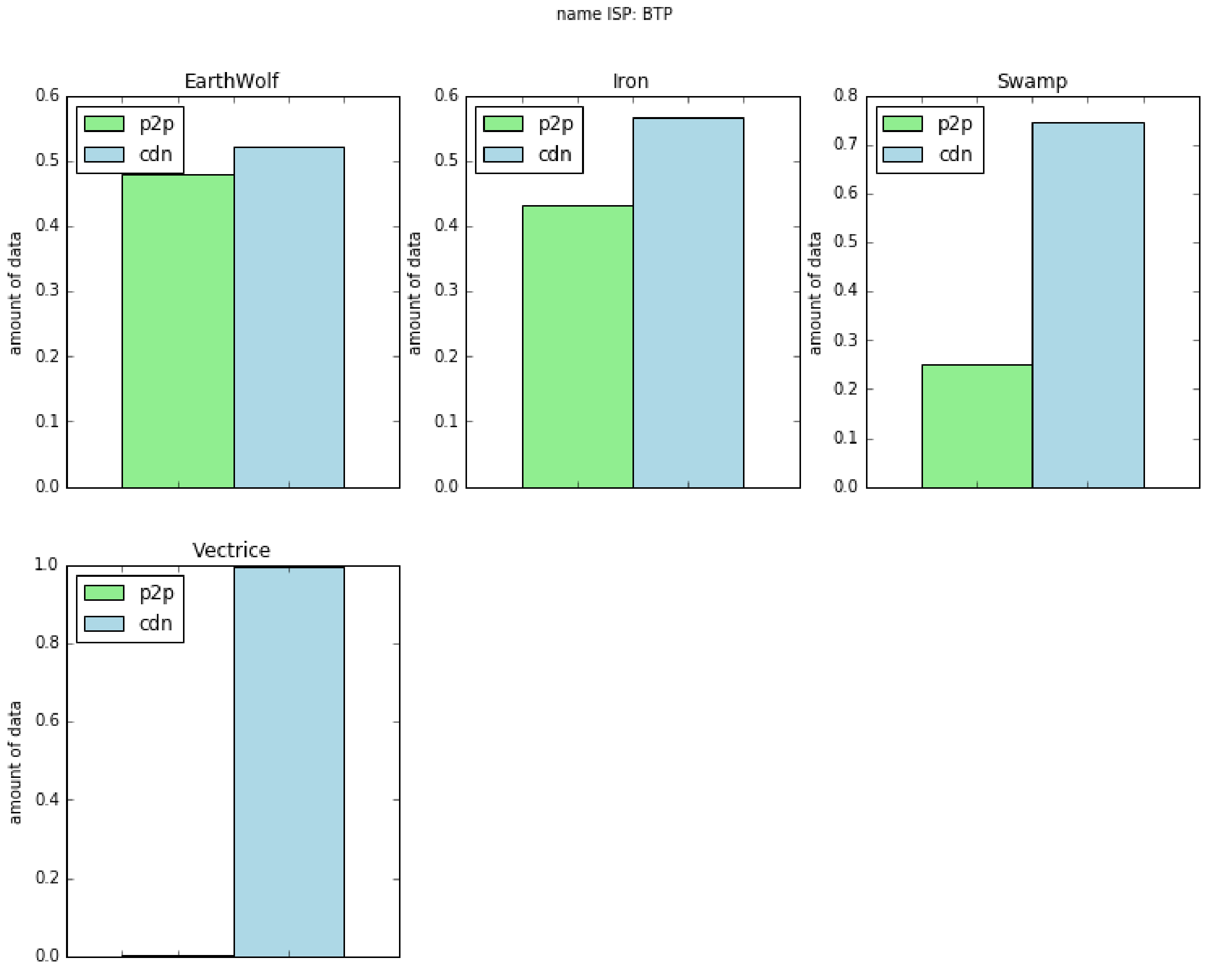
```
##########################GLOBAL VIEW PP AND CDN########################
=============CONNECTED STREAMROOT BACKEND===========================
            isp           p2p           cdn      %p2p      %cdn
0          Olga  8.528806e+08  6.458814e+10  0.013033  0.986967
1        Arange  4.095952e+10  3.200528e+12  0.012636  0.987364
2  Datch Telecam  1.627000e+09  3.573554e+10  0.043546  0.956454
3           Fro  5.633472e+10  2.415763e+12  0.022788  0.977212
4           BTP  1.793996e+12  2.172000e+12  0.452344  0.547656

In [111]:
```

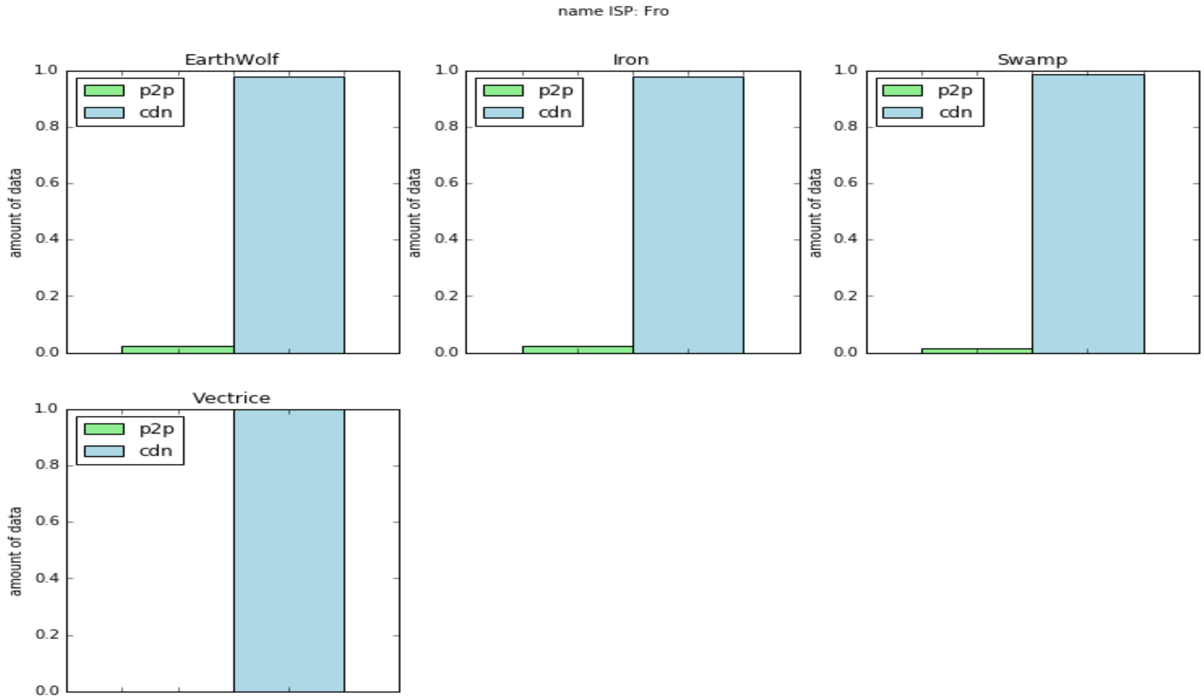


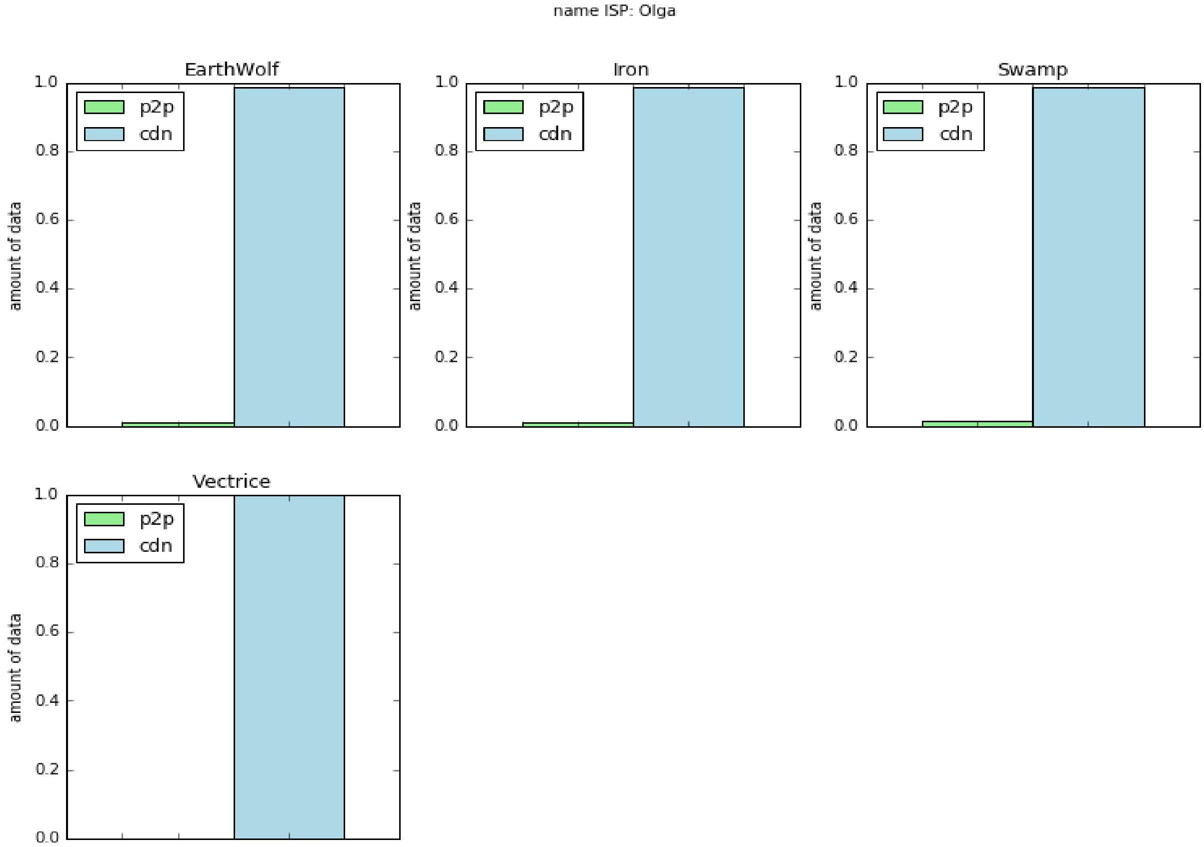
According to the graph peer-to-peer data seems to be more exchange with network of some ISP more than the others again here the most likely to see the peer-to-peer data exchanged is *'BTP'.*

Deeper look on the amount of data pee-to-peer vs CDN per ISP (% p2p and %CDN)









A deeper look shows us what we previously guess *'BTP'* is the one seeing the most peer-to-peer data exchange, and others ISP are more used through out differents browsers for heavy files or else , but divide their bandwith costs by replicating their server content.

Conclusion :

The ISP named ***'BTP'*** is the one that sees the most peer-to-peer data exchanged , that fact is confirm globally by the peer-to-peer and cdn data comparison per ISP and also by the peer-to-peer and cdn data comparison per ISP by streaming content , we can then conclude that the 'BTP' ISP is the one making the most economy about their bandwith expenses , or at least the most sustainable system of server content management in order to face this 'All out vids/pics age' that we are witnessing coming to us.

fews suggestions :

should be more marketing campaign to the potential client (ISP and browsers) and already knowns clients with more financials projection and an example of the benefits of those who are already using the strategy that you are sugggesting plus a suggestion of limiting the content of the server to the heavy files like movies,etc..any videos of than one hour.

The last suggestion could be optimizing 'like hell' the delivery detection algorithm that help a user to find in a peer user in order to download his next recommended video by the broadcaster he is using (Youtube with button 'automatic reading' on) and shorten the latency time beetwen the user data extraction and their availability in the database.