Subject: Re: HTTP client with bandwidth throttling
Posted by dolik.rce on Mon, 01 Nov 2010 20:38:16 GMT
View Forum Message <> Reply to Message

> darksmith wrote on Mon, 01 November 2010 12:42Hi! Sorry for my English ...
> How to implement file download with the bandwidth throttling?
> I did not find anything in the documentation
> Can anyone solve this problem?

Hello darksmith & welcome to the forum

This is very interesting question. It really cought my attention and I just couldn't resist to try  There is a quite simple and naive solution, using the progress callback (actually Gate2) in HttpClient. It has several flaws, but it might give you an idea if you need only something simple. Here is some example code:

```
#include <Core/Core.h>
#include <Web/Web.h>
using namespace Upp;

#define SPEED_LIMIT 1024 // 1 kB/s

inline double PreciseTime(){
 // returns time with milisecond resolution - there should be something more
 // clever than this, but I was too lazy to search...
 return GetSysTime().Get()+(GetTickCount()%1000)/1000.0;
}

bool Throttler(int written,int length){
 static double start=PreciseTime();
 if(length==0) start=PreciseTime(); // the download goes in chunks, we will count
                                    // average speed in each chunk separately (it's easier)
 double elapsed=max(PreciseTime()-start,1e-8); //we don't want to divide by zero!
 double speed=written/elapsed;
 while(speed>SPEED_LIMIT){
  Sleep(10);                   // this loop effectively pauses the download until
  elapsed=PreciseTime()-start; // the average speed doesn't fall back under
  speed=written/elapsed;       // given threshold
 }
 Cout()<<"Time: "<<elapsed<<", "          // This callback is originally meant
        "Downloaded: "<<written<<         // to be used for reporting the progress
        " => "<<(speed/1024)<<"kB/s.\n";  // so we do it here too...
 return false; // we always return false, true would cause aborting the download
}

CONSOLE_APP_MAIN{
 HttpClient c("http://www.ultimatepp.org/index.html_0.png");
 String result=c.ExecuteRedirect(5,3,callback(Throttler));
 SaveFile(GetHomeDirFile("upp.png"),result);
}
```

The biggest problem of this simple solution is that it limits  average speed (across each downloaded chunk) rather than actual download speed. There are probably also some other, hidden trouble  There are possibly better solutions and hopefully someone will tell us any better ideas  Anyway, I think the proper way to handle this would be inside the HttpClient code. It should be also more efficient that way...

Best regards,
Honza