

Re: Losing UDP packets with MFC Sockets

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2005-04/msg01045.html>

- *From:* Vicent Soler <VicentSoler@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 14 Apr 2005 01:35:07 -0700
-

Thank you very much for your response. I'll try to apply some of your suggestions.

Vicent

"um" wrote:

> "Vicent Soler" <VicentSoler@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote
>>
>> We are developing a tool which uses UDP packets to receive data from a UDP
>> Server. The problem we have found is that some UDP packets are being lost
>> when the PC's CPU is near 100% and we think that this problem is related to
>> the Window's input buffer.
>>
>> Any suggestion to solve this problem!! Is there any way to change the input
>> buffer size of the socket and store the received packet while the PC is
>> processing other data? Should we use threads to extract data from sockets?
>>
>> We are really worried about this problem because we can not lose so much
>> packets as we do.
>>
>> Using more than one port, could solve the problem?
>
> Depending on the requirements I would recommend the following
>
> 1) Increase each sockets internal buffer from the default 8 K to something larger
> (I used 32 K)
>
> 2) use a separate ring-buffer or FIFO queue of your own for the input
> and/or output. (should handle complete "records", ie. not single characters)
>
> 3) one or more threads doing the reception and putting the rcvd data into
> the input queue
>
> 4) one or more threads which processes the input queue (ie. processing the
> rcvd data)
>
> 5) add your data to be sent out into the output queue (should be possible from
> anywhere / any thread)

Re: Losing UDP packets with MFC Sockets

- >
- > 6) one or more threads which processes the output queue (ie. sending out to
- > socket(s))
- >
- > 7) use raw sockets, not the MFC wrappers, since they are in my experience not
- > reliable.
- >
- > You need thread-safe routines, ie. use critical sections or a fast locking
- > mechanism, because data is shared among multiple threads.
- >
- > I have used this successfully in one of my projects where realtime finance
- > data needs to be sent out to subscribing clients (ie. this is the server part).
- > Because the timing was ok, ie. no packet losses, I later even added a
- > data encryption, and it still is very fast and no packet loss is happening.
- >
- > The key ingredients are a well defined design and "protocol", fast algorithms,
- > data buffering, multiple threads, and resource sharing (ie. fast locking).
- > And of course a compiled language; not such esoteric things like interpreted
- > or "managed code" stuff for such hi-speed tasks; I had used C++ (VS6).
- >
- > Since you are doing "only" the client part it should suffice to perform
- > this just for the receiver side.
- >
- >
- >
- >
- .

• *References:*

- ◆ *Losing UDP packets with MFC Sockets*
 - ◇ *From:* Vicent Soler
- ◆ *Re: Losing UDP packets with MFC Sockets*
 - ◇ *From:* um
- Prev by Date: *Re: Losing UDP packets with MFC Sockets*
- Next by Date: *Re: Drawing trackbar thumbs*
- Previous by thread: *Re: Losing UDP packets with MFC Sockets*
- Next by thread: *Re: Losing UDP packets with MFC Sockets*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*