

## Re: Serial (rs232 etc.) to IP

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2005-10/msg00835.html>

---

- *From:* "Sonic" <spam@xxxxxxx>
  - *Date:* Thu, 27 Oct 2005 16:55:49 GMT
- 

I'm not sure if I am missing something, but this seems like a stupid conversation :)

TCP is a connection based protocol and as such it ensures that all data is received in the correct order and without errors.

There is no need to 'use' another protocol like Telnet if all you wish to do is transfer a serial stream of characters from one place to another over IP. That is what TCP/IP does.

Open a socket on one end and listen.

Start sending characters to the remote socket.

TCP uses a sliding window mechanism, which is effectively a buffer of data that has been sent that has not yet been acknowledged. The sender will continue to send until this sliding window is full. As data is acknowledged from the remote end, the window opens up some more and more data can be sent.

The size of the window dictates how much data you can send before you must receive an ACK from the other end.

Every time a SYN is sent, the other end must ACK, unless the remote end has implemented delayed acknowledgement. You can normally decline delayed acknowledgement when the socket is first negotiated.

Job done!

Hope this helps.

"Ignacio G.T." <igtorque.remove@xxxxxxxxxxxxxxxx> wrote in message [news:4360df29\\$0\\$26325\\$892e7fe2@xx](mailto:news:4360df29$0$26325$892e7fe2@xx)  
> Paul Keinanen wrote:  
>> On 26 Oct 2005 10:57:55 -0700, cs\_posting@xxxxxxxxxxx wrote:  
>>  
>>  
>>>>..and by doing this also ruin the serial line throughput when a half

## Re: Serial (rs232 etc.) to IP

>>>>duplex protocol is used with short message frames. Some converters  
>>>>wait 10–100 ms after the last serial character received, before the  
>>>>(TCP or UDP) IP–frame is sent. For instance at 115200 bit/s, this  
>>>>corresponds to 100–1000 character times.  
>>>  
>>>This is a valid concern, however there's a bit of a problem with single  
>>>character packets when many common embedded TCP devices try to talk to  
>>>common desktop operating systems.  
>>>  
>>>TCP requires that you hang onto all outgoing data until it has been  
>>>acknowledged, because if it's not acknowledged you will have to resend  
>>>it. Most embedded implementations handle this by sending a packet, and  
>>>then being unable to send another until the first has been  
>>>acknowledged. The Windows TCP stack often takes as much as 200 ms to  
>>>acknowledge a packet when only one is outstanding, so single character  
>>>packets can be slowed down to the rate of only five per second!  
>>  
>>  
>> If you are using a protocol that was initially written for serial line  
>> communication with normal CRC checks and timeout controls, why bother  
>> with TCP, just use simple UDP. If the UDP frame is lost, let the  
>> original serial line protocol timeout mechanism handle any missing  
>> data.  
>>  
>  
> This has at least one problem: these serial line protocols usually assume  
> that packets arrive in order. This makes sense for these protocols (I  
> cannot imagine a message jumping over the previous one and reaching its  
> destiny sooner than the other), but not for IP packets.

---

### • *Follow-Ups:*

- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: Grant Edwards
- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: Paul Keinanen
- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: cs\_posting

### • *References:*

- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: Frieder Ferlemann
- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: packer44
- ◆ **Re: Serial (rs232 etc.) to IP**  
    ◇ From: Paul Keinanen
- ◆ **Re: Serial (rs232 etc.) to IP**

Re: Serial (rs232 etc.) to IP

◇ *From:* Paul Keinanen

◆ ***Re: Serial (rs232 etc.) to IP***

◇ *From:* Ignacio G.T.

- Prev by Date: ***Re: device monitoring***
- Next by Date: ***Re: Parallel Port Wigglers PCI Cards and XP***
- Previous by thread: ***Re: Serial (rs232 etc.) to IP***
- Next by thread: ***Re: Serial (rs232 etc.) to IP***
- Index(es):
  - ◆ ***Date***
  - ◆ ***Thread***