



Understanding timeout settings in Digi One IAP

February 2018

90000649

Contents

1	Introduction	3
1.1	Overview.....	3
1.1.1	Multi-master queuing	3
2	Examining time out dynamic	3
2.1	Following a single poll/response cycle	3
2.2	Relating this to timeout settings.....	4
2.3	Recent changes in timeout settings and effect of baud rates.....	5
3	Setting the correct timeouts	6
3.1	Summary	6
3.2	Setting the Digi One IAP character timeout	6
3.3	Setting the Digi One IAP slave timeout.....	7
3.4	Setting the Digi One IAP message timeout.....	7
3.5	Setting the master application slave/message timeout.....	8
4	Troubleshooting timeout dynamic FAQ	9
4.1	Q: What are good starting defaults?	9
4.2	Q: Can I set all timeouts to one second?.....	9
4.3	Q: I saw timeouts in my application, so I increased the slave timeout. Now nothing works. What should I do?	10
4.4	Q: Why did our data rate drop after we moved our Modbus/RTU slaves to Digi One IAP and are using 100M Ethernet and Modbus/TCP?	11
4.5	Q: Why do normal reads/writes to my Modbus/RTU slave work, but vendor/extended functions timeout?	11

1 Introduction

Abstract

This document summarizes the various timeout settings available within the Digi One IAP and how they relate to external events and settings within the user's master and slave devices. Setting a meaningful combination of timeouts is critical for proper operation.

1.1 Overview

1.1.1 Multi-master queuing

The complex timing used by a multi-master product such as the Digi One IAP can be confusing. You don't need to understand system dynamics or queuing theory to understand multi-master queuing behavior. For example, four Modbus masters all submit a request at the same time to the Digi One IAP. Say it takes about 600 milliseconds for each request to be answered by the serial Modbus slave(s). Does this mean each master can expect an answer in 600 milliseconds? Is one second a reasonable timeout for each master? The best behavior you could expect is the first master in queue gets an answer in 600 milliseconds, the second master in queue gets an answer in 1200 milliseconds, the third master in queue gets an answer in 1800 milliseconds, and fourth master in queue gets an answer in 2400 milliseconds.

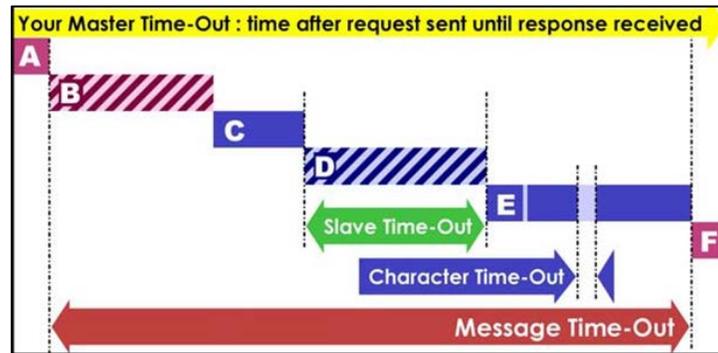
It is a mistake to set your OPC server timeout to one second because the slave answers in less than one second. Even if your Modbus masters with direct serial connection have been using a one-second timeout for years, you must adjust our master application configuration to accommodate a multi-master Digi One IAP. This document helps explain the things that affect timeout settings in your multi-master Digi One IAP system.

2 Examining time out dynamic

2.1 Following a single poll/response cycle

The following time diagram shows a single request-and-response through a multi-master bridge such as the Digi One IAP. The descriptions below the diagram imply a network-master bridging to a serial slave, but the same concepts apply to all combinations of masters and slaves.

The yellow line represents the complete time your master must wait for a response. The segments "A" to "F" represent distinct processes requiring time to complete. The four solid-color segments ("A", "C", "E", and "F") are generally fixed length, while the two striped segments ("B" and "D") are likely variable.



- A. **Request arrival:** This is the time it takes the message to reach and be recognized by the Digi device. This time can be short and predictable, or it can be long and variable. It is affected by network type, speed, and congestion. The time required can vary from 1–2 milliseconds for lightly loaded Ethernet to several seconds for serial, radio, or dialup connections.
- B. **Multi-master queue:** This is the time the request waits in queue for the slave device to “service” other requests already queued. At times it will be zero time. At other times it will be so long your master gives up on the request.
- C. **Serial request shifting:** This is the predictable time the request physically needs to shift across a serial link. At 9600 baud, this is about one millisecond per byte. So a Modbus/RTU read request of eight bytes requires about eight milliseconds to shift from the Digi One IAP to the serial slave.
- D. **Slave processing:** This is the variable time for the slave to recognize and process the request into a response. For some slaves, this time is fairly predictable and may be 10 or 50 milliseconds. Other slaves only process requests during predefined periods or “time slots,” so time D may vary from 10 to 250 milliseconds depending on when it arrives in the slave cycle. Still other slaves require variable amounts of time to answer different requests. Finally, if the slave is “off-line,” the request times out and takes the maximum amount of time.
- E. **Serial response shifting:** This is the predictable time the response physically needs to shift across a serial link. At 9600 baud, a Modbus/RTU read response of 125 registers requires about 255 milliseconds to shift from the serial slave back to the Digi One IAP.
- F. **Response return:** This is the time it takes the message to be sent from the Digi device and be recognized by the master. This time can be short and predictable, or it can be long and variable.

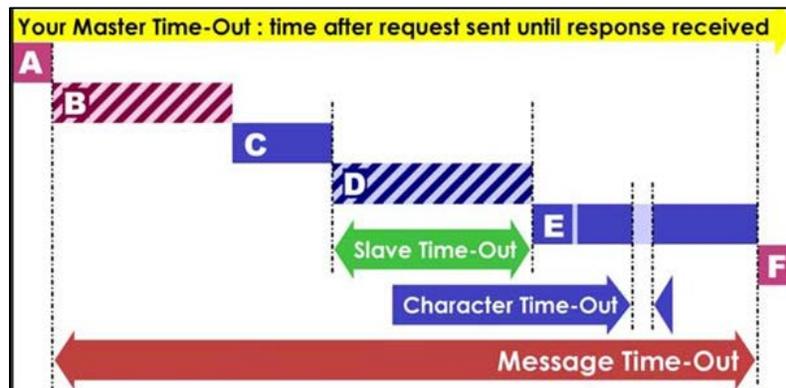
2.2 Relating this to timeout settings

This same diagram is related to the four-core timeout settings you need to manage, one in your external master and three in the Digi One IAP. These are:

- Master’s **Slave Timeout** (yellow arrow): This is the time your master application must be willing to wait for a response from the slave(s), which is really the Digi One IAP acting as proxy for the shared slave(s). This timeout is external to the Digi One IAP, but still must be set appropriately for proper operation.
- Digi’s **Message Timeout** (red arrow): This is the time it takes the Digi One IAP to attempt to obtain a response for the remote master. This must include multi-master queuing delays

plus a full serial slave cycle of poll/response. After this timeout, it is assumed the remote master has no more interest in any response.

- Digi's **Slave Timeout** (green arrow): This is the time it takes the Digi One IAP to wait for the first byte of a response after the last byte of request is sent. Notice that the serial transmit time is not included, so this timeout is baud-rate independent.



2.3 Recent changes in timeout settings and effect of baud rates

The Digi One IAP timeouts have changed from previous versions of IA firmware. Previously, the "message timeout" served as message and slave timeout, plus included processes "C," "D," and "E." It ignored multi-master queuing delays and was baud-rate dependent since "C" and "E" were included.

The old, single-timeout design caused problems in multi-master applications because the timeout would often be too short for multi-master contention but too long for real slave timeouts. Now with the two timeouts separated you can correctly set the slave timeout to reflect the true behavior of your slave device handling a single request. You can also correctly set the message timeout to reflect the queuing delays caused by multi-master access.

When slow baud rates were required, the old method of including the serial transmit times made setting the correct timeout difficult. For example, if a Modbus/RTU slave normally responded within 300 milliseconds of the end of a request, at 9600 baud (about one millisecond per byte) of about 600 milliseconds was set to accommodate the serial shifting out, slave response delay, and serial shifting in of a maximum sized message ($8 \times 1 + 300 + 255 \times 1 = 563$). At 1200 baud (about eight milliseconds per byte) one needed to set a timeout of about 2.5 seconds to accommodate the same maximum sized message ($8 \times 8 + 300 + 255 \times 8 = 2412$ milliseconds). The new design removed the baud-rate dependent processes from the timeout.

Thus, setting a slave timeout of one second literally means that the slave must send us the first byte of a response within one second of us sending the slave the last byte of the request. This no longer means that we can send the slave a full request and receive the full response within one second, and the baud rate is not set too slowly for this to be meaningful.

3 Setting the correct timeouts

3.1 Summary

Setting a valid combination of timeouts requires the correct relationship between all four key timeouts you need to manage. The best defaults to start are:

- Your master application's slave timeout: try three seconds first.
- Digi One IAP's message timeout: defaults to 2.5 seconds.
- Digi One IAP's slave timeout: defaults to 1 second.
- Digi One IAP's character timeout: defaults vary by protocol, but usually is 50 milliseconds.

These defaults safely allow two remote masters to share most serial slaves.

Note: In all cases, these relationships should remain true: Digi One IAP's character timeout is shorter than Digi One IAP's slave timeout, which is shorter than Digi One IAP's message timeout, which is shorter than your master application's message/slave timeout.

3.2 Setting the Digi One IAP character timeout

It is best to start with the smallest timeout and work toward larger, more encompassing timeouts; 50 milliseconds is a safe default for most protocols and direct EIA-232, EIA-422, or EIA-485 links. This character timeout setting is also used on TCP/IP protocols as the permitted time gap between fragmented pieces of a message.

For any protocol with a well-defined end-of-message (including AB/DF1 and Modbus/ASCII), setting a timeout that is too large has little effect while setting it too small causes excess false timeouts. It is best to err on the side of too large.

The one notable exception is Modbus/RTU. The official standard defines the end-of-message condition as a 3.5 character idle-gap with no new data. At 9600 baud, this gap is about four milliseconds. Unfortunately, a lot of products and modem equipment make this impossible. The Digi One IAP uses an intelligent estimation routine on all common Modbus functions that eliminates the need for this timeout. If you are using standard Modbus read and write commands, Modbus/RTU within the Digi One IAP acts as if it had a well-defined end-of-message pattern. Just as with DF1 and Modbus/ASCII, setting this timeout too large has little effect while setting it too small causes false timeouts. The Digi One IAP uses the character timeout to detect Modbus end-of-message only on unknown or vendor-defined commands.

- **When to increase:** If your serial line uses radio modems or other packet-network technology, you may need to increase this value. Try 100, 150, and 200 milliseconds for common radio modems if you are having trouble. For satellite systems, you may have to set this value in the 1-2 second range.
- **When to decrease:** You should only decrease the character timeout if you have a protocol using it to detect end-of-message and you have a direct serial link (no modems or other packet-network technology). For Modbus/RTU, only consider decreasing this value if you make heavy use of uncommon or vendor-defined commands.

3.3 Setting the Digi One IAP slave timeout

Next, look at the slave timeout. This is the time it takes the slave to start to respond after seeing the last byte of the request.

The slave timeout cannot be safely tweaked to maximize throughput within the Digi One IAP directly. There are too many variables involved. The only way to tweak the concept of slave timeout is with a direct serial connection between a single master and one or more slaves. So with your OPC server or configuration tool directly connected to your slave(s), experiment with the timeout setting until you break the link and receive a high error rate. Then increase the timeout 30%. Be aware that you must confirm that all types of commands work with this setting. Writes commonly take longer to complete than reads, especially writes related to an EEPROM update or device reprogramming. Vendor-specific commands can take even longer. So do not be surprised if the slave timeout you optimized for “reads” prevents reprogramming or using vendor-specific commands with your device.

- **When to increase:** The one-second default is very conservative and likely suitable for 99.99% of all slaves. Since this setting is baud-rate independent, you should not change this timeout when you change baud rates. Changing baud rate from 9600 to 1200 or 38400 won't change the time it takes the slave to start to respond. Increase this setting if your slave takes longer than one second to respond. You should also increase it if there is a serial link that passes through wide-area network technology such as a satellite connection with a high or variable latency. Many satellite systems require 1-2 seconds for an end-to-end message to propagate due to internal priority queuing.
- **When to decrease:** You should only decrease the slave timeout if you have a single master directly connected to the slave that can use a shorter timeout. For most small, dedicated slave devices, this value can be set to 300 or 500 milliseconds. If the slave devices have not responded within 50 milliseconds of the last byte of the request, they probably won't respond. Test all possible commands if you want to reduce this. Setting an incorrect value can cause negative consequences, such as when your production is in high gear or you are rushing to change a bad configuration on a failing device. This occurs when your slave devices are the busiest and thus most likely to respond more slowly.

3.4 Setting the Digi One IAP message timeout

This message timeout is the time it takes to queue a request before the remote master issuing the request has given up on seeing any response. This timeout takes into account multi-master queuing delays, so it must be larger than the Digi One IAP slave timeout, yet less than your master application slave timeout.

This timeout exists solely for house cleaning. Suppose three masters each issue 10 TCP requests per second, but the serial slave is only physically able to answer four per second. After 60 seconds, your TCP channel will potentially have buffered over 1,560 old requests waiting for the serial port. Most likely, your masters have given up on 1,530 of these requests. The Digi One IAP rapidly accepts and time-stamps all requests. In the above situation, the Digi One IAP

starts timing out excess requests and returning error responses to your master application without those requests ever being given access to the slave.

- **Important tip:** A timeout event from a Digi One IAP does not necessarily mean the slave device timed out. It can also mean your masters are creating too many requests per second for the limited bandwidth of the slave. The best practice is to set this to (500 milliseconds + (slave timeout X maximum expected masters)). The default value of 2500 milliseconds is appropriate for two masters sharing a slave with a one-second slave timeout. Expected masters should include your programming system even if it connects rarely. If you expect to be running two OPC servers full-time, you should consider a 3.5 second timeout (or $500 + (1,000 \times 3)$). If you expect six remote HMI panels to query your slave, this could be as long as 7.5 seconds ($500 + (1,000 \times 7)$). If you have more than two masters running, you should spend some time to determine the best slave timeout value.
- **When to increase:** The message timeout must be increased as more masters are added, as the baud rate of a serial line is made slower, or as each master starts asking for more requests per second. The Digi One IAP message timeout must be increased in synchronization when increasing the master application timeout. Setting the Digi One IAP message timeout to five seconds when your master application times out at one second is an inappropriate combination.
- **When to decrease:** The Digi One IAP message timeout is really a safety net that is seldom reached. If your master(s) are only sending requests that the slave can answer, timeout events are strictly related to actual slave timeouts. The Digi One IAP message timeout only activates when the master(s) start sending too many requests, or the slave(s) start slowing down their responses. A common example of the latter is when one or more slaves on an RS-485 multi-drop go off-line. For example, suppose six slaves can normally each answer one request per second and the master sends six requests per second. When one of the slaves goes offline, the Digi One IAP waits one second for the off-line slave. This drops the effective throughput to only one response per slave every two seconds. If the master is not smart enough to back off from sending six new requests per second, eventually the Digi One IAP message timeout kicks in and rejects excess requests. Unfortunately, none of the common IA protocols include a message to the master indicating that the system dynamics have changed and congestion is building. Therefore, your master will just receive errors simulating slave timeouts.

3.5 Setting the master application slave/message timeout

This timeout is the easiest value to set, but it can easily be set incorrectly. You must set the timeout in your master application to slightly longer than the Digi One IAP message timeout. Three seconds is a good default to start with, but do not forget to increase this if you increase the Digi One IAP message or slave timeout.

- **When to increase:** This is the first setting you should increase if you start to experience message timeouts. Do not increase the Digi One IAP slave timeout if your application starts to see message timeouts. This is almost certain to aggravate your problem and produce more application timeouts.

- **When to decrease:** Modbus/TCP and the Rockwell protocols all include explicit error responses for slave timeout. Reducing this timeout to speed up retries is rarely a good idea. The Digi One IAP is in charge of slave message timing and true slave timeouts. If the slave times out, the master application will (if appropriate for the protocol) receive a timely error response message and can attempt a retry.

4 Troubleshooting timeout dynamic FAQ

4.1 Q: What are good starting defaults?

Setting a valid combination of timeouts requires the correct relationship between all four key timeouts you need to manage.

The following are the best defaults to start with:

- Your master application's slave timeout: try three seconds first.
- Digi One IAP's message timeout: defaults to 2.5 seconds.
- Digi One IAP's slave timeout: defaults to one second.
- Digi One IAP's character timeout: 50 milliseconds.

These defaults allow two remote masters to share most serial slaves.

In all cases, these relationships should remain true. Digi One IAP's character timeout is *shorter than* Digi One IAP's slave timeout, which is *shorter than* Digi One IAP's message timeout, which is *shorter than* your master application's message/slave timeout.

4.2 Q: Can I set all timeouts to one second?

You cannot set all timeouts to the same value. System dynamics and queuing theory make this a bad combination. This also guarantees you will have problems if a second master (such as your programming tool) connects to the Digi One IAP.

For example, the following are bad methods, and are not advised:

- Your master application's slave timeout: one second.
- Digi One IAP's message timeout: one second.
- Digi One IAP's slave timeout: defaults to one second.

These are bad because they violate the laws of nature (queuing theory, for example). You need at least 100 milliseconds and some compensation for baud rate between the various levels of timeout.

If you are really enamored by the one-second value—and are certain you will never accidentally run a second master—you can consider using the settings below. They at least cause the timeouts to occur first at the slave side of the Digi One IAP.

The following settings are valid, but are poor choices and you should only use them under specific circumstances:

- Your master application's slave timeout: 1.0 second.
- Digi One IAP's message timeout: 0.9 second.
- Digi One IAP's slave timeout: defaults to 0.6 second.

These are poor settings because a second master accidentally started could cause a near 100% timeout rate in your first master.

4.3 Q: I saw timeouts in my application, so I increased the slave timeout. Now nothing works. What should I do?

Seeing timeouts at your master application when your slaves are healthy often means your application is expecting too many requests to be answered per second. Increasing the Digi One IAP's slave timeout decreases the number of responses possible per second if one of the slaves fails to respond. Thus your master application will see more timeouts, not less. The Digi One IAP's slave timeout must be set to a value appropriate for a single master directly connected and should never be changed on the fly to troubleshoot.

This is an example of an inappropriate combination. Since your master application times out first, it creates more requests faster, which builds a longer queue, creating congestion in the Digi One IAP. This ultimately results in the Digi One IAP growing queues by simulating slave timeouts to reduce them.

The following settings can cause these problems to occur, and are not recommended:

- Your master application's Slave Timeout: one second
- Digi One IAP's Message Timeout: 2.5 seconds
- Digi One IAP's Slave Timeout: defaults to 3.5 seconds

These are bad because the relationship between the various levels of timeout is inverted. The Digi One IAP's slave timeout must be the shortest. The master applications timeout must be the longest.

If you see message timeouts in your master, the first thing you must do is confirm that all of the masters combined are not creating more requests than the shared slave can answer within any time period.

Some computer applications by default assume that Ethernet slaves have a high bandwidth and low response delay. Changing your driver from Modbus/RTU to Modbus/TCP may radically alter the defaults settings assumed by the master application. For example, some Modbus/TCP drivers default to a 100 milliseconds timeout. It is nearly impossible for a serial slave at 9600 baud to process a Modbus/RTU request/response this fast. It's more difficult if you add the system dynamic of Ethernet and the Digi One IAP running as a multi-master queuing system.

4.4 Q: Why did our data rate drop after we moved our Modbus/RTU slaves to Digi One IAP and are using 100M Ethernet and Modbus/TCP?

These questions occur and people do invest in Modbus Bridges in an attempt to reduce data latency. Yes, you are using 100,000,000 baud Ethernet for a portion of your path, but that 9600-baud serial line is still your bottleneck and limiting factor.

With a single master running a cyclic poll, you should not see much change in performance. However, large batch processes, such as uploading a large PLC program, take 5 to 10% longer via a Modbus Ethernet bridge than by a direct serial cable. This is due to the system dynamic and various queuing delays involved in going from application to Windows TCP stack, to Digi IAP TCP stack, to multi-master Q, to serial line.

With multiple masters running cyclic polls, you will see large reductions in performance over a single master polling by a direct serial cable. When two masters share one or more slave(s), each has only one-half the throughput. Three masters see only one-third the throughput. If your main goal is faster data updates, converting the serial slave to TCP/IP is not the answer. You need to buy native Ethernet slaves to see major improvements in performance.

There are situations where reducing the length of serial lines can enable higher baud rates. For example, replacing a 1200-meter RS-485 link forced to run at 4800 baud with a remote Ethernet-to-serial converter may allow that same RS-485 link with only two meters of cable to run at 38,400 baud. This may allow up to eight times better data-update rates. Of course, how you get Ethernet from here to there when it is 1200 meters away is an obvious question. The answers to that question are improving all the time.

4.5 Q: Why do normal reads/writes to my Modbus/RTU slave work, but vendor/extended functions timeout?

The Digi One IAP allows all commands from 0x01 to 0x7F to pass. So it is able to handle vendor functions such as 0x40. However, the way end-of-message is detected varies based on function.

For common Modbus and Modicon function codes, the Digi One IAP does not use the character timeout or the 3.5 character timeout to determine message length. It uses protocol knowledge to estimate the message length. If the CRC is zero at the expected length, we end the message with a zero delay.

For all unknown or vendor functions, the Digi One IAP has no choice but to wait for the character timeout to expire as a gap timeout. This defaults to 20 milliseconds for Modbus/RTU. This means that these commands take 20 milliseconds longer for a response to be registered. Many vendor functions relate to physical reset, start or stop commands, so they may take longer for a response.