ImonitorG Quick Guide [9-1-2023]

Installing the “imonitorg” raspberry pi 3B/3B+ on your network will give you wide-ranging and realtime monitoring of your local network and Internet performance, via a webpage available on the raspberry pi 365/24/7 [This is the standalone version of the “imonitor”]

The imonitorg will "come up running" and it is only necessary to set the time zone via the "raspi-config" and the email parameters via the rpi web page, if you desire a daily status email. It uses the normal pi user to run most of the scripts, and root for some. "Pihole" application is also included - point your DNS server to this pi and monitor all DNS activity.

1. Who should use an imonitorg

Residential or small business Internet customers

Single NAT'd IP appearance on the Internet

Cable, ADSL, satellite, cellular, or fiber WAN interfaces with a LAN side ethernet or wifi access. CGNAT access is also covered by setting a variable to designate access as such.

Anybody with a single NAT'd IP appearance on the Internet who wants to monitor and gauge their Internet performance over long terms.

2. Access the raspberry pi

Plug the raspberry pi into an ethernet port on your Internet router, or a switch connected to it. The raspberry pi expects an IP address, gateway and DNS, just like any device that you add to your home network. You will have to find the address of the raspberry pi - usually by going to the router home page and looking for it. Once you have the IP address, just browse to the raspberry pi - this will be htttp only, not https, so you may have to make exceptions in the browser to allow the http. There is really no need for https inside your network, so the exception is perfectly safe. Once you have the web page, you can setup various parameters, tho it will work out of the box. The most important thing you can setup is the time zone and the email configuration. Email will use gmail as a relay [you can send to the gmail address as well]. Of course you need a gmail account to do this, and you need to set up 2FA on gmail.

Alternately, if you don't want to mess with ethernet you can connect a KVM to the raspberry pi [HDMI monitor, USB kbd/mouse] and login to the desktop. This is a standard distribution raspberry pi. The login is pi and the password is "raspberry." Once on the desktop, you can setup a wifi connection - using the icon in the upper right. You can choose to use wifi only like this, or have the wifi in addition to the ethernet. You can of course change the pi credentials if you so desire.

You will definitely want to set the TZ if you are not on the US East Time. Access the pi via cmd line in the normal manner [or via KVM] and run the "raspi-config" to set the TZ and locale if you are not US.

3. Setup raspberry pi interfaces - Ethernet and/or wifi

The "imonitorg" is a generic raspberry pi 3B or 3B+ - the only thing different is all the scripts and cron entries to effect the network monitoring - these will run automatically on boot, once it is able to see the Internet and synchronize ntp clock service. You will login to the raspberry pi using the pi login, either via a KVM desktop, or via ssh to the user pi from another computer. The pi is useable as a normal raspberry pi in all respects. Just do not modify the scripts or cron files. The root file "rc.local" is also specific to this application!

The imonitorg expects to operate behind your Internet router on your private network in the same manner as one of your network devices. It will expect an IP address/gateway/DNS via DHCP. It should not be connected directly to the Internet, as there is no firewall running. *Connect your ethernet and/or wifi interfaces as follows.*

Ethernet only. Plug the imonitorg into your router or a switch on the same network as your router [or a router behind your Internet router]. Wifi networks are detected and scanned for SSID/Signal Level/Secured, but imonitorg does not log into wifi SSID of course if it does not have wifi credentials. Ping and TCP stats are performed via Ethernet only. No Wifi SSID/passwd needs to be provided.

Wifi only. Wifi SSID and passwd are needed. You can provision these via cmd line using "wpa_supplicant" knowledge, or you can do it via connecting a KVM to the imonitorg and doing it much like a Windows or MAC PC - there is a convenient wifi icon in the desktop upper right. Ping and TCP stats will be performed over the wifi since the Ethernet is not connected.

Ethernet and wifi. Wifi SSID and passwd are needed. This provides the most complete testing. Ping and TCP stats are performed over the Ethernet to the Internet. Ping stats are performed over the wifi to your gateway only to gauge wifi network performance. [this mode may disappear in the future due to routers inconsistencies with wifi provisioning simultaneous with ethernet to a single host]

4. Configure imonitorg options from the webpage:

- Control email options

On this standalone, generic rpi, it is possible to receive a daily status email by using the relay capability of gmail for user accounts. Your gmail account must be set for 2FA, and you must obtain a 16 byte key to use as the authorization string,

Set gmail relay [where you can receive the daily email status] on the rpi web page:

gmail relay enable state - YES or NO

gmail account - gmail account "username@gmail.com"

gmail authorization - gmail app password -16 byte key obtained on gmail site

ultimate email - ultimate destination email address[es]

See this link for e.g. instructions: https://johnloop.com/imonitor/gmailRelayInstructions.txt

- Set Management Options

There are no management options on the imonitorg. It is completely standalone.

A user may login as "pi" and perform operations, but he must be careful. All monitoring scripts run as user pi or as user root. You will see various files/scripts referring to "management" and "revssh," but these are all disabled on the generic imonitorg. They were used for development.

- Configure “near” ICMP Ping target, and "LOCK" or "AUTO"

Near ICMP ping target is settable, but it reverts to automatic mode next 2AM. A "lock" option on the near ICMP ping target is available to override this. The ping target is fixed -"LOCKED"- until manually returned to "AUTO. "

The timeout value is fixed at 1000 msec.   AUTO mode configures the ping target as the closest pingable hop starting at hop [5,4,3,2 -configurable with 5 being default] and counting down. It uses **IP address only** to forego use of DNS.   Thus it uses a close target to gauge local [near] ISP performance, independent of DNS and Internet-at-large. ICMP echo request packets [UDP only] are used. The timeout is set to 1 sec default. This near ICMP ping is the basic test of connectivity, but is supplemented by others, discussed following.

It is possible to set the hop start used for the ping determination to further allow customization. This can be 5,4,3, or 2, or even 1-the local router.   Sometimes the auto algorithm settles on a ping target[s] that can become very unreliable. We need to find a reliable near ICMP ping target. This can be adjusted once the ping behavior is noted after a few days.

- No configuration is necessary for “deep” ICMP ping

There is no “deep” ICMP target for the imonitorg. For the imonitor mode, these targets were the collection of the near ICMP targets of the other customers, so they represented "deep" targets for an individual user.

It is possible to place a file "CustPingSites.newtxt" in the /home/pi/tests/customerpings/ directory, and the pi will do the deep ICMP ping for one day starting at 2AM to 1AM. This file should contain an IP address on each line. If you want the ping to be repeated, make a cron file to reload the file each day before 12AM.   These addresses are expected to be ipv4 [ipv6 addresses in this mode has not been tested].

- Configure TCP target, and "LOCK" or "AUTO"

A TCP "connect" is performed to a "foundation" target to initially determine online/offline status each minute, and then a TCP "ping" is performed to a round robin of "100 top web sites" to determine response times each minute. The Internet FQDN DNS is used to invoke the DNS query mechanism. "www.google.com" or "www.msftncsi.com" is suggested for the "foundation" TCP target. A "LOCK" or "AUTO" may be applied to the variable TCP ping. The "AUTO" option uses the top 100 Web sites as variable TCP ping targets [as of spring 2020]. The "LOCK" option uses the foundation target instead of the fixed target as the TCP ping target. The "foundation" TCP connect target remains fixed even with "AUTO" option. It uses a TCP target to better gauge larger Internet performance. It uses TCP half open exchange, sending a SYN, and if a SYN/ACK is received, returns a FIN/ACK. The TCP Ping is only attempted while online [Internet accessible]. Care should be exercised in interpreting the TCP ping data. The top 100 web sites are largely implemented on local CDN sites and tend to be a few hops away in urban locations.

As of 5-2023 there is a variable to set to enable/disable TCP ping plotting. Often, because the TCP pings are to the CDN nearest, there is little utility in the plots. The "anomaly" TCP timeout is always plotted which gives more info. Just assume that TCP pings are "normal" unless you see the TCP timeout plotted in the anomaly section. You can, as root, "echo [YES|NO] > /root/DEFtcpplot.txt" to enable/disable this capability. It is only updated on a reboot [monthly unless forced].

As of spring 2022, the top100 web site listing uses "naked" domains. Occasionally, it is necessary to convert the "naked" domain, such as "google.com" to a valid FQDN such as "www.google.com." This occurs when a domain subscribes to a CDN, which requires a valid domain for CNAME. You can actually edit the /root/Top100WebSites.txt file to correct this problem.

The /root/Top100WebSites.txt file is for US. You can modify this file, or reconstruct it in the same format for different countries [you can eliminate the third column]. It must contain exactly 100 lines.

- Configure curl of URL

Twice an hour, a “curl” of a configurable URL is performed, and the time recorded. A “curl” is just a cmd line way of “browsing” - pulling down a web page. This response time is plotted on the daily and realtime plot with the three other measurements. This scan represents the “highest level [from a protocol level]” scan performed. Only the index page is downloaded, not the images and links referenced in the index page. In addition, the dns response time of a curl to www.google.com is recorded and plotted. This is normally done to your router, as long as it caches or proxies DNS requests. The raspberry pi checks this, and reverts to 8.8.8.8 if the router does not perform caching/proxying.

ISPs will sometimes force the router to use their own DNS, such as comcast often does with 75.75.75.75. It is not possible for the router to override this -usually. The DNS server used for the DNS query is noted in the daily plot. Most often it is your own router.

- Enable *Internal* scan of your network [default OFF, ON]

This is a scan conducted by the rpi on your internal private network. It can be especially important if you have wifi in a wifi maze of networks. It scans for opens ports/services on your network hosts using linux "nmap." It discovers, **lists all hosts** and changes day/day on your internal network via IP address. There is a listing, including a historical listing of HOSTS/ports in “Data snapshots,” complete report in “counts.” These stats are available on the rpi webpage.

The linux utility "nmap" is used to perform a scan of your local network. This can be enabled/disabled on the rpi web page. This is detailed on the rpi web page. A list of IP addresses, change of IP address, services, and change of services day over day is performed.

There is a "findhost" script which spans a week to detect "new [never been on your network before]," "gone [it was previously on your network, but disappeared -this lasts for two scans]", and "persistent [this host has been on your network 10 or the last 12 scans]." This algorithm normally runs twice per day, at 6PM and 1AM [busy and not busy]. It is useful to detect transient hosts and permanent hosts.

- Set Temperature Probe Up and Down email alerts

If temperature probe installed. 10 emails sent at 15 minute intervals if alert remains. Not expected to be enabled for imonitorg. The scripts are present and configurable if you add a temperature probe.

- Configure pihole

The rpi comes with pihole already installed. Confer https://docs.pi-hole.net/ for a description of pihole. The Rpi web page has a link for the pihole administration page, and also links for the pihole log which is collected by the rpi.

Pihole expects a static IP address, and it must be set using either the pihole admin page, or going to /etc/pihole/setupVars.conf and setting it there [and restarting pihole]. You should also go into the admin page ->settings->conditional forwarding and set your network and router address. It is not necessary to use the pihole as a DHCP server.

If you wish to see DNS queries from the browser [chrome/FFOX etc], you will have to turn off DNS over HTTPS. The "DOH" is defaulty enabled as of 2021 in most browsers. Otherwise, record the log as without browser -which would show you the non-browser activity, and then turn off DOH to see additional browser activity.

There is a separate tab at the top of the rpi base page for pihole, and for the extracted logs.

- Addional configuration options

Route target for hop delineation. This is a far end location such as 222.1.1.1 [Japan], or 90.1.1.1 [France] to control direction of the packets. This may not have an effect if the same 5 routers are encountered in your path for any destination. Other customers are but a few hops away from the backbone where it will make a difference.

5. Email advantages/summary

There is a daily status email available by using your gmail account as a relay [see above]. It includes the daily plot.

6. Some advantages of using the imonitorg webpage

The webpage is implemented on your local network only, and is completely safe. it is http only.

It gives all stats reported in the email [comparing it to the “imonitor”], and is the way to configure your imonitorg. Here is a link to a pdf of the current webpage: https://johnloop.com/imonitor/RaspBerryPiWWW.pdf

Perhaps the neatest thing is that you can request a “real time” plot – it starts at 2AM and ends when you request the plot. It is linked on a separate page called by this script. You must wait a minute for it. You can also select a "ping blast" plot, where 60 pings are delivered to the ICMP ping target, and then plotted.

As of 9-1-2021, you can specify the start of the "real time" plot so that you can narrow in on problems. On the plot, there is a "hyperping" indication, which triggers a more intense ping of 6/sec when the 1/min near ICMP ping times out. If these "hyperpings" timeout, they will designated by "ticks" on the plot ascending vertically above the minute they were launched. This gives a much greater feel about the network availability around the 1/min ping failure.

There is no upload to a webserver in the Internet cloud to collect info [many similar services use a cloud account and require you to have an account/login to see the info for your own local network]. There is no use of a cloud server for the generic imonitorg!

The webpage is accessible even if the Internet is down, as long as your local network is OK.

As of August 2022, there is a webpage which provides near-realttime plotting of the performance. There is a link at the top of the main rpi webpage. Clicking this link will open a new tab on your browser which will be updated every 1 minute with the current plot.   You can still do a custom plot from the main rpi webpage, which is linked from this page [click it and refresh to preserve the plot, otherwise it will be overwritten with the next auto update].   You can always "save" a picture of the currently displayed plot by clicking the link at the bottom of the page -the saved plot will open on a new tab on your browser.

The webpage is not loaded with widgets and gadgets, with background loading of dozens [hundreds?] of websites and cookies. It is a very simple implementation showing stats and enabling configuration. There is only one figure included, and no scripting other than for alerts, and to allow you to set imonitorg parameters.

The webpage will show dated ping times, plus giving a link to the plots. Plus:

1 year archive of all plots and speedtests available, services and port changes on your network, plus the daily emails.

1 year archive [plus plot] of Internet offline dates and times.

The "pi-hole" app is installed, and is linked from the main page. Info is here https://pi-hole.net

7. Advantages of mgmt server imonitorg.com

There is no direct contact with this server, tho it serves as a knowledge base and reference for the project.

8. Some advantages of using the imonitorg on your network.

The ping plots are particularly useful for quick comparisons of your network longer term performance. You can quickly scroll thru days of performance in the archive window on the webpage. The same goes for many other plots and stats, including temperature monitoring, if you have a probe. A sample ping plot and temperature plot are included.

Archiving of all data, emails and plots, boot times, IP address records, for at least for 1 month.

Daily scan of your internal network for vulnerabilities [if scan enabled]

Daily scan of wifi maze, listing all visible networks and characteristics, every 30 min

The ethernet [or default ifce] ping tests will probe into Internet. The wifi ping tests probe your local network [as well as Internet if you are wifi only]

Listing of your local network hosts and svcs, plus notification of changes [detected at 7PM].

DNS tests - DNS change alerts.

Browser tab showing near-realtime performance plot. The near-realtime plot includes ping performance for all the various pings employed. An "anomalous" special plotting points out the "errors" encountered during the plotting. The plot shows ipv6 connectivity from minute to minute, the use of the router as the ping target [which compromises perf testing], and ping target changes which may occur on an hourly basis.

Ipv6 connectivity is indicated. The existence of an ipv6 default route is first checked, and even if the pi is unable to do an ipv6 ping of www.google.com, ipv6 is declaed OK. A "roofline" is drawn on the plot if ipv6 fails, meaning there is no ipv6 default route.

The use of the router as the ping target is also noted with a "roofline" on the plot, since this tends to limit the characterization and performance metric for the internet access. Sometimes the near ICMP ping target is AUTO selected as the router, and this is indicated thus. You can also LOCK the near ICMP target to your router and this is flagged by the "roofline."

9. Definition variables [settable via sudo to root]

These variables are located in /root and are set by sudo su – from the pi login. These can be changed by the imonitorg customer as indicated below, either via directly setting them via an ssh session, or via the rpi web page. They are not normally accessed except in special circumstances. Most configurable parameters are accessible on the rpi web page.

1. DEFpi.txt -this is a null file for imonitorg.

2. DEFpingtarget.txt -this defines the near ICMP ping target, auto determined each nite unless it is locked via DEFpinglock.txt. CommToServer.vx picks it up from localDEFpingtarget.txt every hour. Settable on the rpi web page. A "roofline" on the plot is drawn if this target is your router.

3. DEFconnecttarget.txt - this defines the "foundation" TCP connection target, default "www.google.com"; "www.msftncsi.com" is another good one - used by Microsoft to determine network connectivity.

4. DEFcgnat.txt YES for CGNAT ISP, NO for regular ISP [Internet IP is non distinctive]. This must be specified, otherwise Internet IP may wobble. CGNAT networks may often look like mobile phone networks!

5. DEFmgmtserver.txt – this is not used by the imonitorg.

6. DEFinterval.txt - not used

7. DEFmgmt.txt - this is not used by imonitorg.

8. DEFdnspull.txt – this defines where to get my IP address from. Currently "myip.opendns.com" This is hardcoded in the script. "o-o.myaddr.l.google.com" is also used.   **Should NOT be changed**

9. DEFcustemail.txt -this is not used by the imonitorg.

10. DEFmgmtemail.txt -this is not used by the imonitorg.

11. DEFtimeoutinterval.txt - This defines the ping timeout interval, in msec, from 1000 to 15000. Currently "1000" is default for ping and is not changeable.

12. DEFrealrouter.txt -this file exists only if the Rpi is behind an internal router, and is the internal IP address of the ***Internet*** router. Can test for existence of this file to test if behind internal router. **The customer must set this variable.**

13. DEFixc.txt -traceroute destination   222.1.1.1 points to Japan, 90.1.1.1 points to France.   It is really intended as a route eventual destination for "traceroute" even tho only the first hops are used.

14. DEFtemp1sensor.txt -this file exists whether or not there is an actual temp sensor, and whether or not the kernel module is installed.   There are potentially multiple temperature sensors, but this variable indicates #1. Set it to "ON" to enable temp limit checks, "OFF" to disable them [even if there is no temp sensor or kernel module]. The customer must set this manually, and install the kernel modules to make this work [besides installing the thermometer!]

15. DEFtemp1downlimit.txt -the down trigger temp for sensor 1 -set to "1"   degC

16. DEFtemp1uplimit.txt -the up trigger temp for sensor 1 -set to "50" degC

The DEFtemp variables are not expected to be used for imonitorg unless you know how to add the appropriate drivers.

17. DEFemailforIPchange.txt – this is not used for the imonitorg.

18. DEFsa.txt – this is not used by the imonitorg. It is set to ON, “standalone.” "Standalone ON" is the default setting for the imonitorg rpis. Standalone "OFF" was used by development pis to allow remote management.

19. DEFconfig.txt 1,2,3,4,5 for config value - only used on diagram right now. **This must be set by the customer. It is default "5" which should suffice for most.**

20. DEFgmailENABLE.txt - enable email relay via your gmail account. Default NO

21. DEFgmailUSER.txt - gmail user account to authenticate to for relay

22. DEFgmailAUTH.txt - gmail app password used with DEFgmailUSER.txt

23. DEFgmailULTIMATE.txt - ultimate email destination via gmail relay

The DEFgmail*.txt variables can be set from the web page.

24. DEFunsolicitedSYN.txt Default is "ON" - Set to OFF if you expect the pi to receive unsolicited SYNs from off network -not normally the case unless you have port forwarding to the pi. If your LAN network is larger than /24 [255.255.255.0] you may have to set this. Imonitorg only succeeds on /24 networks right now.

25. DEFpinglock.txt AUTO or LOCK; "LOCK" locks value, "AUTO" lets script determine it automatically each hour for the next hour.   This can be set on the rpi web page.

26. DEFhop.txt 5,4,3,2,1   5 is the current hop used for near ICMP. Settable by the DEFhop.txt variable below. If the hop goes to "1" - the router, an anomaly is plotted.

27. DEFscan.txt OFF or ON --internal nmap scan of customer network; default OFF. This can be set on the rpi web page. Performed once each day. Should be ON to discover your network ports/services.

28. DEFdailyemail.txt YES or NO or OFF – this is not used by imonitorg. It is set to OFF.

29. DEFtcppinglock.txt LOCK or AUTO. LOCK means use the value entered in the form -the "foundation" TCP ping target. AUTO means use the top 100 web site list for the variable TCP ping targets [as of spring 2020]. The value entered in the form is always used for the "foundation" TCP connect "test."   If you choose AUTO, it is only for the TCP "foundation" ping test. Note that choosing “AUTO for the top 100 web, these will normally be sites on the nearest Content Distribution network [CDN]. If the LOCK is set to AUTO, the "DEFpingtarget.txt" is set to the router [gateway] to cover the fallback near ICMP ping target case.

30. DEFcreateRTplot.txt – set by entering the start time, and pushing the “PLOTIT” button on the rpi web page to create current plot. Cleared by script that does the plotting. This variable is set via the rpi web page and checked every minute by CommToServer.vX. The plot will show the plot from the designated time to the time it is invoked.

31 DEFcreateMINplot.txt - set by pushing the ping blast "PLOTIT" button on the rpi web page to create ping blast plot. 60 pings in a minute to ICMP ping target.

32. DEFcustomerping.txt – this is set to NO for the imonitorg. It is a global enable not currently used. The customer can invoke customerpings as referred to above. This is a global setting for customerpings.

33. DEFversion.txt - set at image version creation. Simple date. Each execution of "/root/clean_piG_files_generic" is recorded here to signify "reset to factory default:."

34. DEFbootcurl.txt - each boot, the rpi attempts to curl bootcurl.txt from imonitorg.com/bootcurl.txt, which will list the latest version. User can then compare this with 35. DEFversion.txt. This curl occurs in CommToServer.vX and may occur before network exists, so not reliable unless a boot occurs while network exists. Not currently implemented as of 6-2022.

36. DEFpiversion.txt 3B, 3B+ not used at present

37. DEFimonitorversion.txt date, such as 07/22/2020

38. DEFcurltarget.txt -enter a URL for a favorite web site. This is used for the twice-hourly curl test.

39. DEFhop.txt -enter an ICMP auto ping hop start. 5,4,3, or 2, or 1. Default is 5.

40. DEFtcpplot.txt enter YES or NO to enable tcp ping plotting. Default is YES for generic imonitorg.

config1 eth plus wifi, both dhcp

config2 eth only, dhcp

config3 eth0 static/non network, wifi dhcp; packet capture on eth -not currently implemented

config4 wifi only, dhcp

config5 wifi static/non network, eth dhcp; packet capture on wifi -not currently implemented

10. Special Considerations

On a boot/reboot, the system clock may be wildly off until the Internet is acquired and ntp can sync the clock. The rpi and the modem/router should use a UPC for reliable stats. The ping target is recalculated each hour, so restarting the rpi at any other time will use the ping target of the previous hour. If the rpi wants to use the wifi, the SSID/key must be entered, either via connecting a KVM, or by modifiying /etc/wpa_supplicant/wpa_supplicant.conf

There is a special script "clean_piG_files_generic" in /root which you can use to clean the pi for initiation. Beware, it wipes all history.

The raspberry will operate with any IP address it is given via DHCP, **but the scripts may need an overnite to update all the links when the local IP changes. An entire day may be needed in some instances. To determine hosts, a week will be needed to work the algorithm.**

Remember the goal of imonitorg is longer term performance monitoring, not "instantaneous" performance monitoring.

If the rpi changes IP address, the pihole must be reconfigured vua /etc/pihole/setupVars.conf

11. For Information: https://imonitorg.com

John D. Loop   pccitizen@gmail.com or jdloop@johnloop.com

Below is a sample combined plot. Individual ICMP and TCP ping plots are also available via the web page.   It shows the values collected during the day, and the pi going offline at 6:00PM till 1AM the following day. The scripts and presentation are the sole property of John Loop.

12. Converting to a "managed" rpi

It is possible to convert the standalone rpi into a managed rpi via generating a key pair and sending me the public key. In this manner, I will be able to access the rpi for updates and monitor as I develop it further. You will also receive a few extra items, such as an internet scan. Contact me if you are interested. Not generally available.

13. Using pihole

Pihole is installed on all rpis. It is accessed via a link on the main rpi web page, or via http://RPI_IPaddress/admin/index.php. Pihole provides a way to monitor all DNS activity on your local network. You can configure individual clients on your network to use the rpi as your DNS server, and all DNS requests will be routed via the pihole. Pihole was originally intended to be used to filter DNS requests to block ads, but it is implemented here merely to monitor the DNS. You may still configure with the adlist servers to do the blocking. The docs are https://docs.pihole.net.

You can monitor the pihole log on the rpi web page, as it is updated at 5 minute intervals, and also see Yesterday's pihole log. Pihole was originally meant to be used with filter lists to filter out ad/malicious web addresses. As implemented on the rpi, there is no filtering, but it can be added to the pihole in the normal way. These DNS queries are archived as well.

14. Plots

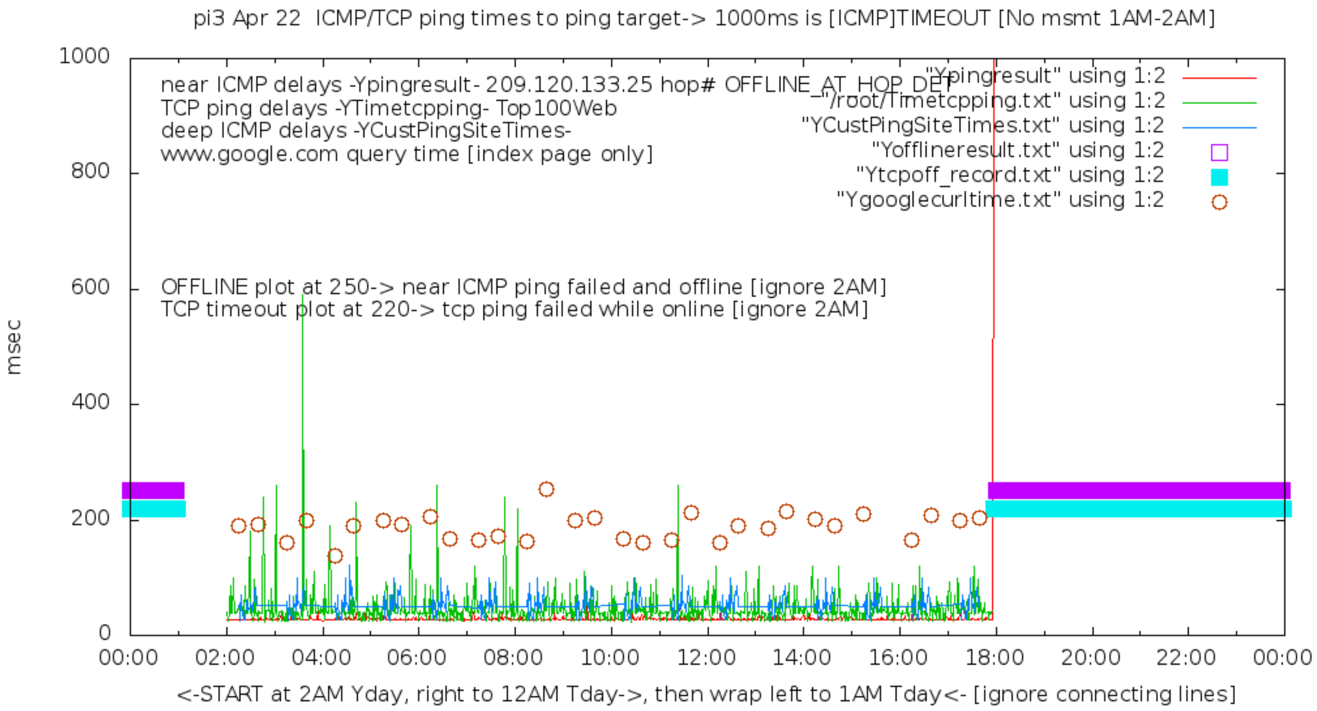
One of the neatest things you have access to is the daily [delivered via email if it is setup, otherwise available on the web page], and on-call real time plots of performance. Here is a sample plot received in the daily email.   Note: these plots are "dated" and current plots contain more info.....

There are two callable plots available on the raspberry pi web page:

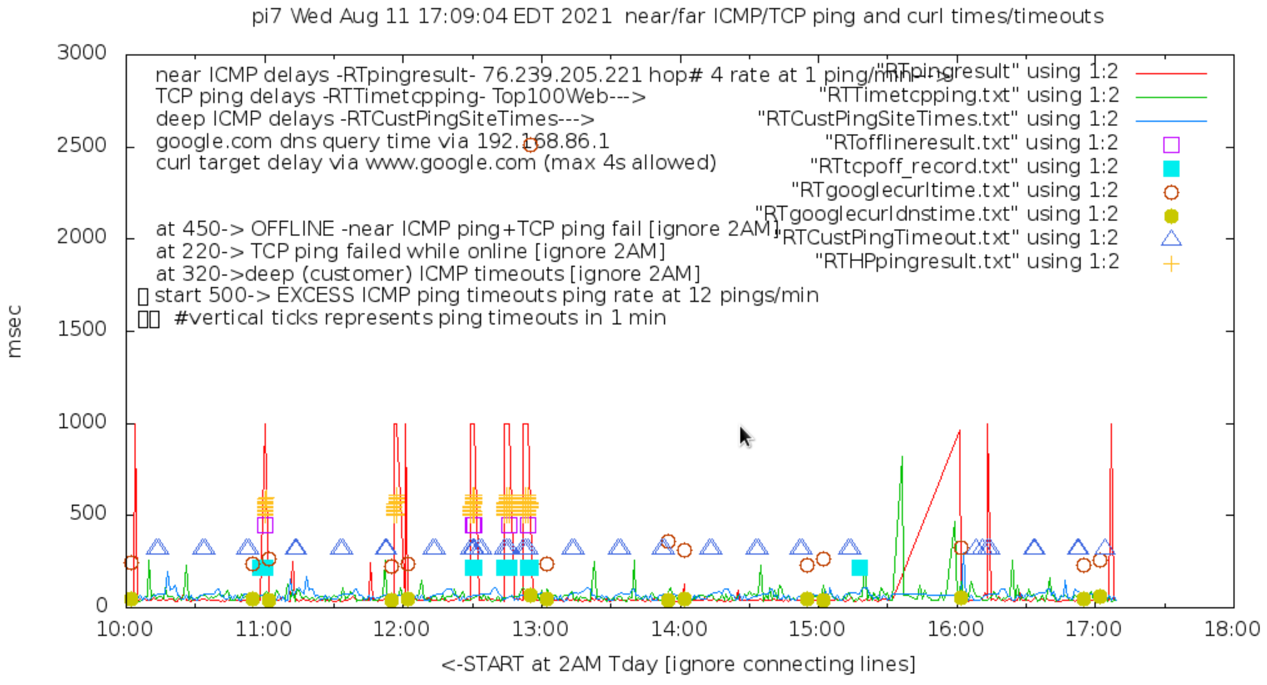
1. The combined plot, shown here, is the same received in the daily email, except it represents the plot from 2AM until the time it is requested.

2. A "ping blast" plot, which represents a ping blast of 60 ICMP echo requests to the ICMP target in the space of 60 minutes. This is only performed when requested.

3. There is also a "near-realtime" plot of the performance linked on the main rpi page. This will update the plot every 1 minute. --keep a tab open to this link.



The third plot is the current incarnation [8-2021] showing additional info. Somebody is having a bad day!.



15. Plot explanation

There are 2 main LINE plots: The ping delays of a close router [pingresult], and the TCP delays of top 100 web sites [Timetcpping]. These are the two colored lines. The close ping delays and the TCP delays are every minute for 23 hours [2AM till next 1AM]. Typically, the close ping delays should be less than the TCP delays and be fairly constant; the TCP delays will vary widely, normally between 20-100msec. The TCP delays will typically be to the nearest Content Distribution Network (CDN) which host most of the major web sites.

The third and fourth plot is the points that represent the twice hourly fetch of www.google.com - the curl time and the dns query time. This will typically have the longest delay. DNS name resolution will have to be performed for this, and for the TCP delays. The ping delays do not need a DNS name resolution.

The fifth and sixth "plot" is actually an overlay on the plots which will represent offline times experienced, and TCP timeouts experienced, but still on-line. These will appear as bars when there are extended periods of offline state, or multiple TCP ping failures. These are plotted at the 250 and 220 msec delay times.

Added in Aug 2021 is the "hyperping" plot. This will be ticks on the plot which designate ping timeouts at 6/min rate, invoked when the 1/min ping rate fails.

Implemented on the last few releases [but not shown on these example plots] are two additional plot lines:

-ipv6 connectivity. There will be a "roofline" if there is no ipv6 default gateway

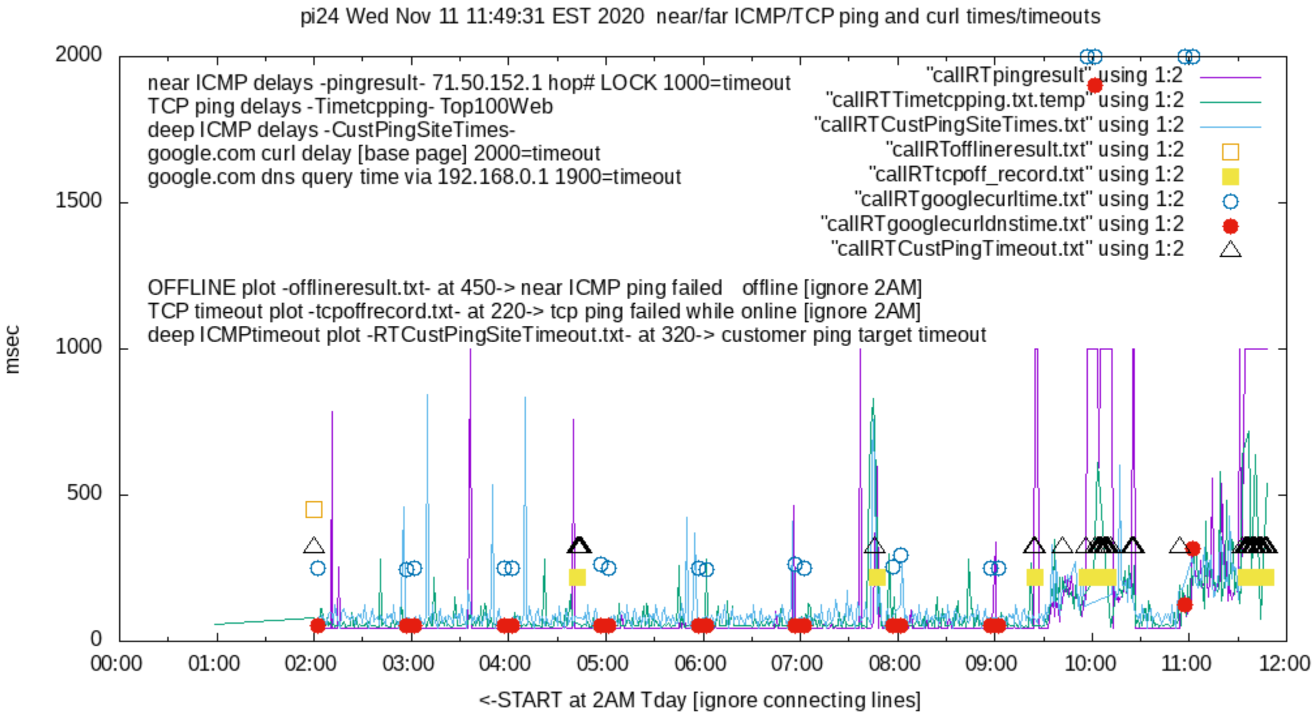
-router is used as near ICMP target, thus compromising Internet performance metrics.

As an option on the imonitorg, the customer can place a file "CustPingSites.newtxt" in the directory /home/pi/tests/customerpings/ and the script will take up this file at [the next] 2AM and perform "deep ICMP" ping plotting. Originally these sites represented the ping targets of imonitor [as opposed to imonitorg] customers, so they represented ICMP pingable sites 5- hops from each of the customers on the trial project. This file "CustPingSites.txt" was collected each night and copied to each pi for the use of the "deep ICMP" plots the next day. For the imonitorg of course there is no connection to the main imonitor server, so this file has to be manually constructed. There is an example in the "files" of the sourceforge site https://imonitorg.sourceforge.io -the user can copy this file and use it, or modify it and place it in the customerpings directory. This will only perform the pings for a 23 hour period starting at 2AM. If the customer wants to make this plotting "permanent," he needs to make a pi crontab entry to replace his file in that directory each day.

E.g. save the file as CustPingSites.mytxt in /home/pi. Make a pi crontab entry to cp the file to "CustPingSites.newtxt" each day anytime before 12AM.

```
25 23 * * * /bin/cp CustPingSites.mytxt tests/customerpings/CustPingSites.newtxt
```

This will copy the customer file to the directory each day at 11:25PM. The CustPingsiteScript will use this script for the next day starting at 2AM. Here is a [partial day using "create RTplot" on web page] sample plot showing the "deep ICMP" plotting:


pi24 Wed Nov 11 11:49:31 EST 2020 near/far ICMP/TCP ping and curl times/timeouts
near ICMP delays -pingresult- 71.50.152.1 hop# LOCK 1000=timeout
TCP ping delays -Timetcpping- Top100Web
deep ICMP delays -CustPingSiteTimes-
google.com curl delay [base page] 2000=timeout
google.com dns query time via 192.168.0.1 1900=timeout
"callRTpingresult" using 1:2
"callRTTimetcpping.txt.temp" using 1:2
"callRTCustPingSiteTimes.txt" using 1:2
"callRTofflineresult.txt" using 1:2 □
"callRTtcpoff_record.txt" using 1:2 ■
"callRTgooglecurltime.txt" using 1:2 ○
"callRTgooglecurldnstime.txt" using 1:2 ●
"callRTCustPingTimeout.txt" using 1:2 △
OFFLINE plot -offlineresult.txt- at 450-> near ICMP ping failed offline [ignore 2AM]
TCP timeout plot -tcpoffrecord.txt- at 220-> tcp ping failed while online [ignore 2AM]
deep ICMPtimeout plot -RTCustPingSiteTimeout.txt- at 320-> customer ping target timeout
msec
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
<-START at 2AM Tday [ignore connecting lines]