

DPC Latency Checker

Thesycon's DPC Latency Checker is a Windows tool that analyses the capabilities of a computer system to handle real-time data streams properly. It may help to find the cause for interruptions in real-time audio and video streams, also known as drop-outs. The program supports Windows 2000, Windows XP and Windows XP x64.

The DPC Latency Checker Tool

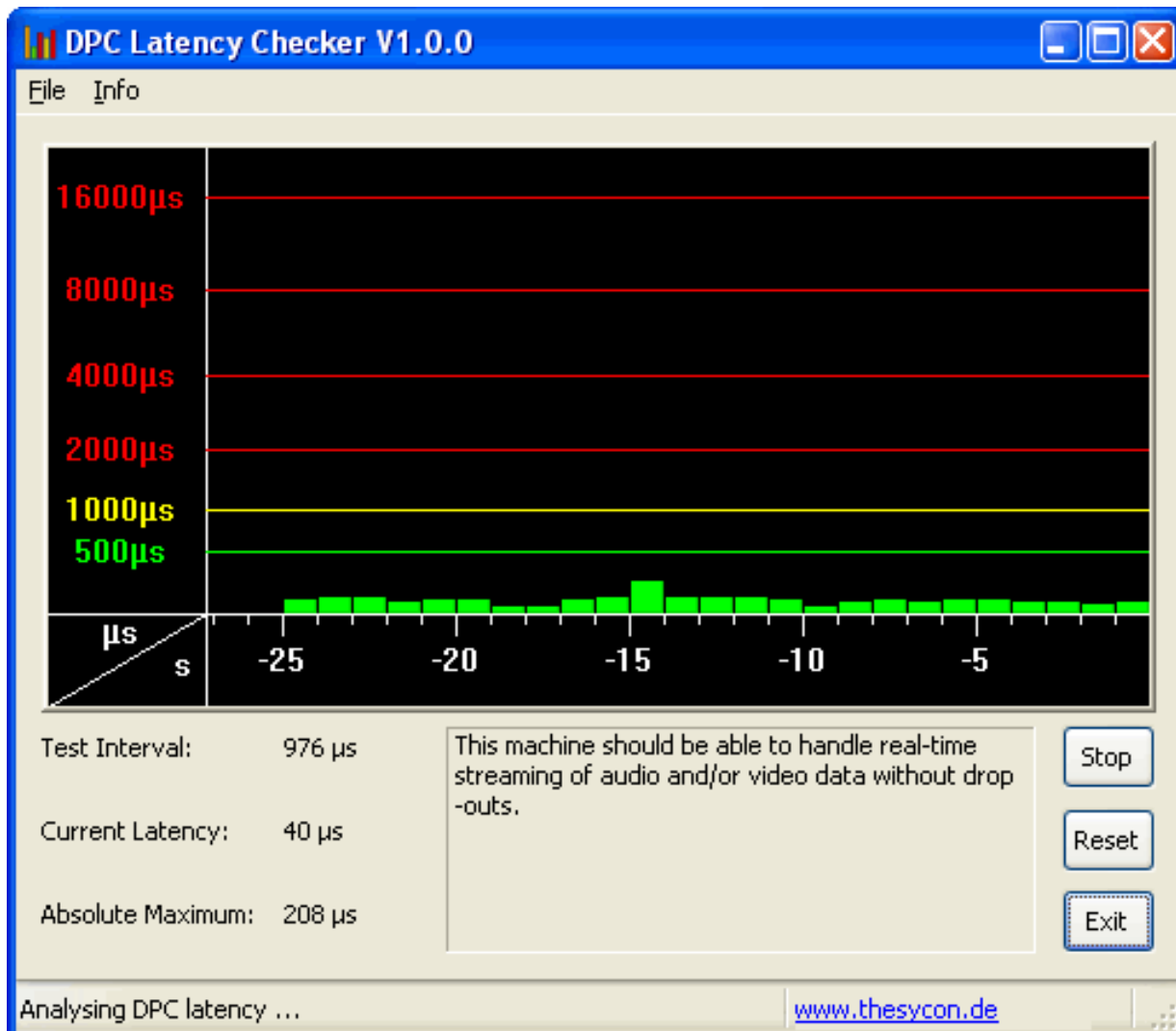
If any kernel-mode device driver in your Windows system is implemented improperly and causes excessive latencies of Deferred Procedure Calls (DPCs) then probably drop-outs will occur when you use real-time audio or video streaming applications. For an explanation of this effect see Background information below.

The DPC Latency Checker tool determines the maximum DPC latency that occurs on your Windows system and thus enables you to check the real-time capabilities of your computer. DPC Latency Checker works independent of any external hardware. Using this tool may be helpful in the following situations:

- ☐ You experience interruptions (drop-outs) in a flow of data processed in real-time, for example an audio stream, video stream or a sequence of measuring data, and you want to find out the reason for this problem.
- ☐ You want to verify that your Windows system is configured properly so that it is capable of handling real-time data transfer before you install the corresponding streaming application.
- ☐ You want to check if a particular computer system is suitable for streaming applications, for example before you buy this system.

Using DPC Latency Checker

DPC Latency Checker is very simple to use. Download dpclat.exe and launch it. No software installation is required. The main window is shown below.

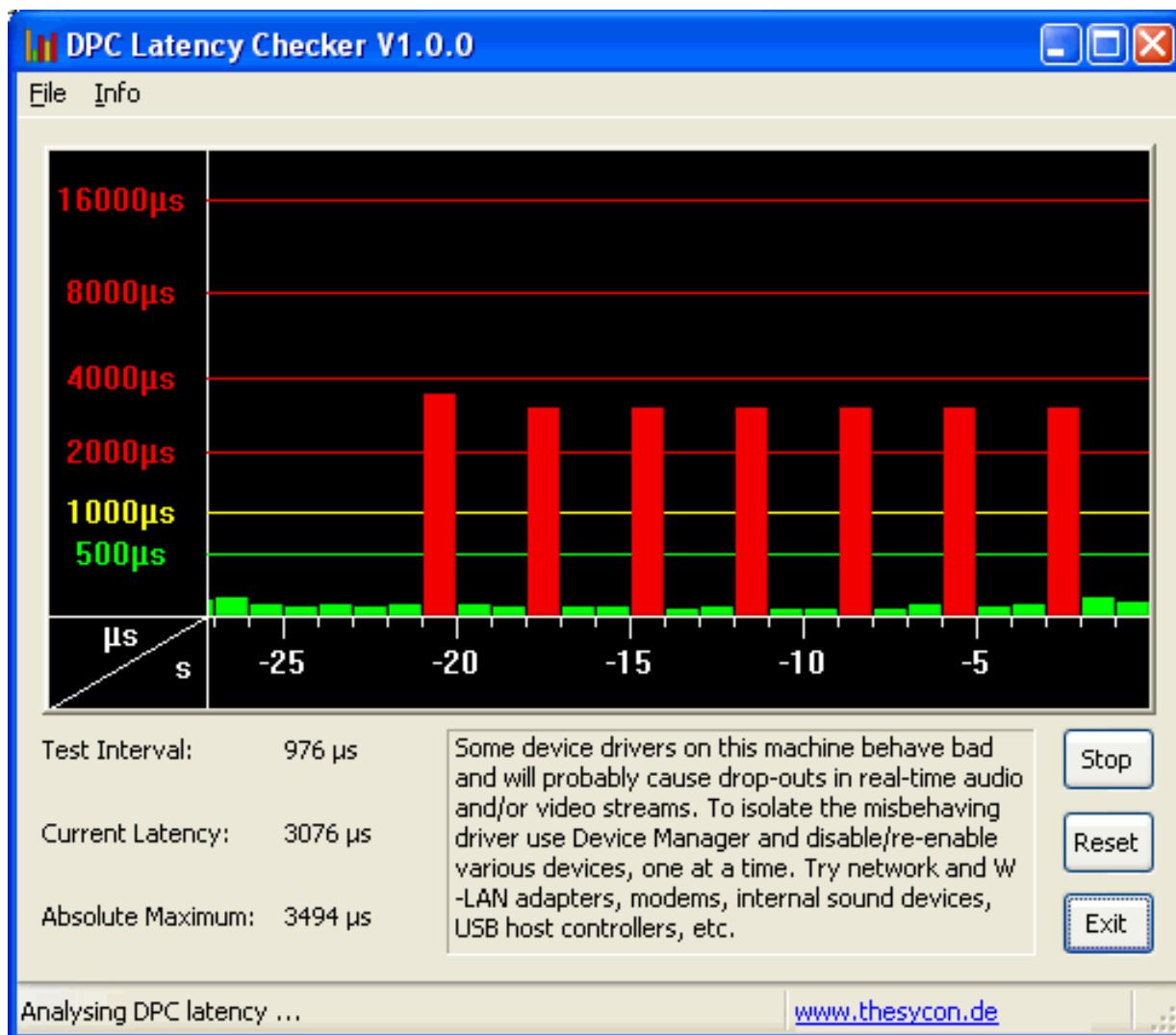


DPC Latency Checker periodically updates its internal statistical data at an interval displayed as *Test Interval*. This statistical data is queried and displayed on screen once per second. The *Current Latency* value indicates the maximum DPC latency measured within the last second. The value labeled *Absolute Maximum* represents the maximum latency measured since the tool was started. Use the Reset button to clear this value.

The bar graph shows the Current Latency value over time. Each bar represents the maximum DPC latency occurred within one second. The most recent value is shown as the right-most bar. Every second, bars are scrolled from right to left and a new bar is added at the right-hand side.

Latency analysis can be stopped by means of the Stop button. If the button is clicked again the tool restarts measuring of DPC latencies.

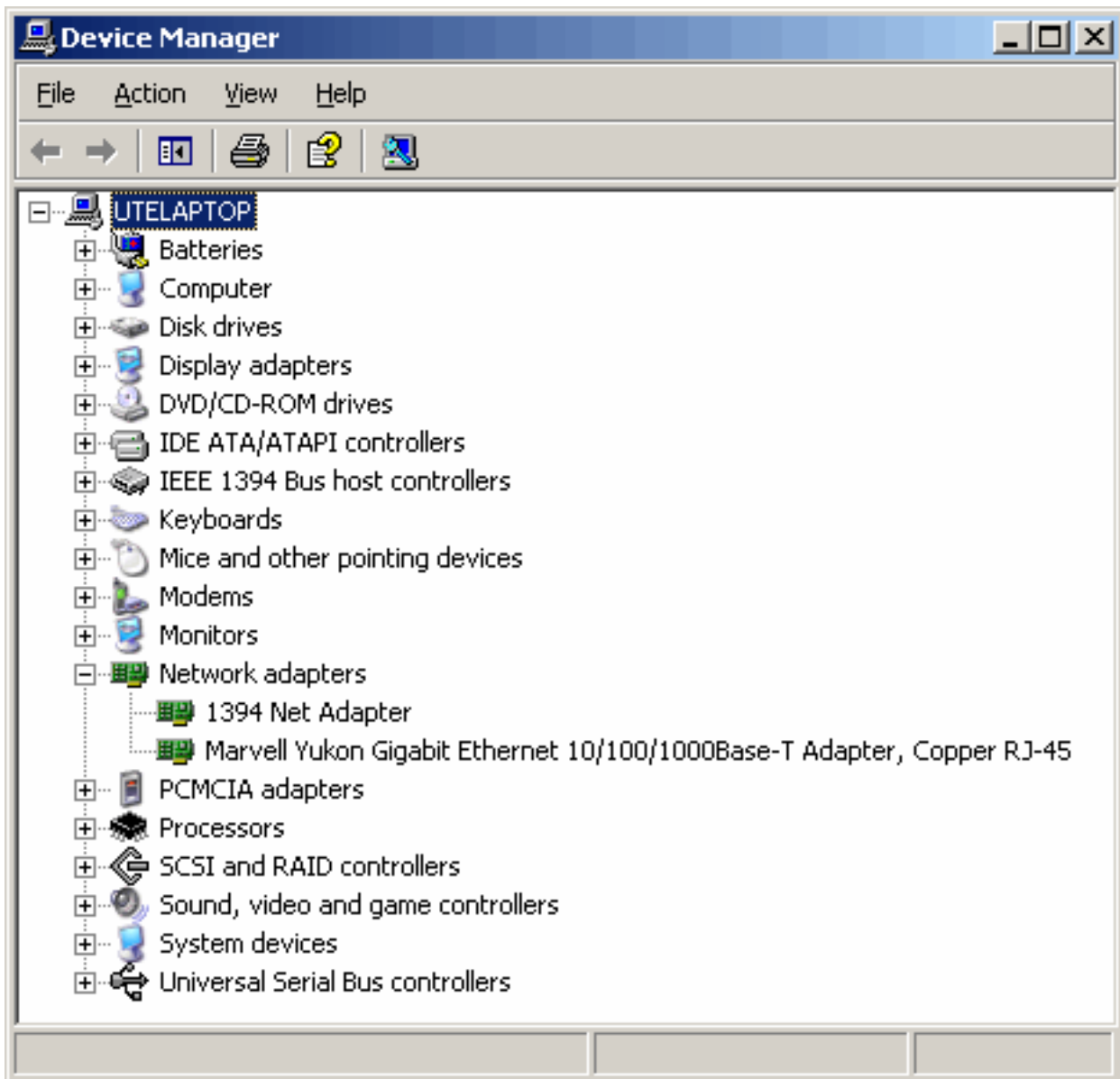
In the figure below, a typical problem scenario is shown. Occurrence of excessive DPC latency is indicated by a red bar.



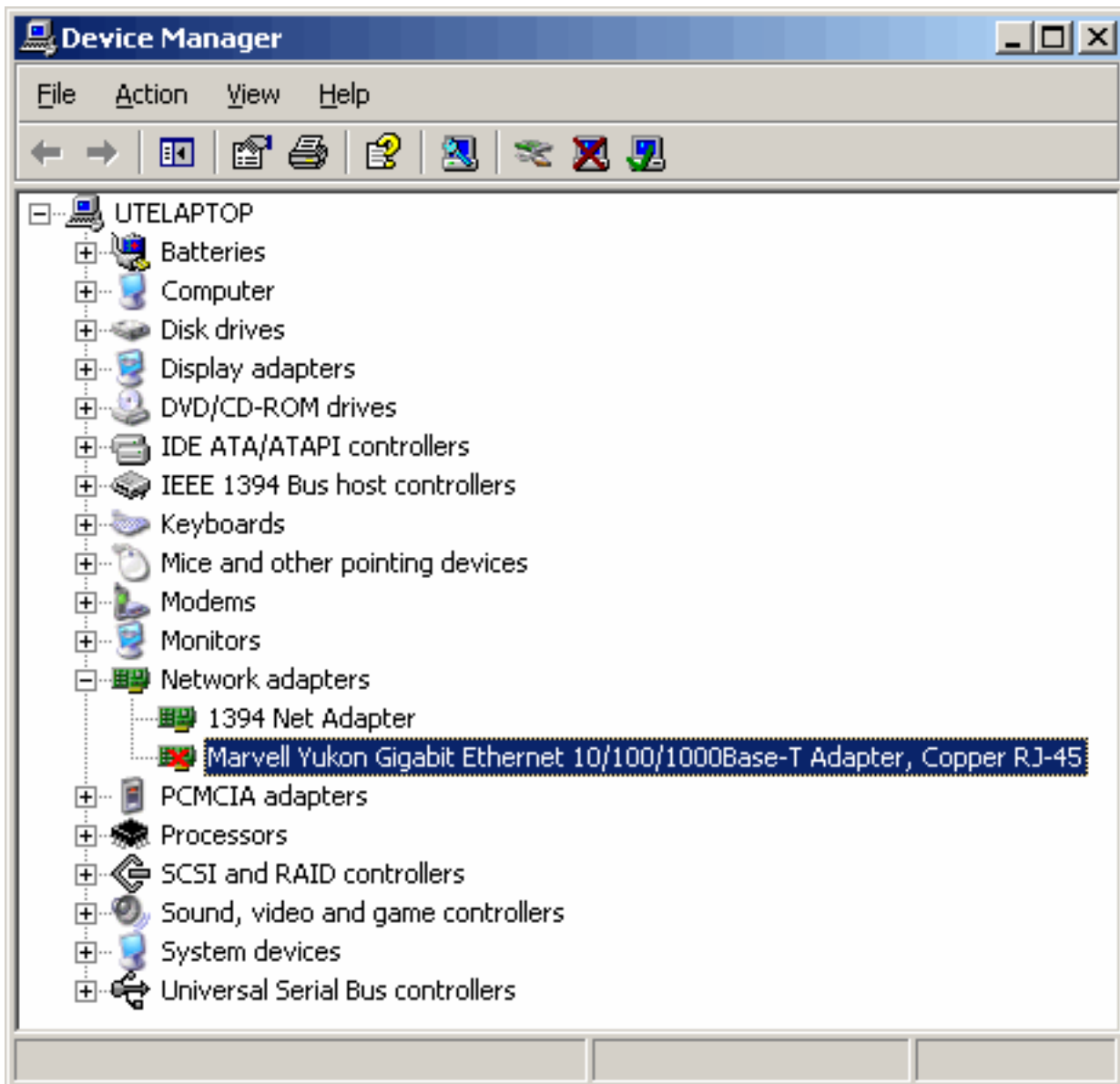
At a period of 3 seconds a kernel-mode driver in the system is stalling the CPU for about 3.5 milliseconds. So every 3 seconds DPC latency is exceptional large and drop-outs in real-time data streams will typically occur.

Analysing drop-out problems with DPC Latency Checker

You can find out the driver which causes the excessive DPC latencies by disabling individual devices using Windows Device Manager. To open Device Manager, choose Properties from the context menu of the My Computer icon on your desktop and click the Device Manager button on the Hardware tab.



In Device Manager disable individual devices, one at a time. To disable a device, right-click on the corresponding item (e.g. on the Ethernet adapter) and choose Disable from the context menu. Windows marks a disabled device with a read cross as shown in the figure below. To enable a device again, choose Enable from the context menu.



After you disabled a single device, carefully watch DPC Latency Checker. When excessive latency values disappear you found the responsible device driver. If there are still exceptional large DPC latencies try the next device.

In many cases DPC latency problems are caused by specific types of devices. So you should try the device types listed below first.

- ☐ Network adapters for Ethernet and Wireless LAN (W-LAN)
- ☐ Internal modems
- ☐ Internal sound devices (on-board sound systems)
- ☐ Any PCI or PCI Express add-on card, any PCCard or ExpressCard, e.g. TV tuner cards, ISDN or DSL adapters, modems, etc.

Warning: Don't disable devices that are essential for your computer to function!

You should **not** disable:

- ☐ any device listed in Device Manager under System devices or Computer,
- ☐ the hard disk that contains the system partition,
- ☐ the IDE/ATAPI or SATA controller this hard disk is connected to,
- ☐ the system keyboard,
- ☐ the mouse, track point or touch pad device,
- ☐ the USB controller external keyboard and/or mouse devices are connected to,
- ☐ the display controller listed under Display adapters.

When you have identified the device driver which is responsible for the drop-outs consult the device vendor's Web site or customer support to find an update for this driver. If this is not possible you may decide to keep the concerned device disabled while you are using streaming applications.

By using the method described above, in some cases it may not be possible to clearly identify a device driver that is responsible for the problems. In such a situation you could try to use the tool RATT available from Microsoft. However, RATT is not easy to use and analysis results might be difficult to interpret. To download RATT, search Google for "Microsoft RATTV3".

Background information: Why drop-outs occur

Processing of streaming data in real-time is a very challenging task for Windows based applications and device drivers. This is because by design Windows is not a real-time operating system. There is no guarantee that certain (periodic) actions can be executed in a timely manner.

Audio or video data streams transferred from or to an external device are typically handled by a kernel-mode device driver. Data processing in such device drivers is interrupt-driven. Typically, the external hardware periodically issues interrupts to request the driver to transfer the next block of data. In Windows NT based systems (Windows 2000 and better) there is a specific interrupt handling mechanism. A device driver cannot process data immediately in its interrupt routine. It has to schedule a Deferred Procedure Call (DPC) which basically is a callback routine that will be called by the operating system as soon as possible. Any data transfer performed by the device driver takes place in the context of this callback routine, named DPC for short.

The operating system maintains DPCs scheduled by device drivers in a queue. There is one DPC queue per CPU available in the system. At certain points the kernel checks the DPC queue and if no interrupt is to be processed and no DPC is currently running the first DPC will be un-queued and executed. DPC queue processing happens before the dispatcher selects a thread and assigns the CPU to it. So, a Deferred Procedure Call has a higher priority than any thread in the system.

Note that the Deferred Procedure Call concept exists in kernel mode only. Any user-mode code (Windows applications) runs in the context of a thread. Threads are managed and scheduled for execution by the dispatcher.

While there is a pre-emptive multitasking for threads, DPCs are executed sequentially according to the first in, first out nature of a DPC queue. Thus, a sort of cooperative multitasking scheme exists for Deferred Procedure Calls. If any DPC runs for an excessive amount of time then other DPCs will be delayed by that amount of time. Consequently, the latency of a particular DPC is defined as the sum of the execution time of all DPCs queued in front of that DPC. In order to achieve reasonable DPC latencies, in the Windows Device Driver Kit (DDK) documentation Microsoft recommends to return from a DPC routine as quick as possible. Any lengthy operation and specifically loops that wait for a hardware state change (polling) are strongly discouraged.

Unfortunately, many existing device drivers do not conform to this advice. Such drivers spend an excessive amount of time in their DPC routines, causing an exceptional large latency for any other driver's DPCs. For a device driver that handles data streams in real-time it is crucial that a DPC scheduled from its interrupt routine is executed before the hardware issues the next interrupt. If the DPC is delayed and runs after the next interrupt occurred, typically a hardware buffer overrun occurs and the flow of data is interrupted. A drop-out occurs.

Download DPC Latency Checker

Legal Notice: The dpclat.exe program is free for non-commercial use. It may be copied for private purposes. The program may not be included in any commercial software or software collection and may not be distributed without a written permission. Commercial licences are available. Contact dpclat@thesycon.de for more information.

The DPC Latency Checker "dpclat.exe" is available for download at <http://www.thesycon.de/dpclat> for free.