# Technical Brief

## FirstPacket Technology
## Improved System Performance

May 2006
TB-02434-001_v02

# Table of Contents

# Introduction

The NVIDIA® FirstPacket™ technology is featured with motherboards in the NVIDIA nForce® 600 and the NVIDIA nForce 500 Series.

FirstPacket improves performance for networked games and other latency-sensitive traffic by defining a high-priority transmit queue dedicated to user-defined applications. In effect, this allows latency-sensitive packets to bypass less important data transfers and respond more to applications.

## Bandwidth vs. Latency

Traditionally, the metric used to compare networking performance was *bandwidth.* Higher bandwidth meant better performance. That's why it's easy to see that a DSL line is better than a modem; it's approximately 1,000,000 bits per second (1 Mbps) versus 56,000 bits per second (56 Kbps). Similarly, cable modems (3 to 10 Mbps) outperform DSL.

However, newer applications are emerging—networked games, voice over IP (VoIP), and others—that don't send a lot of traffic; their bandwidth needs may average 100 Kbps, but they are sensitive to *latency.*

Latency is a measure of the roundtrip time between two communicating devices. Higher latency means that it takes longer for the two machines to exchange information. In games, the state of the players changes rapidly and you could be "dead" before you know it if the latency is too high.

Ideally, when latency-sensitive applications are present, they should have preferential access to the network interfaces in the system.

# The Problem

## Latency-Sensitive Applications Are Unknown

In a typical PC, the network hardware and driver software (and the operating system) are unaware of—and unable to reduce—latency. The interfaces that allow applications to send and receive data are effectively identical, whether the application is a latency-tolerant application like FTP or a Web browser; a latency-sensitive application like a game; or VoIP applications like Skype, Gizmo, or any IM-based VoIP applications such as Google Talk, Yahoo! Instant Messenger, and MSN Messenger.

It's not possible to update the operating system (OS) application protocol interfaces (APIs) to enable latency to be influenced by a parameter associated with each socket. Even if the OS could be updated to provide such APIs, the application developers would have to modify their code to take advantage of the new or modified APIs, and those applications would only work best on the newer OS. Fortunately, it is possible to make some measurable improvements just by adding some intelligence to the network driver, leaving the OS and the applications alone.

Traffic handling within the PC can be influenced in the *transmit direction* (TX, upstream), but traffic arriving in the receive direction (RX, downstream) is not controllable because a PC has to receive the packets as it gets them, which is one at a time. Also, the network switches and routers may drop or reorder traffic, which can affect the performance of the applications, whether or not they are latency-sensitive.

## Large Data Transfers Increase Latency

The key point is that most latency-sensitive applications send very small packets, but the level of system activity can differ:

- In an essentially idle system, one in which there are few or no other active network applications, these latency-sensitive small packets can be transmitted as soon as they are ready to go, with minimal queuing delay.
- In systems that have other active applications, there is no guarantee that the latency-sensitive packets will go out in any order other than first-come, first-served. What's worse, the amount of delay suffered by two independent packets may be vastly different. This is because some applications (like file transfer or file sharing) tend to transmit as much data as a packet will hold, and these packets take a correspondingly longer time to move onto the wire, which interferes with the ability of the system to transmit the smaller latency-sensitive packets.

This variability of delay in transmission is known as *jitter*—the receiving system sees that the packet arrival times have a large amount of jitter, which isn't caused by the network but happened before the packets even left the sending PC.

To get an idea of the delay that a packet might cause, consider that a minimum-sized Ethernet frame is 64 bytes, and that approximately 20 bytes of timing overhead are associated with each frame. At gigabit Ethernet speeds, where each bit is transmitted in 1 ns, the whole frame is transmitted in 8*(64+20) = 672 ns, which is a little over two-thirds of one microsecond. On the other end of the spectrum, a full-sized Ethernet frame would take 8*(1518+20) = 12,304 ns, which is over 18 times longer!

It is obvious that an application that sends only a few large frames can cause a substantial additional delay for a latency-sensitive application. In a single-queue system, once a frame is in the queue for transmission there is no way to re-order it to allow a smaller frame to leave the system sooner.

# The Solution

## FirstPacket Technology

Applications that tend to send maximum-sized frames also tend to be tolerant of delays in network delivery. FirstPacket technology leverages that tolerance to the benefit of the latency-sensitive traffic. Figure 1 shows how, in a typical system, game packets can be slowed down by FTP packets because they share a common transmit queue.

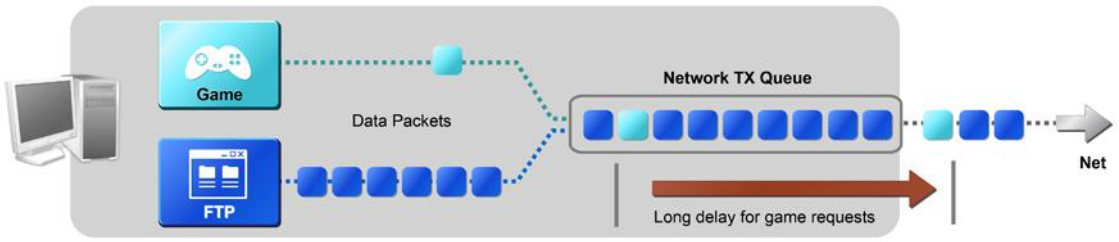


Figure 1. Traffic Behavior without FirstPacket

Figure 2 illustrates how traffic is handled with FirstPacket. In short, FirstPacket technology creates a two-queue system, with what is effectively a “fast lane” and a “slow lane” inside the Ethernet driver, and the small packets from user-approved applications are given access to the fast lane.

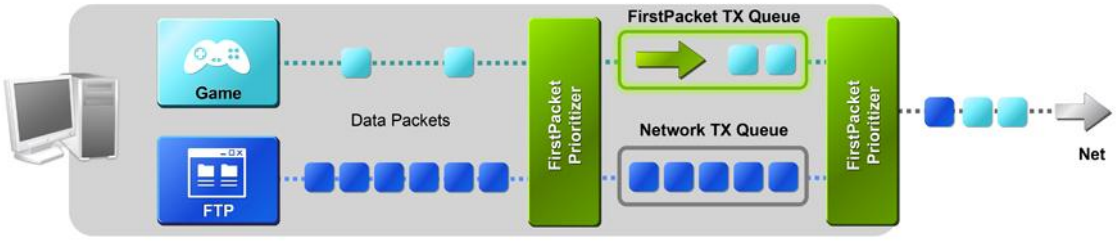


Figure 2. Traffic Behavior with FirstPacket

Applications can only use the fast lane with the user’s permission. The slow lane will drain its traffic fast enough so that the connections will not time out, but not fast enough to interfere with the traffic in the fast lane. The reason that the fast lane is not simply the “small packets lane” is that not all applications that send small packets are latency-sensitive. Even if they were, the user may prefer to give some applications improved performance and let others fight it out in the slow lane.

Remember, the FirstPacket technology fast lane is not higher bandwidth; it’s the “bounded-latency lane.” There is no way for a PC to increase the speed of the upstream link—the upstream bandwidth is a mostly static characteristic of the user’s

Internet broadband access. Whether the user has DSL or cable modem, the upstream bandwidth is limited to a small multiple of 100 Kbps. All the driver can do is ensure that the packet gets onto the upstream link as expeditiously as possible. Ultimately it is up to the user to avoid sending their system mixed messages; users who want to prioritize traffic from latency-sensitive applications should not simultaneously send large amounts of latency-tolerant traffic.

# Benefits

## Manages Traffic

The benefit to the user is that they have a new way to manage the traffic in their PC, letting them more effectively mix latency-sensitive and latency-tolerant applications so that the performance of both types of applications is acceptable. In fact, the performance of latency-sensitive applications may be completely unacceptable on a busy system unless FirstPacket technology is available and enabled.

## Resolves Performance Issues

FirstPacket technology operates by creating an additional transmit queue in the network driver. This queue is used to provide expedited transmission for user-approved applications, which presumably are latency-sensitive applications (although the user is free to choose any application).

When these applications are allowed preferential access to the upstream bandwidth (not necessarily guaranteed, but with far less jitter than would otherwise be the case in a single-queue design), their system's performance is greatly improved. A game that was unplayable before may be completely usable. And, VoIP connections will not be dropped due to the traffic in the local PC. (No guarantees can be made about how the network will treat the traffic after it leaves the PC).

## Improves Many Applications

Any application that tends to use small packets and requires bounded end-to-end latency will benefit from FirstPacket technology. Any networked game, plus voice over IP, video over IP, and any interactive applications (network meetings, like WebEx) will benefit.

The benefit is achieved by forcing the latency-tolerant applications to wait a bit longer to access the wire. This is a good trade-off, since the user has indicated that their preference is to have the latency-sensitive applications be given priority over the other applications.

# Performance

One way to measure the effect of FirstPacket technology is to measure the "ping" of a game playing across the Web to a game server. Testing at NVIDIA was done measuring the ping of a game played without other network traffic, and then a large FTP upload was started and the ping was again measured. Finally, FirstPacket was enabled and a new measurement was taken. These results are summarized in Figure 3. The results show that FirstPacket can reduce the ping of an Internet game by up to 50 percent, depending on the amount of traffic being bypassed. In addition, heavier traffic contention means that FirstPacket will reduce latency more.

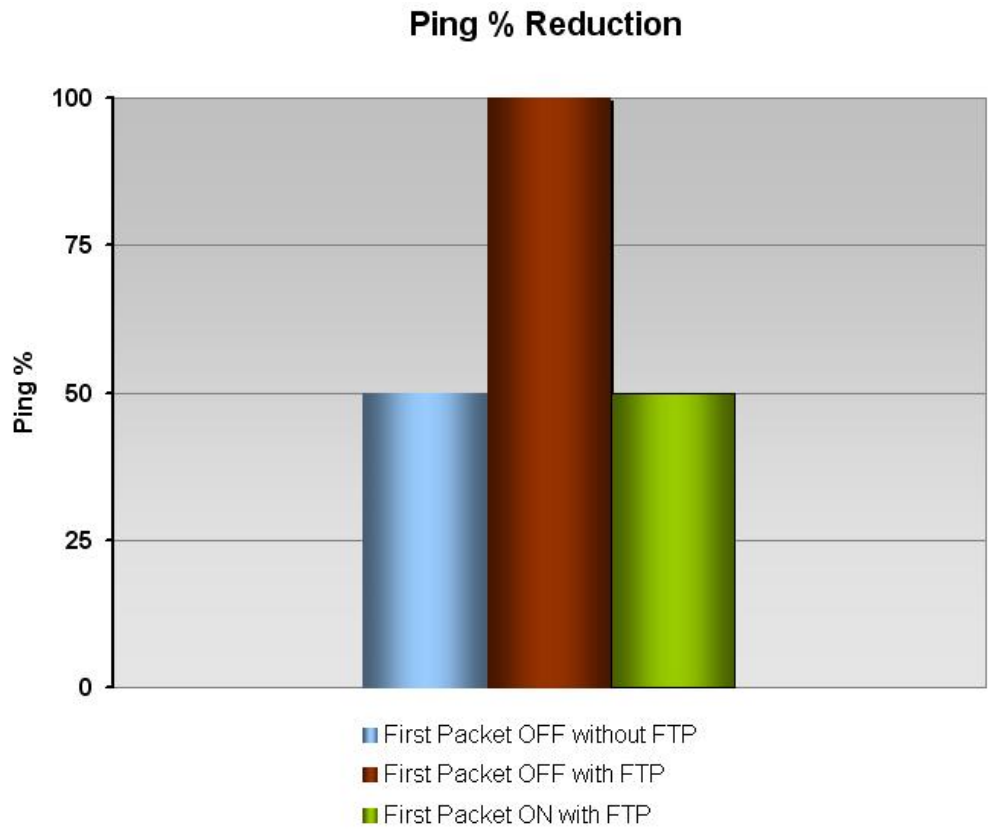


Figure 3. Effect of Ping Time Performance

# Conclusion

NVIDIA FirstPacket is an effective technology that allows latency-sensitive applications to effectively share a limited resource—the upstream bandwidths—so that preferred applications do not notice the operation of other applications that are latency-tolerant. Likewise, the latency-tolerant applications may barely notice that they have been relegated to the “slow” lane.

FirstPacket technology gives users much more flexibility in how they may use their computers, allowing them to successfully do more things at the same time.

**Notice**



**Trademarks**



**Copyright**



NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com