

# F5 Fast L4 Acceleration and the F5 Smart Coprocessor (prioritized Fast L4 Acceleration)

## How the features work and the benefits they bring

White Paper

Authored by Ryan Romney

### **Executive Summary:**

F5 has recently introduced its Smart Coprocessor. This uses a new technology which provides significant enhancements to its better-established Fast L4 Acceleration, by intelligently prioritizing flows. This paper provides a description of the Fast L4 Acceleration technology followed by a description of the Smart Coprocessor.

F5 systems act as full proxies terminating traffic between clients and servers. Some workloads require very little manipulation by the system but burden the processor with traversing its entire network protocol stack to process every packet. This processing also adds latency and jitter into the processing of packets within a given flow.

F5 has significantly improved system performance and customer experience by offloading these repetitive workflows allowing this traffic to be processed by a Field Programmable Gate Array (FPGA). This whitepaper answers the following questions: How much does FPGA acceleration benefit F5 customers and systems? Why did F5 choose to offload these processes onto FPGAs? What would the impact be of restricting this work to be executed by system software instead? What if the processes that are offloaded can be selected and prioritized?

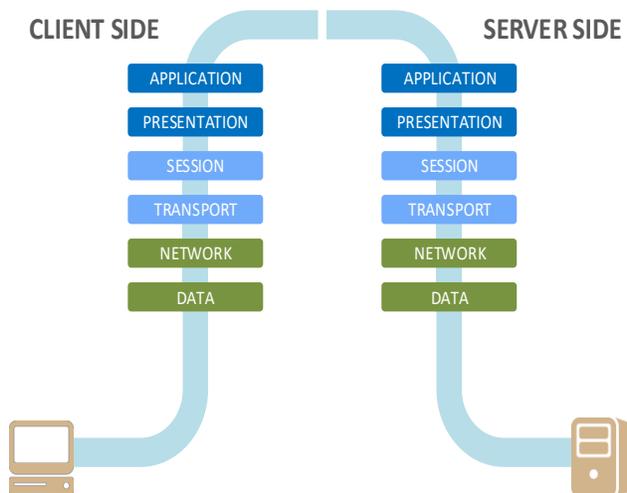
Reduced CPU utilization, latency and jitter are critical to large enterprises and service providers. 5G networks require high performance and minimal latency as applications get closer to the network edge and the need for Multi-access Edge Computing grows. 5G will result in increased compute intensive traffic loads, such as QUIC (Netflix, Youtube), HTTP2 or SSL, which need to be managed effectively. This paper shows how Fast L4 Acceleration and the Smart Coprocessor technology does this.

### **Fast L4 Acceleration**

#### **The Big-IP Environment:**

Big-IP systems are deployed as full proxies between clients and servers. This necessitates that Big-IP systems process all network traffic that is being transported between these clients and servers. With a full proxy architecture, Big-IP systems must manipulate, inspect, drop, and generally do what is required for all traffic from both sides and in both directions.

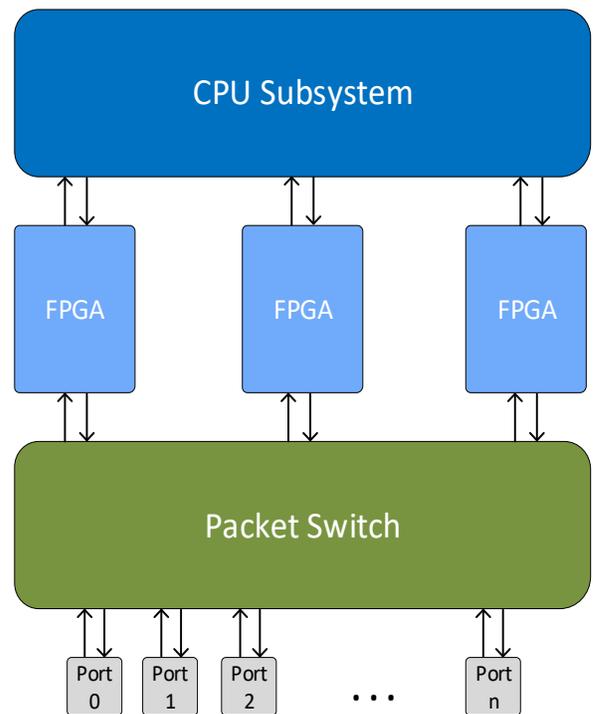
Big-IP systems are designed with a data path that aggregates traffic via network ports to an internal switch. The switch distributes the traffic to the CPU subsystem via FPGA bridges that buffer the traffic and translate between network protocols. Big-IP software can instruct the CPU subsystem how to perform key ADC



functions like load balancing, applying security policies, performing encryption and compression, as well as address translation and more.

CPU workloads associated with individual packets and traffic flows vary widely. One flow may need only a straightforward address translation while others might require more in-depth processing to enforce security protocols, apply data compression, mitigate flow fragmentation and more. Every task executed by the CPU subsystem occupies CPU cycles. Fast L4 Acceleration attempts to offload some tasks from the CPU to the FPGA to free up the CPU cycles associated with performing those tasks in favor of executing other workloads. Offloading these tasks also dramatically improves the latency associated with these operations.

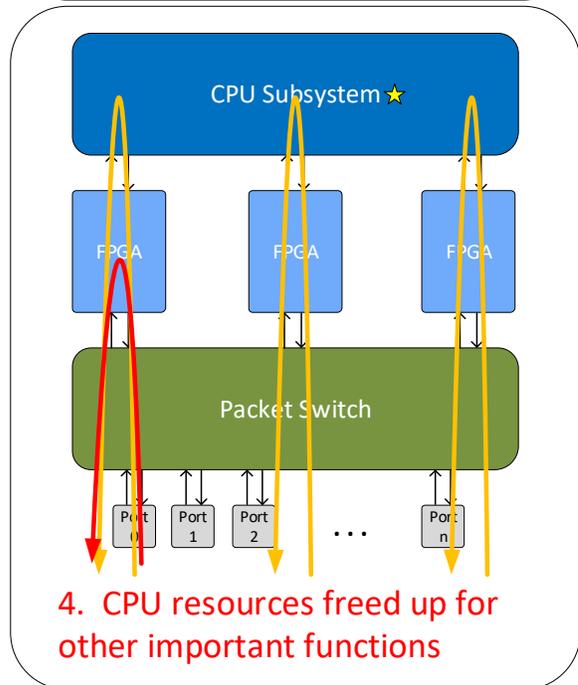
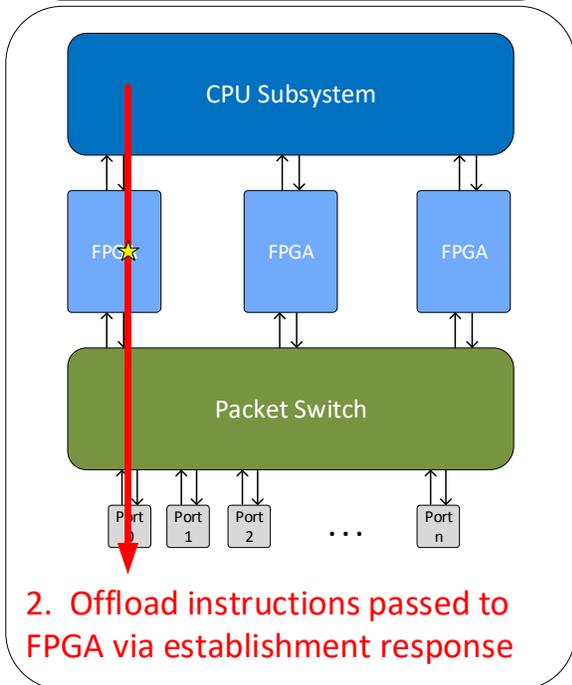
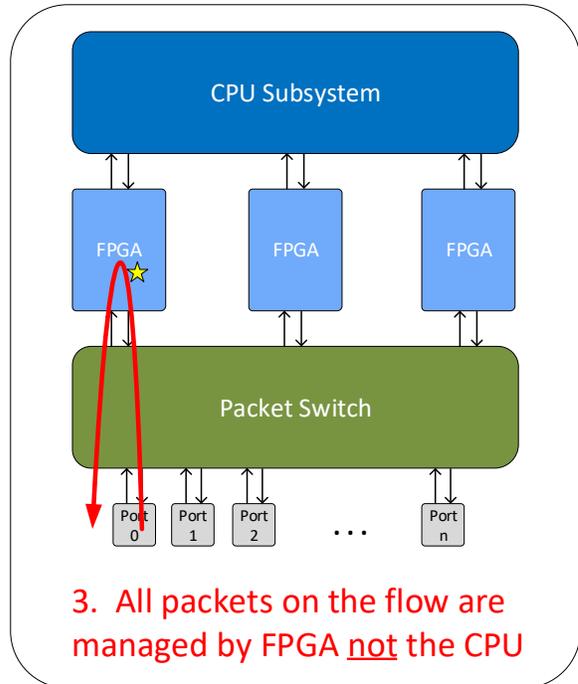
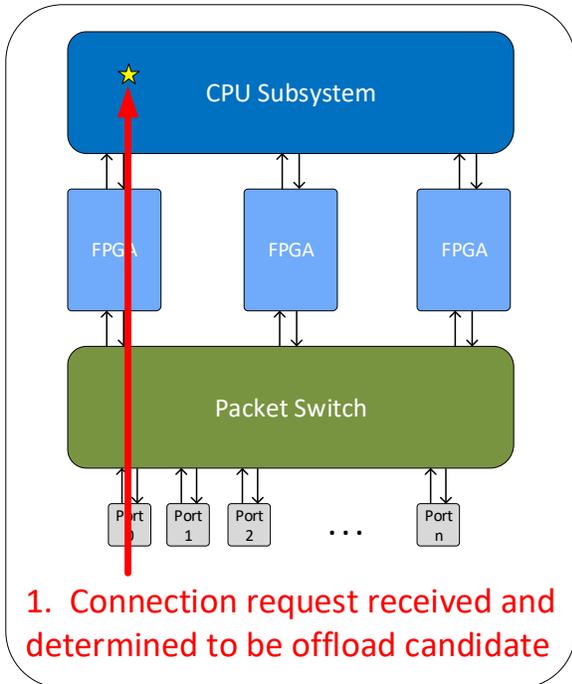
Fast L4 Acceleration leverages the ability of the FPGAs in the system to be programmed for system specific functions using a high-level programming language. This allows for Big-IP systems to reallocate their CPU resources away from mundane tasks like address translations and protocol management to higher value tasks which are not so easily offloaded.



Fast L4 Acceleration attempts to improve overall system performance by freeing up CPU resources, reducing packet processing latency and jitter, as well as potentially increasing throughput.

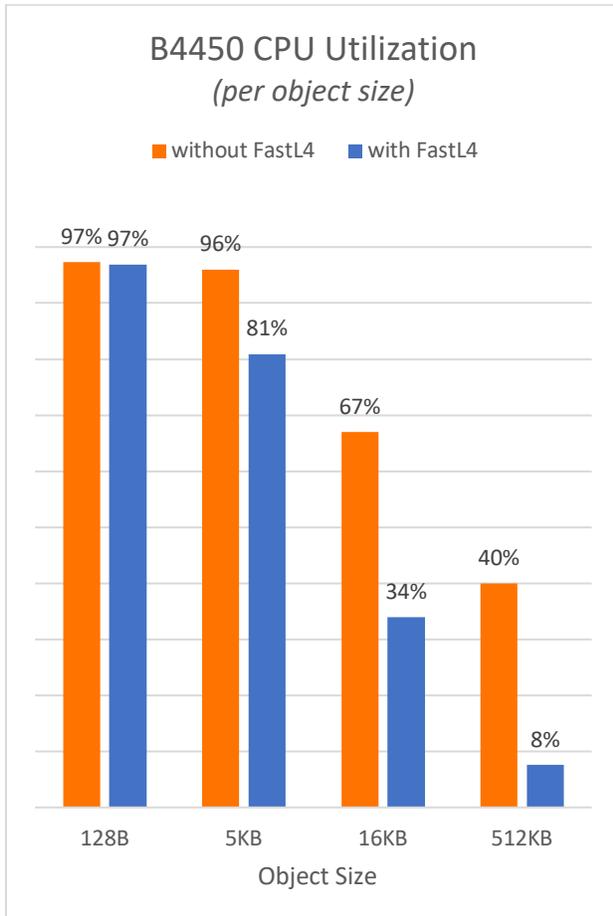
### How Does Fast L4 Acceleration Work?

As the CPU subsystem establishes TCP and UDP flow connections on behalf of clients and servers that the system is proxying, it identifies flows that would be good candidates for being offloaded into the FPGA. Once identified, the CPU subsystem communicates its request to offload these flows to the FPGA. These requests are passed embedded within the initial flow establishment responses that the CPU sends to the appropriate client/host across the system's data path. Once the FPGA is informed of the flow offload request it can then process and reroute all data packets on the specified flow without relying on the CPU or any of its resources.

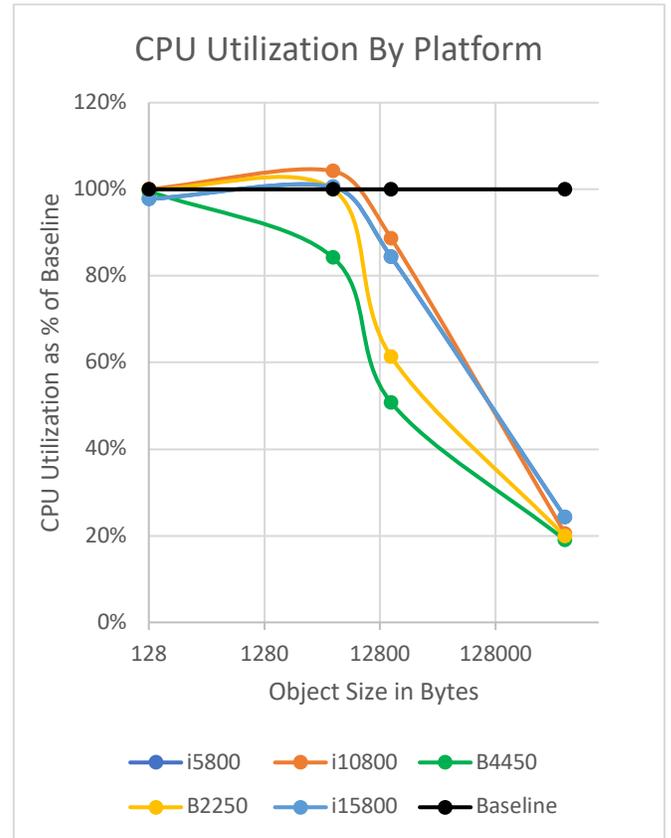


### CPU Impacts of Fast L4 Acceleration:

The key objective of offloading workloads from the CPU Subsystem to the FPGA is to free up precious CPU cycles from straightforward, frequently executed tasks like L4 packet processing in favor of other tasks which a general-purpose processor is better suited to perform. The success of this effort is evident in two metrics that demonstrate how much impact the offload has had. The first and simplest is CPU utilization. It is evident that a B4450 blade can achieve increasingly better CPU cycle savings as flow object size increases in the presence of Fast L4 Acceleration.

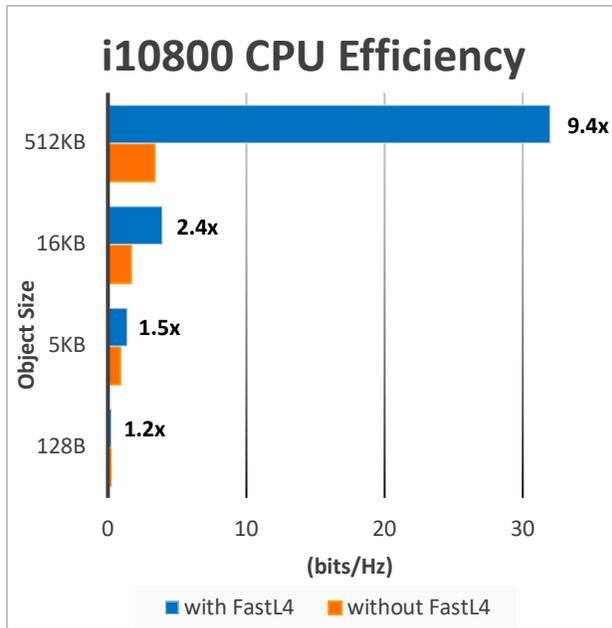


Similar CPU Utilization savings are seen across all Big-IP platforms. It is very consistent and universal regardless of system configuration, processor selection, core counts, etc.



The second metric is CPU efficiency. CPU efficiency quantifies how efficiently the CPU is using every Hz of processing time to move every bit of object data transacted by the system. The following graph shows how the i10800 platform during characterization was able to achieve more than 9 times the efficiency of a non-accelerated baseline run.

Efficiency is also impacted by object size. The larger the object, the more efficient the processor is at transacting object data.



### Enhanced throughput due to Fast L4 Acceleration:

Fast L4 Acceleration can achieve significant gains in throughput for systems with overprovisioned network bandwidth. Most Big-IP systems are architected to support twice the L4 bandwidth as the L7. This means that the bandwidth capacity between the FPGA and the network ports is double the bandwidth capacity to the CPU subsystem. These bandwidth relationships are detailed in the table below.

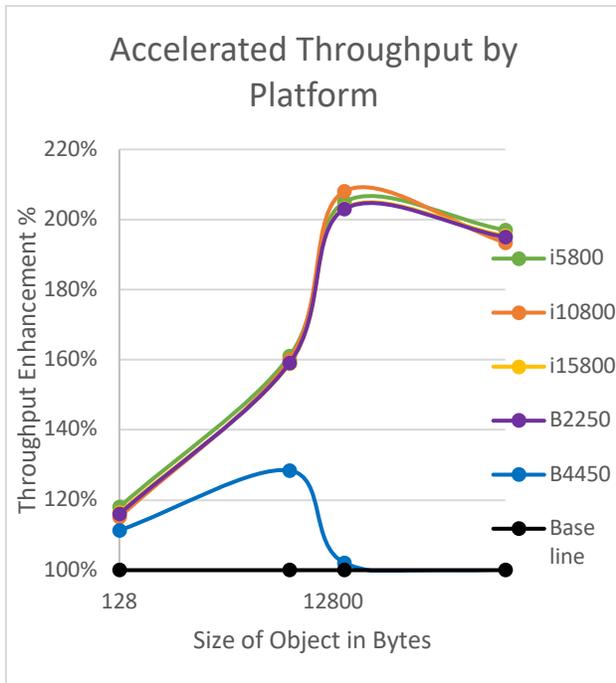
Platform	Ratio L4/L7 Bandwidth	L4 Bandwidth	L7 Bandwidth
i5800	2x	80 Gbps	40 Gbps
i7800	2x	80 Gbps	40 Gbps
i10800	2x	160 Gbps	80 Gbps
i15800	2x	320 Gbps	160 Gbps
B2250	2x	160 Gbps	80 Gbps
B4450	1x	240 Gbps	240 Gbps

### Why does flow object size impact CPU efficiencies?

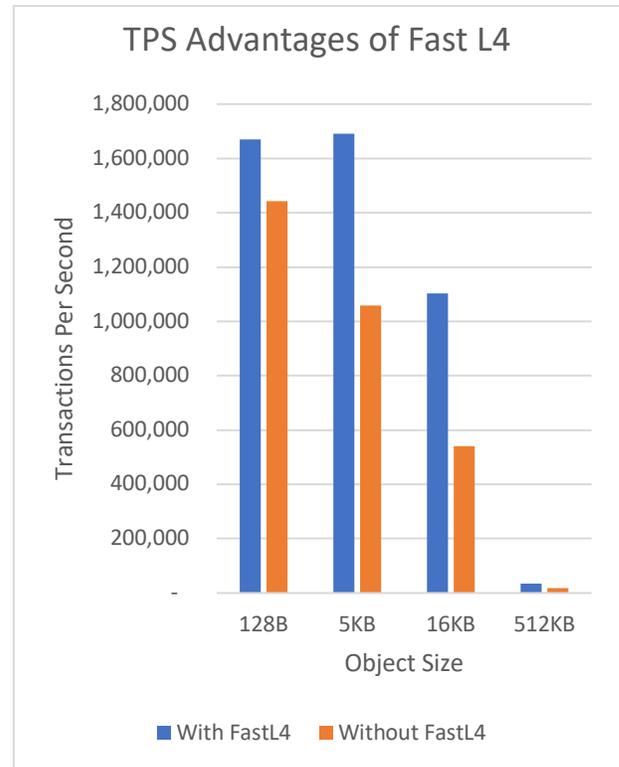
The setup and tear down of flows requires transmission of several small handshake packets. These packets generally contain no data. The passing of these packets is necessary to setup and tear down a flow, but they penalize the efficiency of the system because they aren't used to transact flow object data. So as flow objects increase in size a smaller fraction of the data passed is attributable to setup and tear down. Conversely a larger portion of the transfer is object data. Thus, larger objects are more efficient than smaller ones.

Networks vary widely, but it is much more common for a network to be occupied transacting large objects rather than small ones. This lends itself well to Fast L4 Acceleration since the larger objects reap even more of the benefits of the feature over smaller objects.

The following graph shows how Fast L4 Acceleration can take advantage of the additional bandwidth to the FPGA on these overprovisioned systems to deliver double the throughput over a non-accelerated system. It is demonstrated how the throughput of B2250, blade can be greater than 2x while the B4450 blade throughput remains relatively constant.



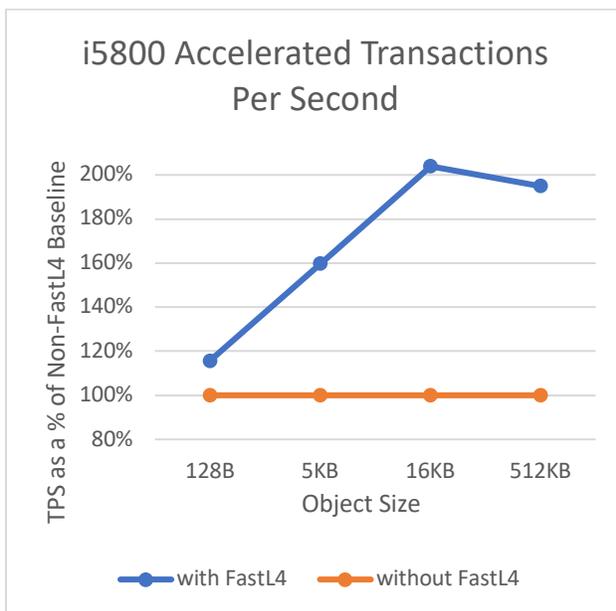
The number of transactions per second (TPS) of these systems scales linearly with throughput. This allows these overprovisioned L4 systems to also see a doubling in TPS as well as in throughput.



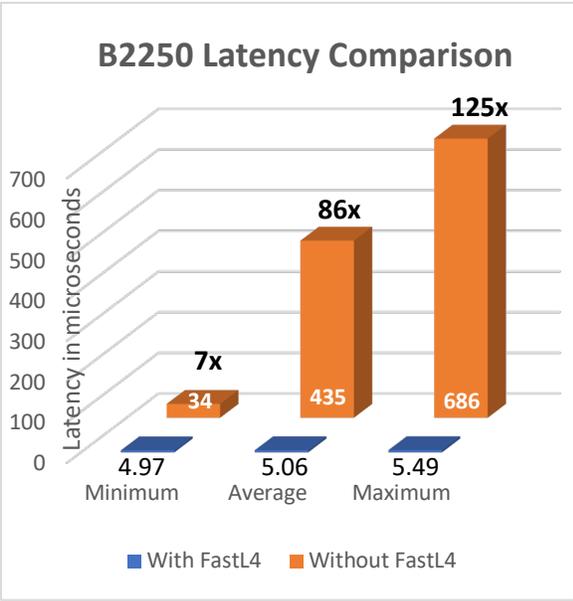
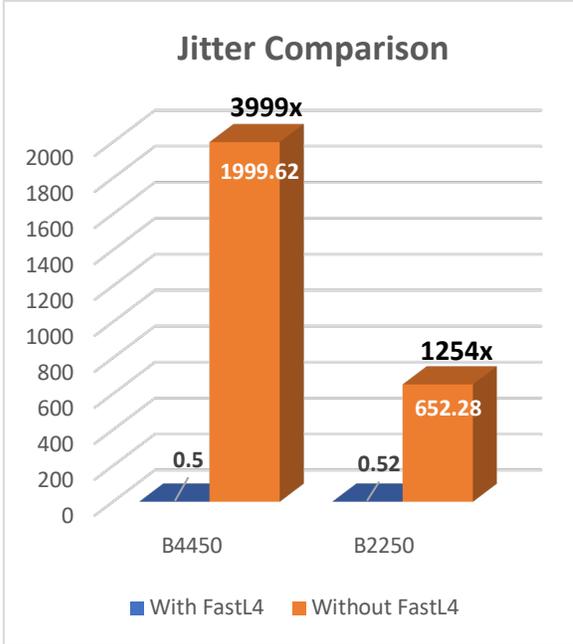
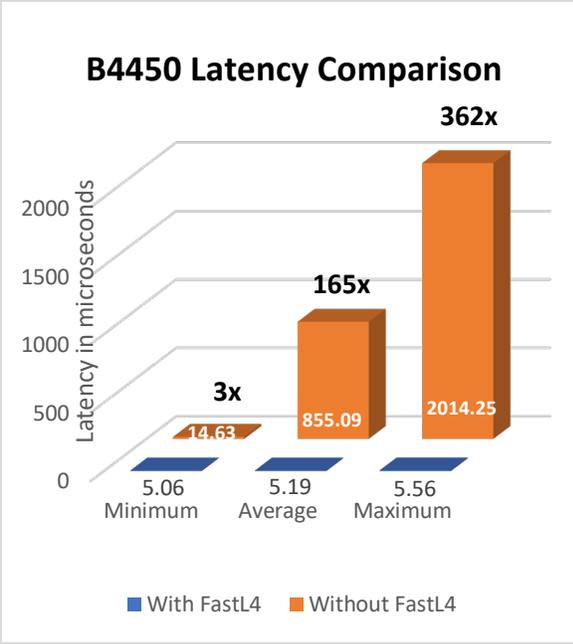
**Improved latency due to Fast L4 Acceleration:**

Accelerated flows are not operated on by the CPU. As such they are not required to be transmitted across the PCIe bus to or from the CPU subsystem. This not only saves the time required to transmit the packets across that bus but also the time to buffer and extract these packets from FPGA memories. These packets also avoid all processor time required to cache, parse, transform, and generally be operated upon. This has positive implications on system latency.

Below are examples of network packet latency through two Big-IP systems. The latency without the Fast L4 offload is hundreds of times longer than with the Fast L4 offload. This has a significant impact on latency sensitive applications like VoIP.



The following graph depicts the raw number of transactions per second binned per object size.



Not only is the latency reduced by multiple orders of magnitude, but the jitter is also dramatically smaller in the presence of Fast L4 Acceleration. The delta between maximum and minimum latencies is a fraction of a percent of what it would be without this offload enabled.

**Where does latency and jitter matter?**

Latency and jitter are of increasing relevance in today’s edge networks. Along with ever increasing demands for more bandwidth, latency and jitter are becoming key factors of a network’s performance and capability. Future network enabling technologies like 5G, Edge Computing, IoT, Virtual Reality and Augmented Reality all will depend on fast dependable application acceleration. That means the lowest latency and jitter possible each time a packet is touched. FPGAs excel at this and will be a key enabler in the networks of the future.

**Fast L4 Acceleration provides significant performance gains:**

Fast L4 Acceleration significantly increases the system performance of Big-IP platforms. This offload lowers CPU utilization to less than a fifth of its non-offloaded amount. It also increases CPU efficiency by nine-fold. System throughput was also doubled for most Big-IP systems enabling Fast L4 Acceleration. This offload also decreased network latency by as much as 99.8% and jitter by 99.98%.

These performance gains were made due to the ability of the system to offload significant CPU workloads onto the FPGA(s) available in the system. Software re programmability of FPGAs allow them to be programmed to address a myriad of other functions currently being allocated to the CPU subsystem.

## F5 Smart Coprocessor (Prioritized Fast L4 Acceleration)

The Fast L4 Acceleration feature is well accepted by F5 customers because of the numerous benefits it provides. Currently, F5 products support a Fast L4 flow cache capacity of up to either 128K or 256K flow entries. This serves customer networks that have concurrent flows numbering up to approximately 128K well. However, as the number of concurrent flows increases above that threshold customers see a reduced benefit from this feature.

### Flow cache thrash:

This reduced benefit is in part due to flow cache thrash. Flow cache thrash occurs when flows are inserted and then evicted before the flow has been fully offloaded due to flow collisions. A flow collision occurs when a new unique flow hashes to the same flow cache location as an already (yet disparate) active flow. As flow concurrency increases, flow collisions also increase. As the flow cache fills up there is an increasing probability that a newly introduced flow will collide with an existing flow that occupies the flow cache location that the new flow hashes to. Thus, the fuller the flow cache, the greater the thrash.

When collisions occur, offloaded flows can be removed from the flow cache in favor of new flows. As all flows are treated as having equal importance even a small or low priority flow could collide with a higher priority fatter flow.

In such a scenario, the smaller lower priority flow could replace the larger high priority flow in the flow cache. Thus, the larger flow would no longer be offloaded by the FPGA. This mutes the benefits that can be achieved with the Fast L4 feature in terms of CPU utilization, latency, throughput and jitter. This means that networks with flow concurrency approaching 256K are not able to get the full benefits of Fast L4 Acceleration.

### Prioritized flows:

For users with flows approaching 256k – which might be large enterprises or service providers - the ability to prioritize flows being accelerated by Fast L4 Acceleration can overcome this limitation. This allows Big-IP to dynamically assess the size of the flow being transacted and adjusts the priority of fatter flows higher while pushing the priority of smaller flows lower.

As a result, prioritized Fast L4 Acceleration would give preferential treatment to fatter flows ensuring that the bigger the flow, the more likely it is to benefit from Fast L4 Acceleration.

Prioritized Fast L4 Acceleration essentially reserves precious flow cache space for those flows which benefit the most from the offload. The larger the object offloaded by a Fast L4 flow, the greater the benefits of the offload. Compute intensive, large object flows such as L4, CGNAT and QUIC traffic is identified and offloaded to the coprocessor, leaving short lived flows like DNS requests to the CPU.

In addition, once a fat flow gets into the flow cache, it will stay there. It won't be bumped out by smaller, low-priority flows.

Security applications is an area in which prioritized Fast Layer 4 Acceleration can provide benefits. Longer lived sessions with heavy

processing requirements such as encrypted traffic can gain significantly by being offloaded. In one specific use case, offloading QUIC traffic is an area where the customer will benefit.

Service providers will also see that increasing the efficiency of processing via the FPGA and the CPU reduces the number of retransmissions, reducing radio frequency traffic, another important benefit.

The Smart Coprocessor feature (prioritized Fast L4 Acceleration) is available on all iSeries i5000 appliances and above and on the VIPRION B2250 and B4450 with BIG-IP TMOS 15.0.

<b>Smart Coprocessor Functionality</b>	<b>Available</b>
<b>i15000 series</b>	✓
<b>i11000 series</b>	✓
<b>i10000 series</b>	✓
<b>i7000 series</b>	✓
<b>i5000 series</b>	✓
<b>i4000 series</b>	✗
<b>i2000 series</b>	✗
<b>VIPRION B4450 blade</b>	✓
<b>VIPRION B2250 blade</b>	✓

