

TCP Offload in Servers Relieves Network traffic Bottlenecks

Delivering 20G Line rate Performance with
Ultra low latency

A TOE Story



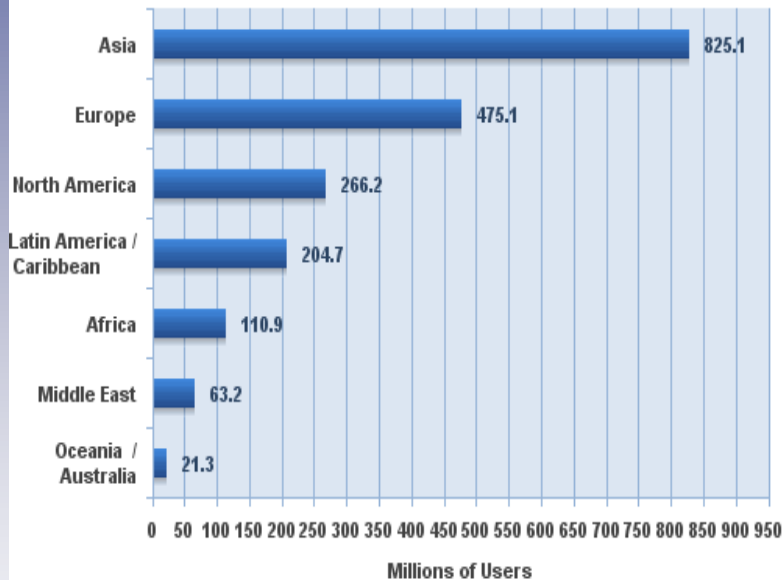
Santa Clara, CA
November 2012

*intilop Corporation
4800 Great America Pkwy. Ste-231
Santa Clara, CA. 95054
Ph: 408-496-0333, Fax: 408-496-0444
www.intilop.com*

Content	Slide
- Network Traffic growth	3
- TCP/IP in Networks	5
- TCP Offload VS Non Offload	6
- Why TCP/IP Software is too slow?	8
- Market Opportunity	12
- Required Protocol Layers.....	13
- Why In FPGA?	15
- Intelop's TOE – Key Features	18
- 10 G bit TOE – Architecture	21

Network Traffic Growth

**Internet Users in the World
by Geographic Regions - 2010**



Source: Internet World Stats - www.internetworldstats.com/stats.htm
 Estimated Internet users are 1,966,514,816 on June 31, 2010
 Copyright © 2010, Miniwatts Marketing Group

Global IP traffic will increase fivefold by 2015:

Global IP traffic is expected to increase fivefold from 2011 to 2016, approaching 70 exabytes per month in 2015, up from approximately 11 exabytes per month in 2010.

By 2016, annual global IP traffic will reach about 0.7 of a zettabyte. IP traffic in North America will reach 13 exabytes per month by 2015, slightly ahead of Western Europe, which will reach 12.5 exabytes per month, and behind Asia Pacific (AsiaPac), where IP traffic will reach 21 exabytes per month

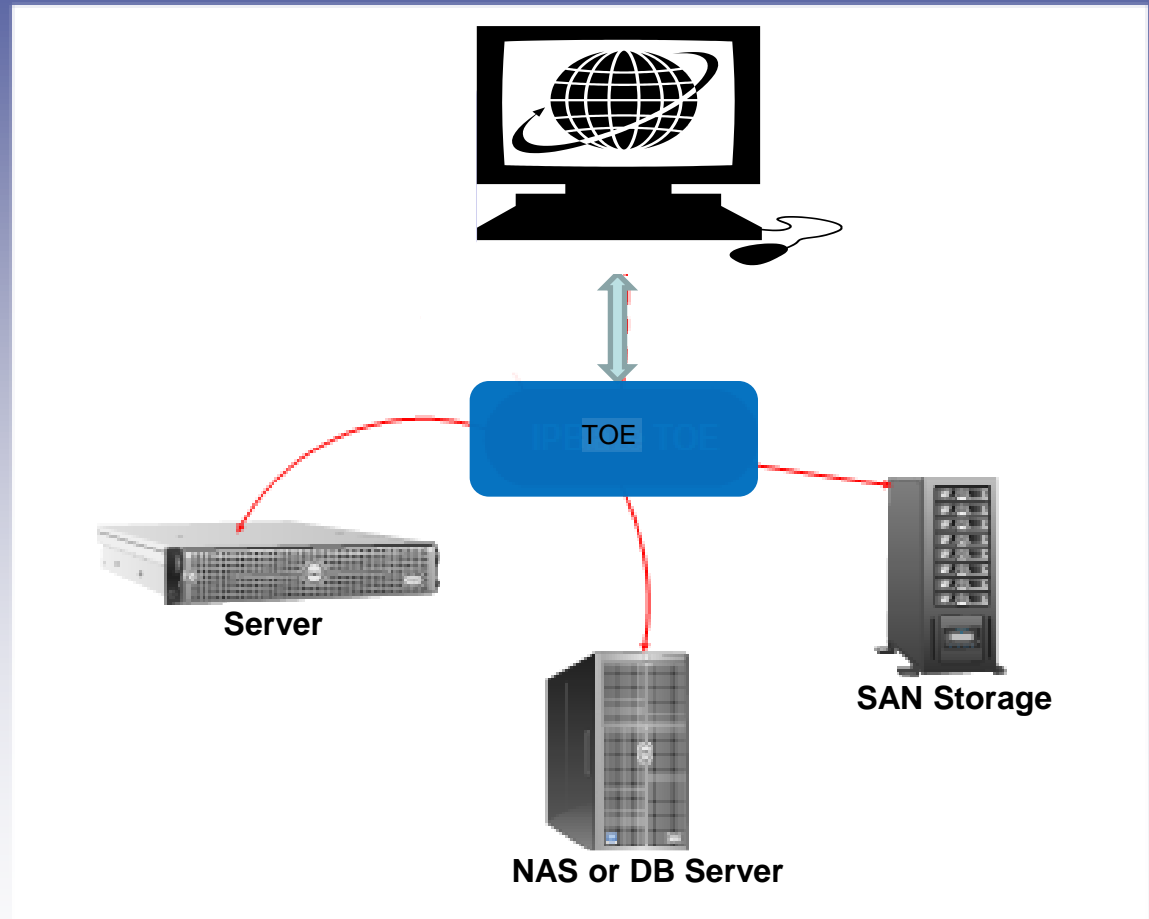
Middle East and Africa will grow the fastest, with a compound annual growth rate of 51 percent, reaching 1 exabyte per month in 2016.

An optimized TCP stack running on a Xeon Class CPU – @2.x GHz when dedicated to just one Ethernet port can handle data rate of up to about 500 MB/s before slowing down.

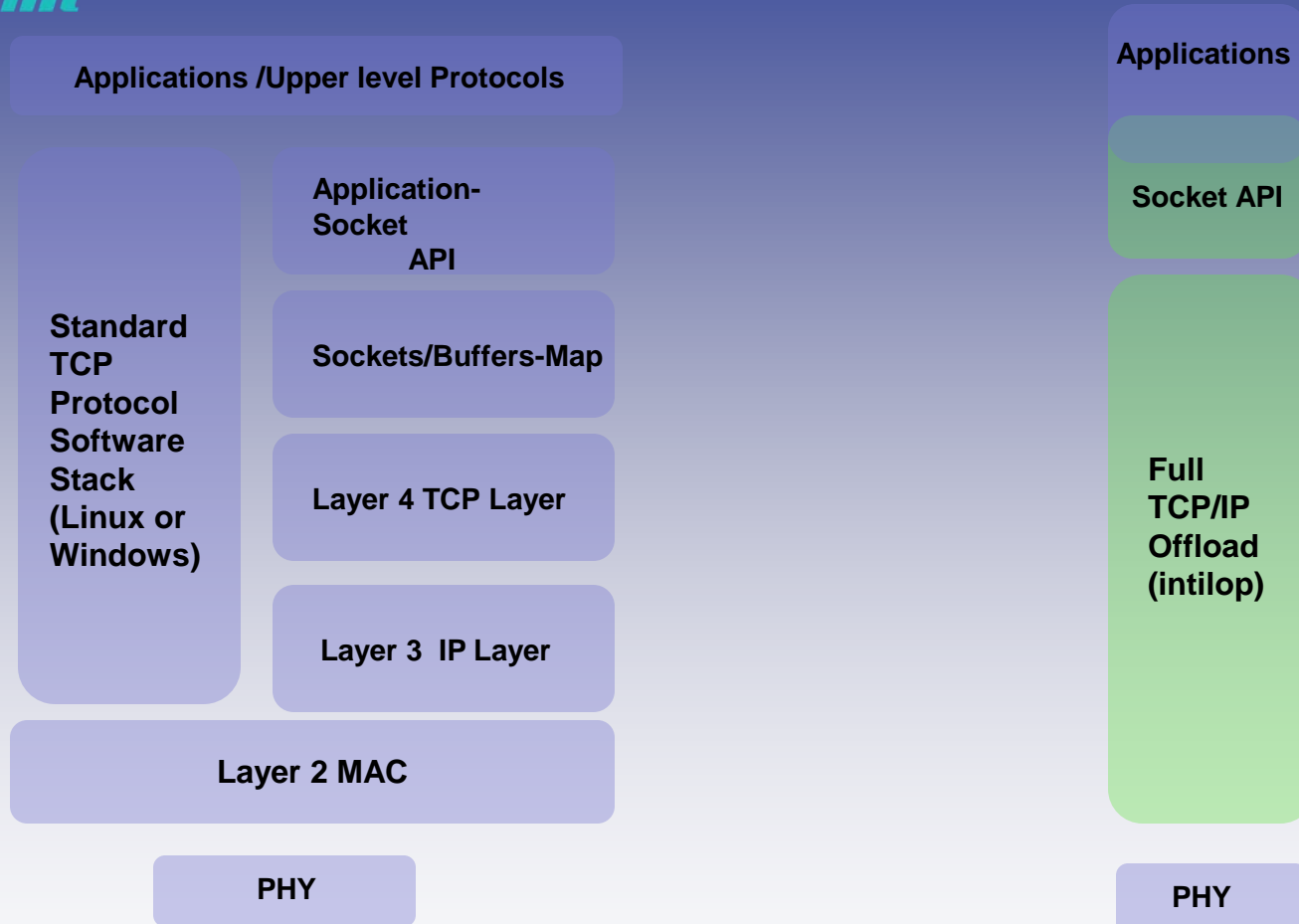
Terabyte (TB)	10^{12}	2^{40}
Petabyte (PB)	10^{15}	2^{50}
exabyte (EB)	10^{18}	2^{60}
Zettabyte (ZB)	10^{21}	2^{70}
Yottabyte (YB)	10^{24}	2^{80}

TCP/IP in Networks & Challenges

- Increasing TCP/IP performance has been a major research area for the networking system designers.
- Many incremental improvements, such as TCP checksum offload have since become widely adopted.
- However, these improvements only serve to keep the problem from getting worse over time, as they do not solve the network scalability problem caused by increasing disparity of improvement of CPU speed, memory bandwidth, memory latency and network bandwidth.
- At multigigabit data rates, TCP/IP processing is still a major source of system overhead.



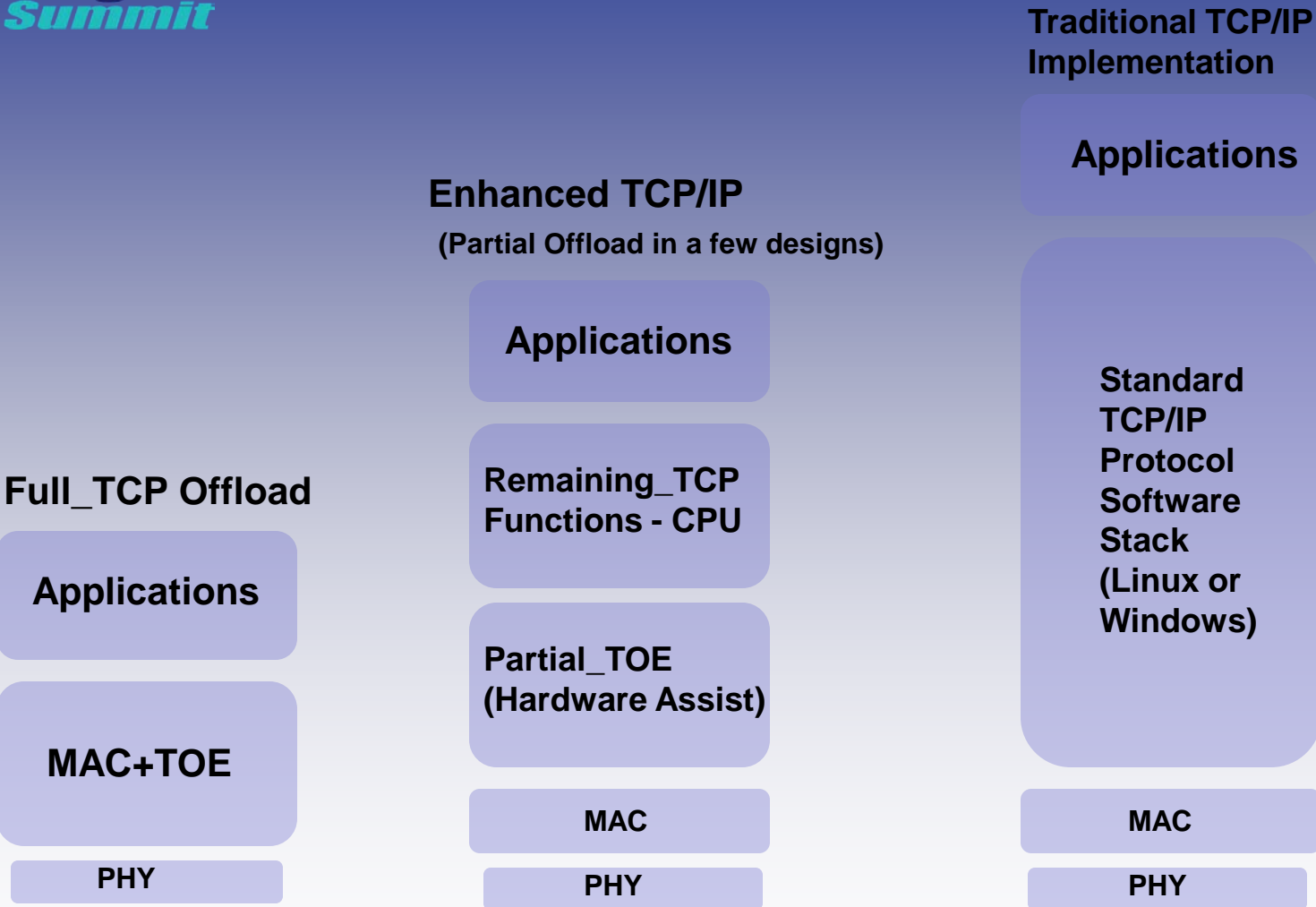
TCP Offload VS Non Offload



Current TCP/IP Software Architecture

TCP Offload Architecture

Various degrees of TCP Offload



Why TCP/IP Software is too slow?

Traditional methods to reduce TCP/IP overhead offer limited gains:

- After an application sends data across a network, several data movement and protocol-processing steps occur. These and other TCP activities consume critical host resources:
- The application writes the transmit data to the TCP/IP sockets interface for transmission in payload sizes ranging from 4 KB to 64 KB.
- The OS segments the data into maximum transmission unit (MTU)–size packets, and then adds TCP/IP header information to each packet.

Why TCP/IP Software is Slow

- The OS copies the data onto the network interface card 's(NIC) send queue.
- The NIC performs the direct memory access (DMA) transfer of each data packet from the TCP buffer space to the NIC, and interrupts CPU to indicate completion of the transfer.
- The two most popular methods to reduce the substantial CPU overhead that TCP/IP processing incurs are TCP/IP checksum offload and large send offload.

Why TCP/IP Software is Slow?

- TCP/IP checksum offload:**

- Offloads calculation of Checksum function to hardware.
- Resulting in speeding up by 8-15%

- Large send offload (LSO) or TCP segmentation offload (TSO):**

Relieves the OS from the task of segmenting the application's transmit data into MTU-size chunks. Using LSO, TCP can transmit a chunk of data larger than the MTU size to the network adapter. The adapter driver then divides the data into MTU-size chunks and uses an early copy TCP and IP headers of the send buffer to create TCP/IP headers for each packet in preparation for transmission.

Why TCP/IP Software is too Slow?

- CPU interrupt processing:**

An application that generates a write to a remote host over a network produces a series of interrupts to segment the data into packets and process the incoming acknowledgments. Handling each interrupt creates a significant amount of context switching

- All these tasks end up taking 10s of thousands of lines of code.

- There are optimized versions of TCP/IP software running which achieve 10-50% performance improvement;

- Question is: Is that enough?**

Market Opportunity

Accelerated Financial Transactions, deep packet inspection and storage data processing requires ultra-fast, highly intelligent information processing and Storage/retrieval-

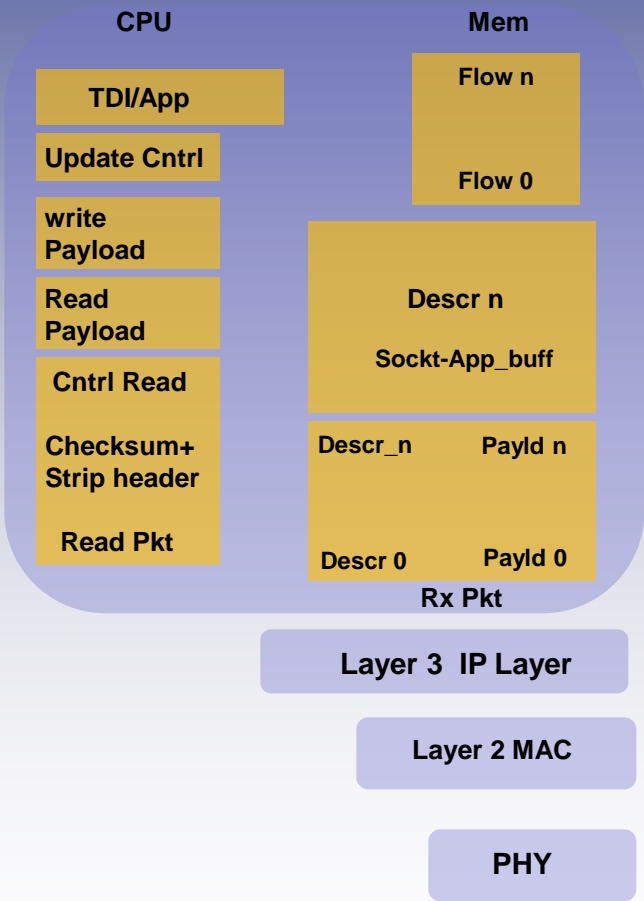
Problem:

The critical functions and elements that are traditionally performed by software and can not meet today's performance requirements.

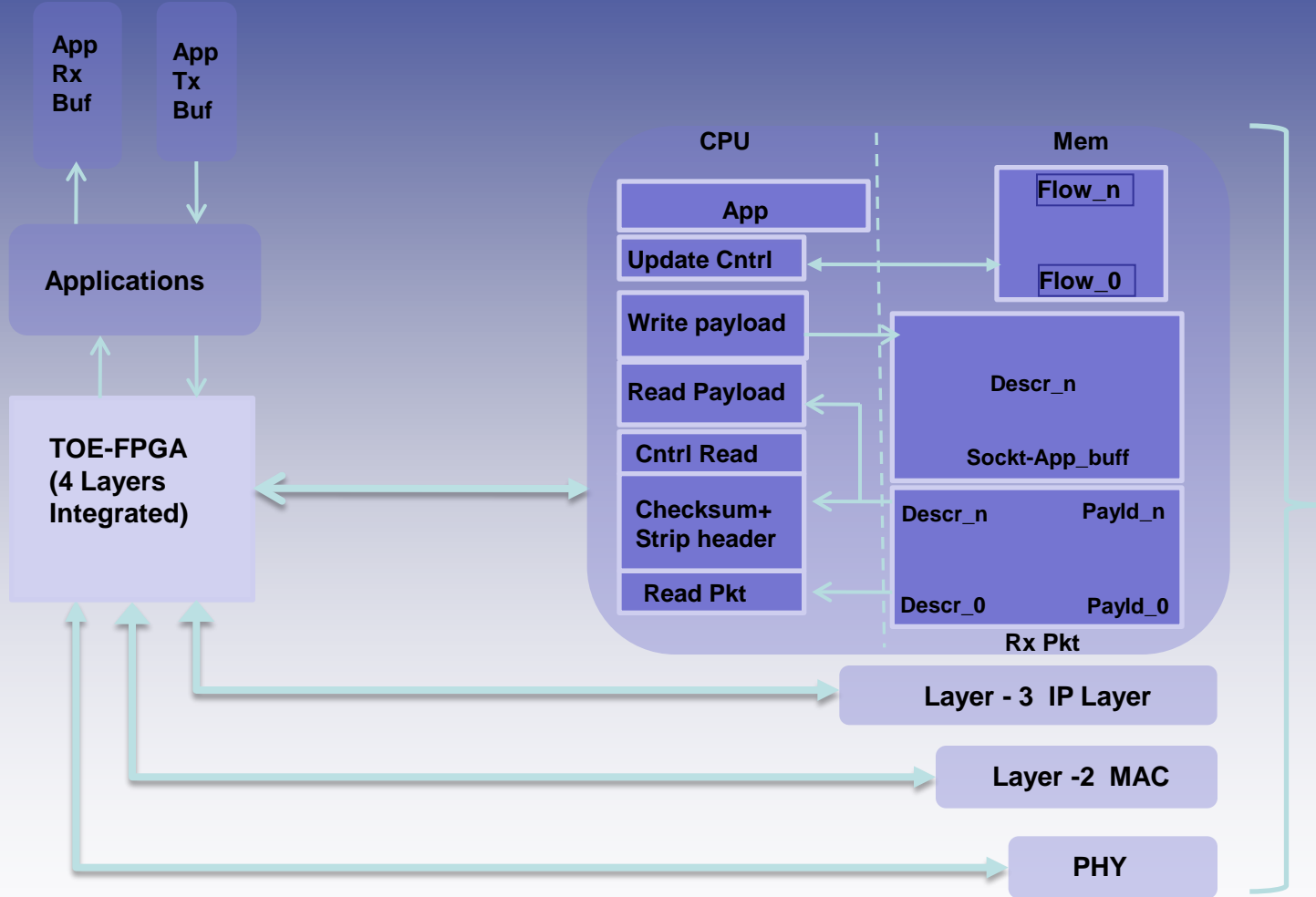
Response:

-We developed the critical building blocks in 'Pure-Hardware creating Mega IPs' to perform these tasks with lightning speed.

Required Protocol Layers



TCP/IP protocol hardware implementation



Why In FPGA?

•Flexibility of Technology and Architecture-

- By design, FPGA technology is much more conducive and adaptive to innovative ideas and implementation of them in hardware
- Allows you to easily carve up the localized memory utilization in sizes varying from 640 bits to 144K bits based upon dynamic needs of the number of sessions and performance desired that is based on FPGA's Slices/ALE/LUT + blk RAM availability.
- Availability of existing mature and standard hard IPcores makes it possible to easily integrate them and build the whole system that is at the cutting edge of technology.

- **Speed and ease of development**

- A typical design mod/bug fix can be done in a few hours vs several months in ASIC flow
- Most tools used to design with FPGAs are available much more readily, are inexpensive and are easy to use than ASIC design tools.
- FPGAs have become a defacto standard to start development with
- Much more cost effective to develop

- **Spec changes**

- TCP spec updates/RFC updates are easily adaptable.
- Design Spec changes are implemented more easily

- **Future enhancements**

- Addition of features, Improvements in code for higher throughput/lower latency, upgrading to 40G/100G are much easier.
- Next generation products can be introduced much faster and cheaper

Intilop's TOE – Key Features

•Scalability and Design Flexibility

- The architecture can be scaled up to 40G MAC+TOE
- Scalability of internal FIFO/Mem from 64 bytes to 16K bytes that can be allocated on a per session basis and to accommodate very 'Large Send' data for even higher throughput.
- Implements an optimized and simplified 'Data Streaming interface'
(No INTRs, Asynchronous communication between User and TOE)
- Asynchronous User interface that can run over a range of Clk speeds for flexibility.
- Gives user the ability to target to slower and cheaper FPGA devices.

Intilop's TOE – Key Features

- **Easy hardware and Software integration;**

- Standard FIFO interface with User hardware for Payload.
- Standard Embedded CPU interface for control
- Easy integration in Linux/Windows. Runs in 'Kernel_bypass' mode in 'user_space'

- **Performance Advantage**

- **Line rate TCP performance.**

- Delivers 97% of theoretical network bandwidth and 100% of TCP bandwidth. Much better utilization of existing pipe's capacities
- No need to do "Load balancing" in switch ports resulting in reduced number of 'Switch Ports' and number of Servers/ports.
- Latency for TCP Offload; ~100 ns. Compared to 50 us for CPU.

- **Patented Search Engine Technology being utilized in critical areas of TOE design to obtain fastest results.**

10 G bit TOE - Key Features

- Complete Control and Data plane processing of TCP/IP sessions in hardware → accelerates by 5 x – 10 x
- TCP Offload Engine- 20G b/s (full duplex) performance
- Scalable to 80 G b/s
- 1-256 Sessions, depending upon on-chip memory availability
- TCP + IP check sum- hardware
- Session setup, teardown and payload transfer done by hardware. No CPU involvement.
- Integrated 10 G bit Ethernet MAC.
- Xilinx or Altera CPU interfaces
- Out of sequence packet detection/storage/Reassembly(opt)
- MAC Address search logic/filter (opt)
- Accelerate security processing, Storage Networking- TCP
- DDMA- Data placement in Applications buffer
 - > reduces CPU utilization by 95 +%
- Future Proof- Flexible implementation of TCP Offload
- Accommodates future Specifications changes.
- Customizable. Netlist, Encrypted Source code- Verilog,
- Verilog Models, Perl models, Verification suite.
- Available ; Now

•These main building block IPs that are integral components for Network Security engine that performs deep packet inspection of network traffic at multi G bit line rate, sustained, full duplex.

1. ***10 G bit TCP Offload Engine***
2. ***1 G bit TCP Offload engine***
3. ***10-G Ethernet MAC***
4. ***1-G bit Ethernet MAC***
5. ***4-16 Port Layer 2 switch with IEEE-1588. 1 G bit per port.***
6. ***Deep-Packet Content Inspection Engine***

THANK YOU