

# **Integration of ASIC-based Wire-speed Multi-service Traffic Management Into Next-Generation Networking Architectures**

Written Exclusively for *Integrated Communications Design*

By

Bidyut Parruck, CEO  
Azanda Network Devices

As optical networking moves into its next evolutionary phase, a number of rapidly converging factors are driving dramatic changes in the underlying semiconductor-level technologies that provide the foundation for next-generation system architectures. One of the critical arenas that must be addressed is ensuring unrestricted scalability of the network processing, traffic management and switching functions. Working in balance with one another, these key functions must simultaneously keep up with higher speeds, more channels and multiple service types for millions of individual traffic flows – all with sustained wire-speed performance at escalating optical bandwidths.

In earlier networking architectures operating at lower speeds and with fewer channels, these key network-processing functions could be implemented within the same physical device; starting with general-purpose processors and moving on to more specialized network processors. However, as the demands for faster wire-speed performance and greater multi-service traffic complexity have now kicked into overdrive, the reliance on monolithic all-in-one network processing alternatives can no longer keep pace with real-world requirements.

Forward-looking system architects are turning now to innovative approaches that more intelligently partition the processing pipeline by leveraging a new-generation of highly targeted ASICs to provide maximum speed, flexibility and scalability for traffic management functions. As will be explored in this article, the blending of packet processor/classification engines to handle higher level functions and programmable ASICs to handle wire-speed multi-service traffic management enables the creation of new dynamically adaptable intelligent line card designs. These ASIC-based line cards can seamlessly bring together high port densities and multi-services for simultaneously handling IP packet, ATM and SONET TDM traffic, while also providing inherent scalability for wire-speed performance at OC-192, OC-768 and OC-3072 levels.

## **Driving Forces**

Over the past few years, the undisputed performance advantages of fiber optics over traditional copper networking has enabled optics to completely pervade WAN transport applications as well as migrating rapidly through many MAN environments and even some targeted corporate LAN requirements. At the same time, optical bandwidths continue to escalate exponentially, from 2.5Gbps OC-48 to 10Gbps OC-192 and are now moving quickly on toward 40Gbps OC-768 and beyond. In addition, the widening use of Dense Wavelength Division Multiplexing (DWDM) is dramatically increasing the

number of channels that can be carried on each physical fiber, thereby also driving up the system level challenges for effectively terminating the additional DWDM channels and putting the increased bandwidth to effective use.

Due to the explosive growth of the Internet over the past few years, native IP packet traffic has now become the most prevalent networking protocol on the planet. However, the widespread deployment, usage and reliance upon legacy ATM, TDM and SONET technologies also means that they cannot simply be set aside in favor of universal use of IP packets. Instead, what is actually needed is an efficient bottom-up architecture that can seamlessly handle all traffic types, whether packet based, cell based or TDM, while intelligently providing cross-protocol traffic management for millions of simultaneous traffic flows.

### **Challenges of Wire-Speed Processing**

Although the term “wire-speed” gets quite a bit of attention in the industry today, thus far there has been no clear cut set of parameters and metrics that allow for a consistent agreement on what it means. In general, the term “wire-speed processing” evokes a common understanding that all of the internal processing is carried out in such a manner that traffic flowing from the ingress point to the egress point continuously moves at the same bandwidth as the fiber (“wire”) that the port is serving. However, depending upon the amount and type of processing that is included in the base assumptions, the resultant claim of wire-speed can mean very different things for real-world performance.

Although wire-speed processing ultimately has to be evaluated under the stress of real-world application demands, surrogate assessments can be derived by looking at all of the key elements in the network processing pipeline and projecting likely scenarios regarding the amount of processing cycles required for each element. The following are just some of the basic processing elements that must be considered.

#### Classification

Incoming data must be analyzed in order to determine the specific flow to which it belongs. Depending upon the depth of classification required, performing this step at wire-speed can require varying degrees of processing power. For example, simple IP look-ups or MPLS label checking based on header information can be performed much more quickly than deeper context-based switching algorithms that require analysis of each packet’s internal content.

#### Ingress Traffic Queuing/Switching

All incoming traffic must be queued for its movement into the switch fabric based upon specified criteria and policies. Because the queuing process involves appending specific switch-header information to each packet, conducting this step at wire-speed generally requires at least 25% more bandwidth than the ingress wire-speed. For example, traffic queuing operations for a 10Gbps link may actually need to run internally at sustained rates of 12.5Gbps or higher in order to deliver on the promise of wire-speed processing.

### Switch Fabric Transit

In the switch fabric, in order to allow for peak saturation and to handle back-pressure from multiple flows simultaneously contending for the same egress points, actual processing requirements can be 2X to 4X higher than the rated wire-speed bandwidth. For supporting 10Gbps speeds, this can typically mean needing 20-40Gbps of aggregate switch fabric bandwidth to guarantee wire-speed performance. If the switch fabric is not optimally designed for non-blocking throughput, back-pressure from egress points can push congestion all the way back to the ingress points. In addition, both the ingress and the egress traffic managers need to be able to keep up with the overall switch fabric speeds.

### Traffic Shaping/Scheduling/Queuing for Egress

Traffic management functions at the egress point must be able to efficiently handle the high-speed flows from the switch fabric and perform traffic shaping/queuing operations without creating additional back-pressure. Depending upon the specific application, these egress functions can include complex handling of a variety of queuing schemes for traffic shaping, Quality of Service (QoS) management, etc., which can require a significant degree of extra processing headroom. For example for weighed-fair-queuing mechanisms, the traffic manager must constantly maintain cognizance of all queues to ensure that even lower-priority traffic is not blocked when there are empty queues available. Staying abreast of many separate queues and up to a million individual traffic flows requires a blending of both ultra high performance hardware and intelligent processing schemes that avoid unnecessarily wasting clock cycles.

Although each of the above processing operations can be individually very difficult at speeds of 10Gbps and above, effectively putting them all together takes the challenge to a whole new level. When considering the architectural tradeoffs, it is critical for system designers to keep in mind that their end customers invariably will be running a wide variety of diverse service mixes and traffic types with unpredictable peaks and spikes in both traffic levels and processing demands. In order to deliver sustained wire-speed processing, it is vital that each function be able to scale to full performance independently of the processing demands at other points within the pipeline. For example, in a monolithic network processor that has to handle all functions, wire-speed traffic management can become compromised if the need to spend too many cycles on deep packet classification tasks temporarily bogs down the core processor.

To put the magnitude of the challenges into perspective, consider that at OC-192 10Gbps speeds a true wire-speed process only has 30ns to perform any operation. Any slower and the processor falls behind the wire-speed objective. For traditional data/instruction oriented processors, this becomes even more challenging because every time a branch is encountered the processor's flow is disrupted. Since most network traffic management operations are inherently branch-oriented, the only way that conventional processors can keep up is through brute force replication of the processing cores. To overcome the practical limitations of even the most advanced semiconductor processes, network processor architectures have had to go to multiple cores, with as many as 16 cores needed in some architectures just to get to OC-48 speeds. As wire-speed requirements increase,

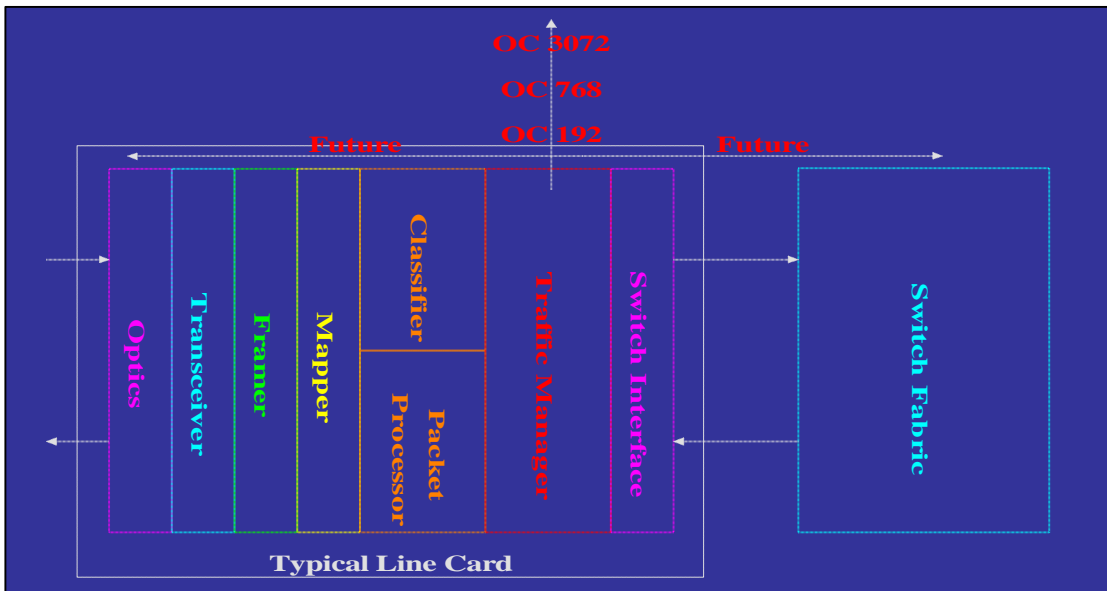
these multi-core all-inclusive processing architectures simply cannot scale to keep up without proliferating so many processors that the software programming challenges become nearly insurmountable.

The complex tasks involved with multi-layer packet classification, analysis, traffic shaping, policing, QoS policy management, etc. have already become more than can be effectively handled in software as network speeds push beyond OC-48 levels. Not only are the required software structures exponentially increasing in complexity and making it virtually impossible for development to keep pace with market requirements. At the same time, the sheer complexity of the code makes it nearly impossible to provide required levels of predictability and deterministic performance needed at OC-192 speeds and beyond. In effect, with these multi-core software-intensive architectures, the system designers' ability to create efficient code becomes a too significant and uncontrollable variable in the system's ultimate performance capabilities.

### Partitioning the Processing Pipeline with ASIC-based Traffic Managers

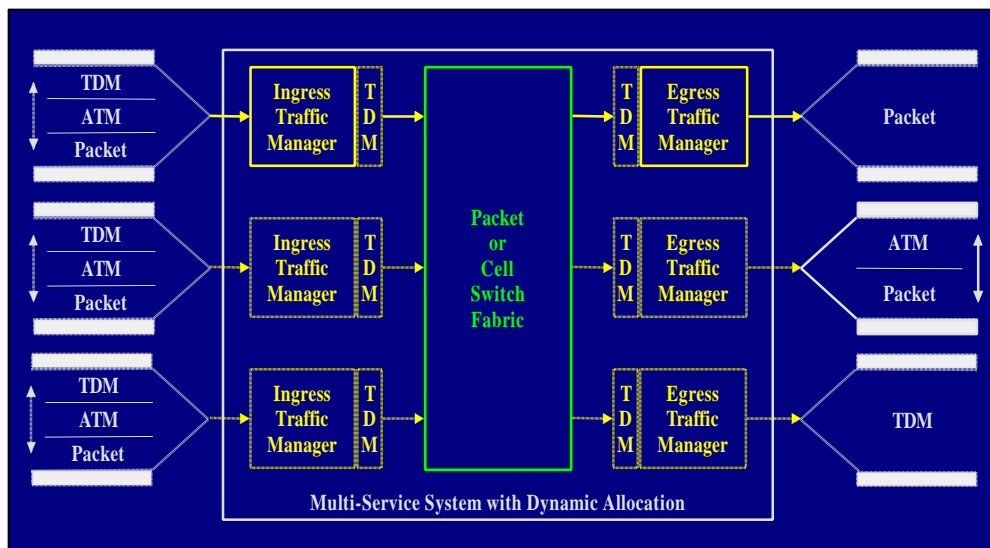
In order to support escalating optical bandwidths and wire-speed traffic management, next-generation system architectures must partition the traditional network processor functions into a programmable packet processor and a separate hardware-optimized traffic manager. Essentially, by decoupling low-level, on-the-fly traffic management functions, such as shaping, policing and scheduling, from the higher level packet processing and classification tasks, these next-generation line card architectures will be able to efficiently scale up through OC-768 and ultimately to OC-3072 levels.

#### Next-generation Line Card Partitioning



This partitioning of functionality also allows for a greater degree of implementation flexibility at the system level because the packet processor/classification functions now become optional pieces that can be omitted to streamline the pipeline for specific applications. For example, in a network using MultiProtocol Label Switching (MPLS), Label Edge Router systems that must aggregate, classify and assign labels to the traffic need to have a different level of functionality from the Label Switch Router systems that only have to read, process and forward MPLS labeled traffic. For the LSR systems in the core, the processing pipelines can be significantly simplified by eliminating the packet processor/classifier functions and simply programming the traffic manager ASICs to handle all of the MPLS processing and forwarding.

In addition, by optimizing the traffic management function for dynamically handling multiple traffic types, such as packet, ATM or TDM flows, the use of fast efficient ASIC hardware can enable next-generation systems to seamlessly route any ingress to any egress, without the unnecessary overhead of higher-level software processing.



**Any Ingress to Any Egress**

The targeted implementation of traffic management functions in ASIC processors also enables each key segment to evolve and scale independently, thereby improving the smooth forward extensibility of overall system architectures. For example, as new high level network processing parameters change, the software-based packet processor/classifier devices can be reprogrammed or revised without impacting the stability of the low-level wire-speed traffic management functions. On the other hand, the processor-intensive and well-proven traffic management algorithms, such as weighed fair queuing, can be optimized for very high speed without worrying about the next market-driven changes to higher level packet classification requirements.

## Critical Design Issues

The sections below touch briefly on just a few of the key semiconductor-level and module-level design issues that must be addressed in order to optimize next-generation network processing pipelines based around programmable ASIC traffic management devices.

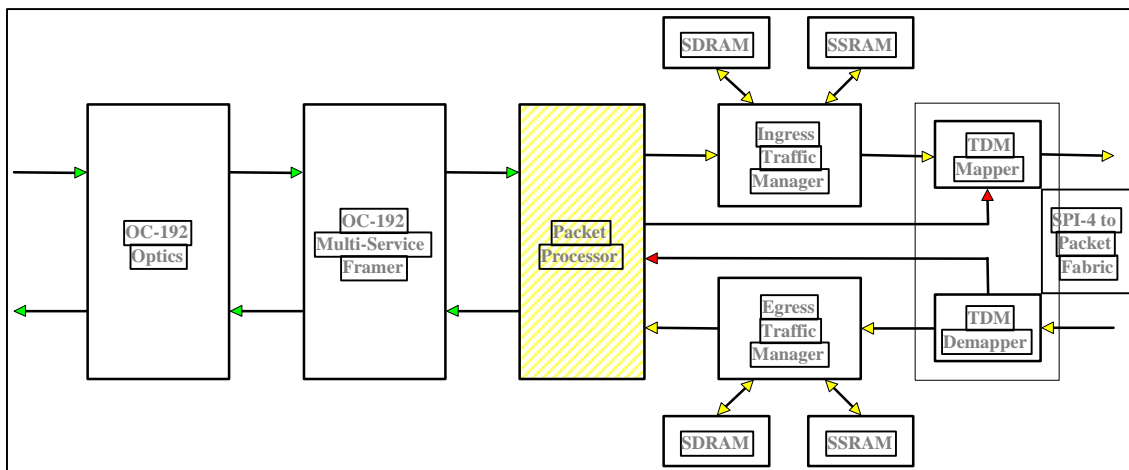
### Optimizing I/O Interfaces

In order to maintain optimal non-blocking data flow, the design of inter-function I/O is critical at each interface point along the processing pipeline. For instance, the use of an advanced protocol such as SPI-4 Phase 2 for interfacing processors to the Framer can greatly improve data flow by eliminating inter-processor polling and utilizing more cycles for actually moving data. Similarly, as described below, the use of DDR interfaces to external SRAM can double effective memory bandwidths, thereby boosting throughput for the many memory-intensive functions associated with traffic management, scheduling, queuing, etc.

### Integrating Multi-faceted Memory Structures

Although conventional memory bandwidth limitations were not a significant factor at OC-48 speeds, they have become clear concerns at OC-192 and represent insurmountable roadblocks at OC-768. At speeds of 10Gbps and higher a single DRAM bank simply cannot support fast enough storage and retrieval of data at wire-speed rates.

Advanced ASIC-based traffic management devices must be designed around an optimized memory strategy that leverages efficient simultaneous use of multiple on-chip and off-chip memory structures. For example, as shown in the figure below, dedicating separate multiple banks of DDR SRAM for ingress flows and for egress flows can efficiently provide for non-blocking high-bandwidth memory structures, which are independently accessed by the ASICs to manage many different queues at wire speeds without any inter-queue congestion.



### Using Efficient Data Structures

Another key consideration is the organization of the data structures themselves. In order to minimize the number of memory access cycles, the data structures should be arranged so that the ASIC can effectively anticipate required data in order to make multiple accesses from multiple locations within the same memory cycle. In addition, by anticipating certain data that will need to be accessed more frequently (such as MPLS label forwarding criteria) and storing it in small internal SRAM locations, the overall processing speed can be further optimized.

### **Tying It All Together**

The bottom line from a system architecture standpoint, is an enhanced ability to leverage each part of the network-processing pipeline for optimal performance and scalability. Rather than trying to shoe-horn everything into a single monolithic software-intensive multi-processor scheme that can't keep pace with escalating wire-speed requirements, the whole pipeline can be optimized through a complementary blend of high-level software-based packet processors and low-level speed-optimized programmable ASIC traffic managers.