WP244 (v1.0) April 10, 2006

Traffic Management in Xilinx FPGAs

By: Nick Possley

Traffic management is a required functionality in today's networking and telecommunications equipment as more services are delivered via packetized shared transport mediums. This type of delivery is more cost effective due to statistical multiplexing gain achieved. Traffic management is essential for maximizing the gain.

Traffic management controls access to a shared transport medium by arbitrating the available bandwidth in a number of ways for a defined level of granularity of packetized services.

Most designers need unique functionalities for their systems in addition to low cost and high performance. Xilinx FPGAs provide the best solution. A carefully designed Traffic Manager solution can be scaled and tailored to exactly match the needs of the customer in terms of logic; the customer only pays for the silicon needed. Hence, FPGAs provide the most cost-effective and high-performance solution in this market.

Background

In traditional Time Division Multiplexing (TDM) systems, traffic management is very simple. Every connection is assigned a time slot. There is no contention or arbitration to manage. This is really a network provisioning exercise.

As networks have evolved, shared transport mediums (for example, Ethernet or ATM) have become the preferred network. Costs are much lower due to statistical multiplexing gain (for example, two phone conversations can share the same connection because people do not talk constantly). However, these shared medium transports are expected to provide the same level of service as old TDM services, when bandwidth and latency are "carved" out of the shared medium.

Level of service can be defined with the following aspects:

- Availability of the connection
- Latency of the connection
- Bandwidth of the connection

These three aspects of a connection affect how well a service is delivered. In general, the Traffic Manager plays less of a role in availability because availability is dependent upon the physical connections of the network. Routing and switching play a heavier role in availability.

Traffic Management should be done at a defined service level. Services contend for bandwidth. Examples of services are streaming video or web browsing. Other networks can define services at different layers, such as a LAN service for a company as opposed to a streaming video user sitting at home. These different layers of service can be combined (for example, a video user on a LAN service) to provide more efficient traffic management but might not be necessary because the layers of service can be combined at different locations in the network. The layer of the service affects the complexity of the Traffic Manager. Some only need to be simple while others are much more complex.

Bandwidth and latency allocation can be done in several ways. *Scheduling* which service sends a packet next and *shaping* how fast a service can send a packet in a given time period are two ways that work well together. The scheduler arbitrates between services and can give more bandwidth to one service or allow one service to preempt another providing low latency. Scheduling can be done using different algorithms. Some algorithms are better for managing the bandwidth than others but are more complex. This design trade-off must be made. The shaper controls the output bandwidth utilization of a given service in total, not with respect to other services.

Pathological cases exist where many services contending for a given transport are bursting well beyond their allocated/expected rate, creating congestion in the Traffic Manager. In these cases, *congestion management*, an important part of the Traffic Manager, is the only area where the Traffic Manager plays a role in availability. Congestion can be managed using different algorithms. Some algorithms allow more availability than others, depending on the service. Therefore, it is important to support the best congestion management algorithms.

One additional way of providing a level of service is to monitor the arrival rate of a given service and drop packets when the service exceeds a given rate. This functionality is called *policing*.

Queuing is required in a Traffic Manager because the packets are being collected as the output bandwidth is being scheduled. The packets must be stored in a queue while they are waiting to be sent.

Solid traffic management can be achieved when combining queuing, scheduling, shaping, congestion management, and policing. This combination results in optimal service delivery in a shared medium transport.

Applications

There are multitudes of applications for a Traffic Manager, from a simple port level statistical multiplexer to full delivery of "triple play" services.

Given that a simple statistical multiplexing application is fairly straight forward, this section focuses on Multimedia Services delivery, covering several different architectures in the metropolitan and access networks. Figure 1 shows a network example for Multimedia service delivery.

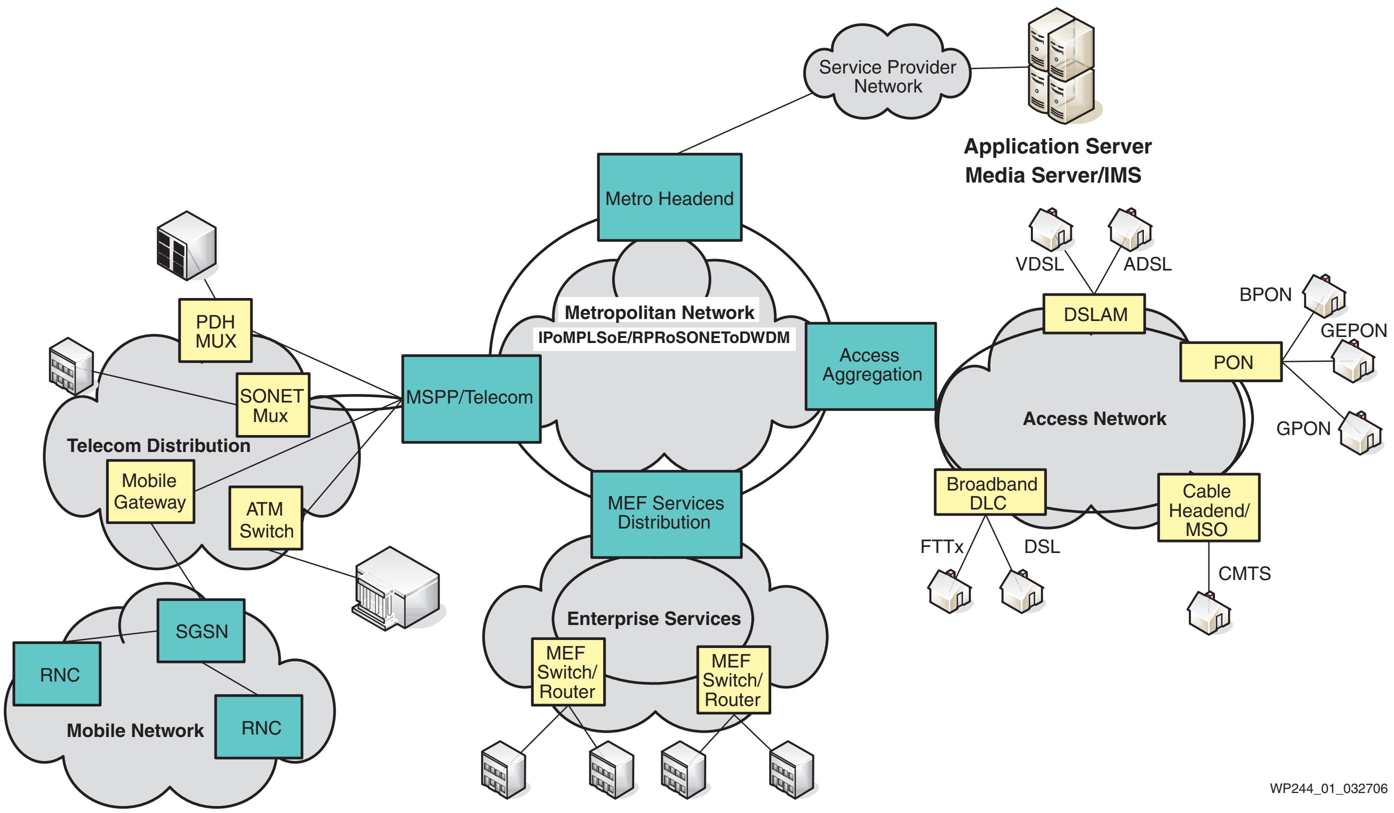

Figure 1: **Metropolitan Network Diagram**

The network diagram in Figure 1 shows the various types of service endpoints and networks that distribute content and provide connectivity services. Content distribution services are distributed from the service providers down to all of the endpoints. Connectivity services exist between endpoints, e.g., business-to-business LAN services. Most of the communications mediums are statistically multiplexed; some are TDM based. The central network is a statistically multiplexed network that can transport TDM over packets as well as packetized services. Packetized services include Voice, Video, and Data services. To ensure reliable delivery of TDM traffic and all services, traffic management must exist at every point in the network. The only case where traffic might not be needed is when there is a surplus of bandwidth such that there is no contention. This condition is normally temporary, and bandwidth needs to be efficiently utilized as the number of subscribers grows.

The granularity of traffic management can change depending on the location in the network. The metropolitan network, in some cases, does not directly deliver content distribution services to subscribers. In those cases, the services are not managed as per subscriber but as a service per distribution point. In the access network, the content is directly delivered to a subscriber, so each subscriber is individually managed. The application servers must manage content distribution to the service distribution points, which requires a fine level of granularity of traffic management on the server.

For connectivity services, the metropolitan network must manage the service per subscriber. This can be seen as a fine level of granularity, but the number of subscribers is typically not as large as the content distribution because the service is provided for medium to large businesses typically.

Access Traffic Managers

Access equipment delivers services directly to subscribers. Due to access transport medium constraints (for example, DSL), there are less than 1000 subscribers served by any one piece of equipment. As a result, the traffic management functions do not have to be large. In general, it is important to be able to deliver subscriber services within the service agreement limits. Therefore, the QoS must be controlled per subscriber and then per type of service per subscriber. A two-stage hierarchy can be used with traffic shaping per subscriber and per type of service.

IETF and IEEE define eight types or classes of service. Typically, only the following four classes are needed:

1. Voice can be classified as real-time traffic, requiring low latency.
2. Streaming video/audio is classified as non-real-time but still needs low latency.
3. Control traffic requires the highest priority and lowest latency, such as Internet Group Management Protocol (IGMP).
4. All other data traffic, such as gaming or FTP.

More classes can be defined, but the value is often lost to a specific subscriber. Other classes are useful to prioritize between subscribers. Figure 2 illustrates the traffic management functions.

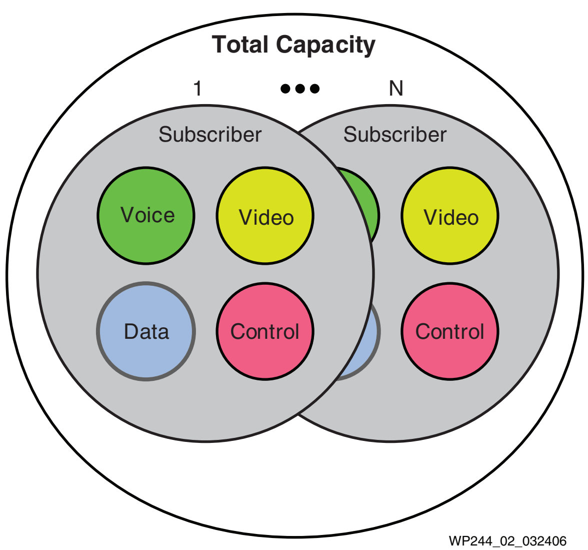

Figure 2: **Access Traffic Management**

To enforce the quality of services, the traffic must be classified to a given subscriber and priority, shaped, and then statistically multiplexed according to the priority and subscriber. The packets must be parsed and compared to a list to determine the subscriber and class. The priority list in descending order is:

1. Control (highest)
2. Voice
3. Video
4. Data (lowest)

The different classes of service must be shaped. Shaping entails limiting the bandwidth within a specified time period. In some cases, the bandwidth can be limited in two different time periods to provide better control in bursty traffic situations. Control has the highest priority with the least bandwidth required. Therefore, it is important to shape the control traffic to a low rate so it cannot consume bandwidth from the other classes. Voice has the next highest priority and also has a fairly low bandwidth requirement. Video has a higher bandwidth requirement, and Data gets whatever is left over from the others (the *best effort*). Assuming that the total bandwidth usage from the other three classes is not constant and does not, in the worst-case, consume more bandwidth than the total, there always is some bandwidth for Data. In most cases, a considerable percentage of bandwidth can be used for Data.

Shaping for the subscribers is only required to provide a defined level of service and/or restrict the bandwidth to a port rate. In DSL, the port has a set defined rate and a shaper can be used to ensure that the packets are never sent beyond that rate. In a cable multimedia services operator (MSO) application, there is a shared bandwidth medium for all subscribers. One subscriber might pay for a 3 Mb/s service while another pays for a 1 Mb/s service. The shaper limits the bandwidth to the rate that the subscriber is paying.

The statistical multiplexing of the classes for a given subscriber is implemented by a scheduler. On a packet-by-packet basis, a scheduler looks across all packets, waiting to act after the shaper has allowed them to pass. Control packets are automatically sent ahead of all other packets. The rest of the priorities are treated the same.

Statistically multiplexing the subscribers is a little different from the classes. The concept of fairness is important in this case because bandwidth is being shared between equal priorities. The bandwidth needs to be shared fairly among the subscribers, requiring the scheduler to consider the packet length in the decision to send the packet. Subscribers with many large packets should send a packet less often than a subscriber with many small packets. In some cases, it is required to give more bandwidth to one subscriber over another that is proportional to the overall used bandwidth. These cases require the scheduler to provide more weight to one subscriber over another, for example, sending two packets for one subscriber and one packet for another subscriber in the same amount of time. This leads to the requirement for weighted fair bandwidth sharing. In addition, it is important to give any extra bandwidth to best effort classes when fewer subscribers are utilizing their set bandwidths. As a result, the scheduler should only service those subscribers with packets that are ready to send and not waste any resources/bandwidth on the others.

Metro Traffic Managers

In the metropolitan network in an access aggregation application, the number of subscribers is large, possibly in the 256K to 1M range. It is a burden for the metro equipment to track and enforce the subscriber services. In addition, the metro equipment might not actually know the number of subscribers because there might be two separate networks. As a result, the metro equipment is only concerned with enforcing services bundled for a particular access node. Figure 3 illustrates this case.

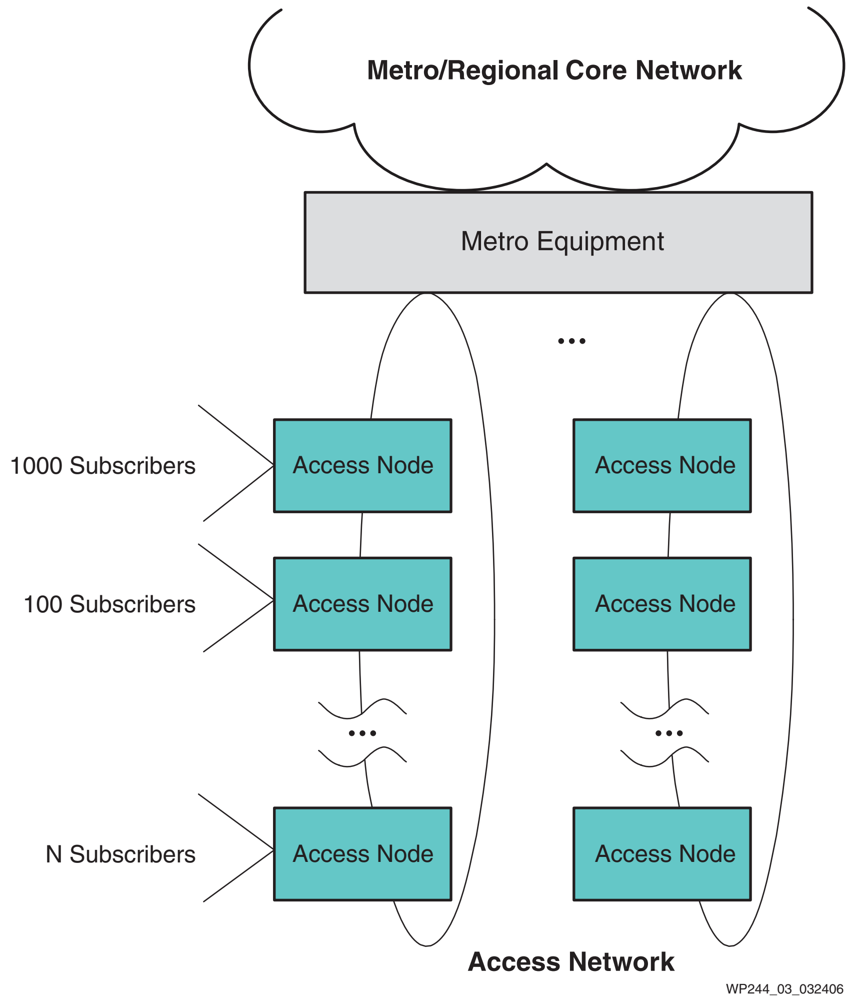

Figure 3: **Access Aggregation/Distribution**

As shown in Figure 3, many subscribers can be serviced by one metro node. The metro equipment must allocate bandwidth to each of the access nodes. Different amounts of bandwidth are allocated based on the number of subscribers at each of the nodes. Some consideration must be made for oversubscription. Typically, not all subscribers are active simultaneously.

Other issues must also be managed. For instance, video distribution could consume a large amount of bandwidth if the multicasting of channels per subscriber is performed by the metro equipment. Video should be distributed to all of the access nodes as multicast traffic. Each access node can choose to receive the traffic or not. The access nodes then, in turn, control the access of the channels to the end subscribers.

In the downstream direction, the Traffic Manager in the metro equipment can exist per subtended network. The Traffic Manager shapes traffic per access node per Class of

Service and schedule traffic per access node per class of service. The access nodes are all statistically multiplexed together.

In the upstream direction, the Traffic Manager receives all of the traffic per access node and allocates bandwidth on the metropolitan network per access node per class of service. In this case, bandwidth can be controlled by a fixed amount per node or can be allocated based on percentage. The service provider, with a Traffic Manager, can choose either way.

The rates for traffic management in the metro equipment have to be higher to accommodate all of the subtending equipment, but the granularity of traffic management is less than in access networks. Therefore, fewer queues, shapers, and so forth are required.

Mobile Networks

In mobile networks, bandwidth is very precious. In fact, there are government regulations for a minimum required efficient use of spectrum in wireless communications, making traffic management even more vital as more services are added to wireless distribution networks.

In wireless networks, Traffic Managers need to re-allocate unused bandwidth to subscribers as well as enforce Quality of Service to provide the required performance. In addition, the Traffic Manager must control congestion in the most efficient manner.

Both the 3GPP High-Speed Shared Downlink Packet Access (HSDPA) and the IEEE WiMAX can supply data services over-the-air. The transport medium is still shared, but differently from wired networks. Wireless networks are a “sea” of channels in which one channel can only be assigned to any given subscriber. Channel assignments are updated on periodic intervals to “share” the bandwidth among multiple subscribers. This operation calls for a different type of scheduling algorithm, because the result of a scheduling decision is a list of channel assignments. Because the quality of the channel to a given user must be considered in the scheduling decision, the Traffic Manager must have communication with the baseband functions. Therefore, the Traffic Manager function being co-located with the baseband functions provides for an efficient implementation.

The channel assignment provides the base data connection to the subscriber. There can also be multiple classes of service per subscriber, and the Traffic Manager can provide the scheduling mechanism into the base connection to enforce the quality of service for the various classes of service per subscriber.

Figure 4 illustrates the system configuration.

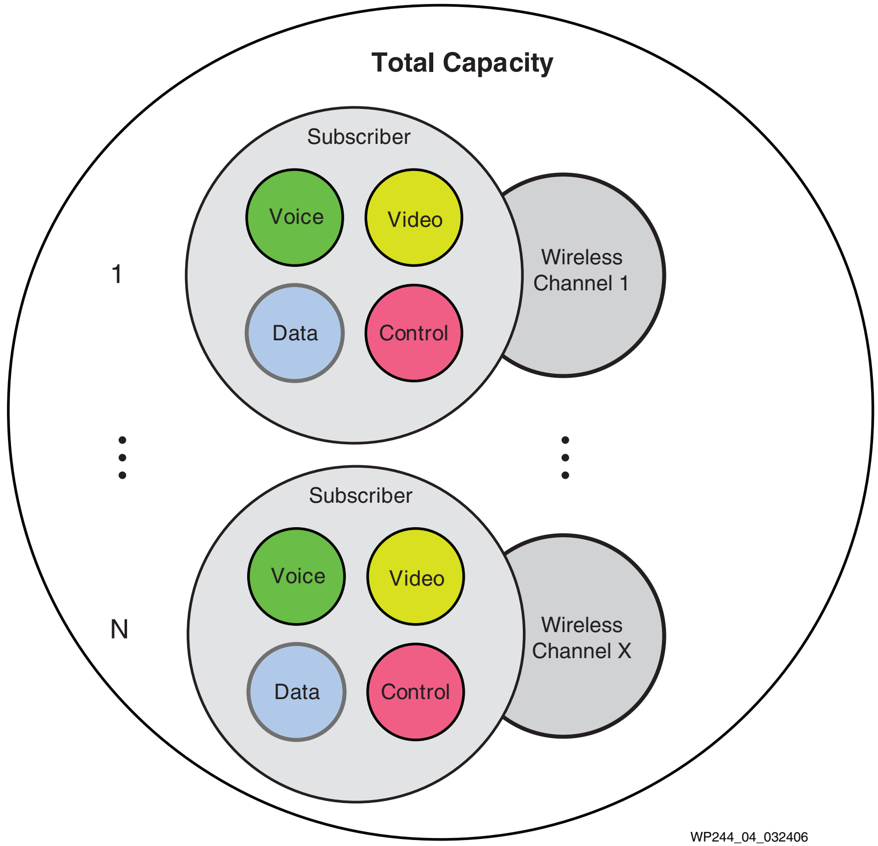

Figure 4: **Traffic Management in Mobile Networks**

Figure 4 shows the subscribers all contending for wireless channels. More than one channel can be assigned to a given subscriber. The total available capacity changes in time as the channel qualities change across all of the subscribers.

Congestion is a problem for this type of network when there are more subscribers than channels. The amount of bandwidth that can be provided to each subscriber is a function of the rate at which channels can be assigned. The faster channels can be reassigned, the more bandwidth each subscriber can receive. Performing this kind of scheduling algorithm in a processor-based implementation will, most likely, result in poor service distribution, because some assignments can take milliseconds. A dedicated hardware-based approach can provide for channel reassignment in the nanosecond to microsecond range depending on the system scenario.

Traffic Manager

Based on the "Applications" section, a Traffic Manager requires a number of functions and scaling factors to work in a variety of applications. From a high-level perspective, a Traffic Manager must shape packet-based traffic streams, arbitrate access to shared media bandwidth on a packet granularity, and control congestion during periods of bandwidth oversubscription.

These functions can be implemented in many ways. For a Xilinx FPGA, the Traffic Manager can be based on a set of hardware functional blocks that can be scaled and added or deleted to conform to the application requirements. Other implementations based on processors can also work; however, due to the Von Neumann architecture of most processors, there can be an additional penalty for loading instructions. An

FPGA, by definition, provides reprogrammable logic best suited for state machine implementations, not for building processors.

Figure 5 shows the Xilinx FPGA based Traffic Manager implementation.

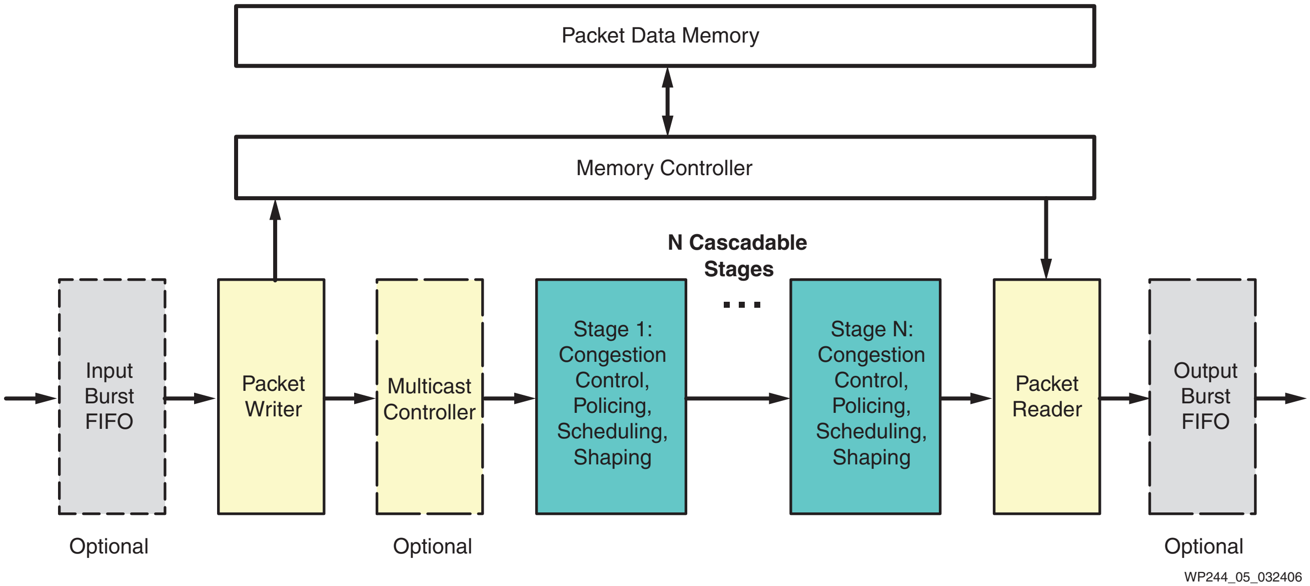

Figure 5: **Xilinx Traffic Manager**

The Input and Output Burst FIFOs provide for burst compensation when a channelized interface is used, such as SPI4.2. The Packet Writer, Packet Reader, and Memory Controller provide the means for the Traffic Manager to store packets. Descriptors are handed off to the Multicast Controller, if needed, which replicates the packets into a set of unicast queues as defined by the multicast group. The stages comprise the unicast queuing structures in a hierarchy. Each stage can accommodate up to 256K queues. Various congestion control, policing, shaping, and scheduling algorithms can be configured. In addition, various memories can be used in conjunction with each stage.

Each stage can provide 256K queues, with up to 5 stages. The purpose for the stages is to provide the hierarchies needed to arbitrate bandwidth from different classes of service into a subscriber and then between subscribers. The stages are just parallel sets of state machines that operate on buffer descriptors, achieving speed and efficiency.

Within a given stage, the congestion control state machine operates on a queue based on the threshold. If the queue fills beyond a set threshold, the state machine has to do something, such as drop packets, throw away packets at a rate, or set a flow control signal. The shaping state machine looks at the queue's list of packets to dequeue and makes a decision based on the token count for the queue whether to dequeue the packet or not. The scheduler state machine looks at the ready-list of packets per queue waiting to be dequeued once handed off by the shaper. The scheduler makes a decision for each queue based on the scheduling algorithm criteria. After a packet is set to dequeue from the scheduler, a decoder decides to which queue the packet is passed in the next stage. This is a logical queue.

Policing is used to enforce a rate upon a given stream. Unlike shaping, policing is done before a packet is enqueued. When a packet is ready to be enqueued, the policer checks if the packet is allowed based on the token counts for that queue. If there are not enough tokens, the packet is dropped.

Architecture Challenges

Clearly, to efficiently utilize the FPGA, the state machines have to be time-sliced or shared in time for multiple contexts. A queue is a context. For 256K queues, scaling can be an issue. What if only one queue is active? What if all queues are active? To accommodate all of these issues, the state machines are event-driven, which means they only operate on an active queue. The case where all queues are being used is not a problem because the worst-case packet rates cannot fill the queues faster than the Traffic Manager can act upon them.

Another concern is memory structure. This implementation uses a linked-list memory structure for storing packets and queues, which provides the most efficient utilization of the memory. Special optimizations have been put in place to minimize any memory bandwidth loss due to maintaining free-lists, and so forth.

Making all of these functions scale is another challenge. Basically, the logic scales based on the datapath width, which can be adjusted based on the rate. In addition, the size of the data structures scales with the number of elements.

Virtual Output Queuing

If configured in the Traffic Manager, optional flow controls can exist everywhere. On the output of the Traffic Manager, a flow control interface allows the user to turn off any queue anywhere in the hierarchy. This feature allows for virtual output queuing in which a message can be received from the switch fabric and translated into a flow control configuration into the Traffic Manager.

System Architecture Use Cases

Based on the Xilinx architecture, the following three use cases demonstrate the flexibility and scalability of the implementation. Assuming a distributed Traffic Manager architecture, three system examples are provided: Access line card, Metro line card, and UMTS base station.

These examples assume a distributed Traffic Manager system architecture (see Figure 6) as opposed to a centralized architecture (see Figure 7).

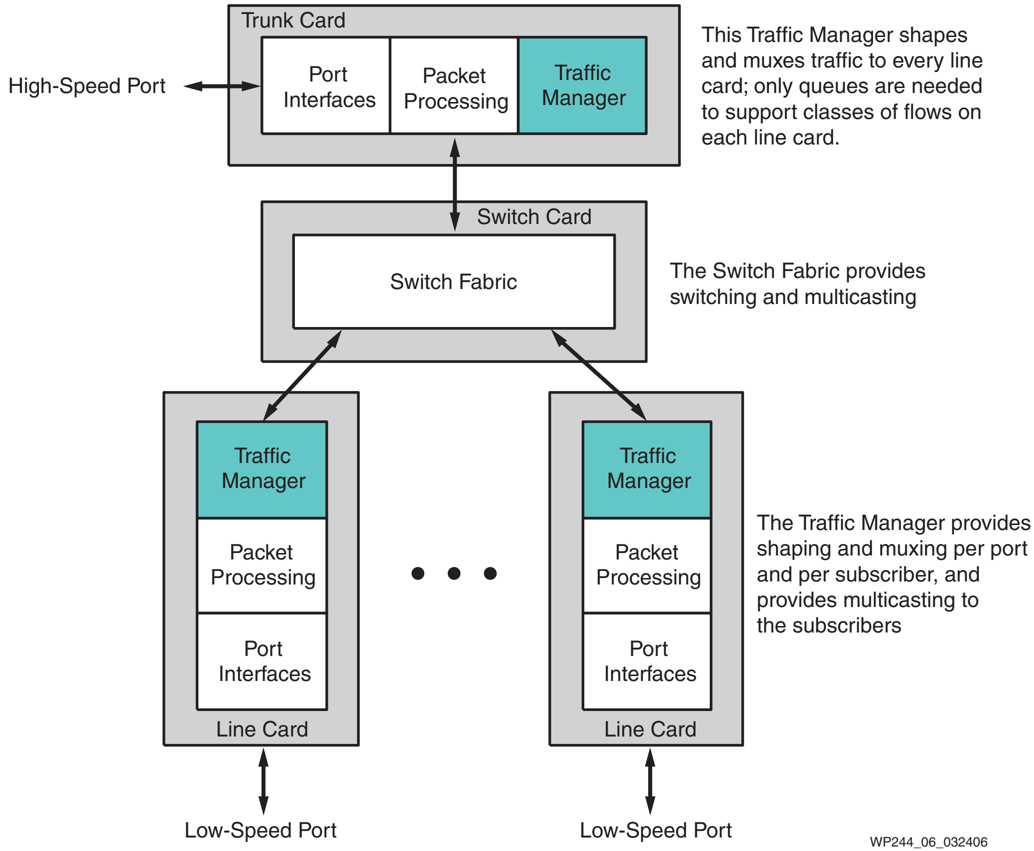

Figure 6: **Distributed Traffic Manager Architecture**

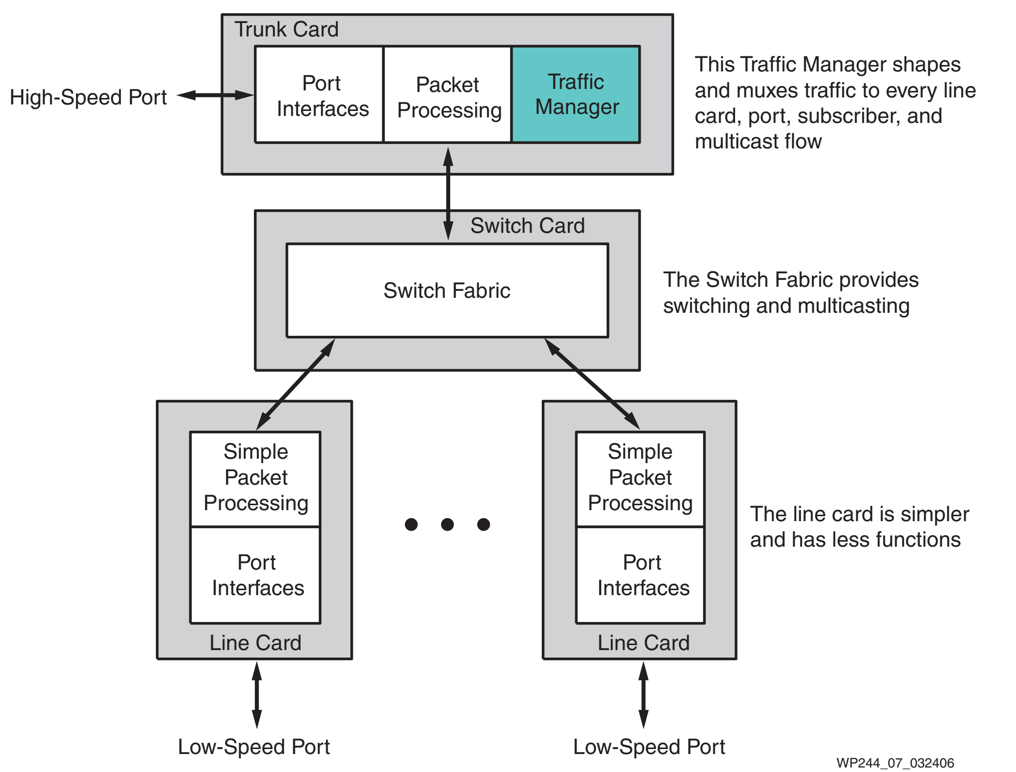

Figure 7: **Centralized Traffic Manager Architecture**

The more common distributed architecture provides for more efficient utilization of system bandwidth inside the equipment due to shaping and arbitration for every card into the switch fabric. One line card cannot block other line cards, if set up properly. In addition, the cost is more distributed throughout the system, such that customers do not pay for more than they use.

Access Equipment Line Card

An access equipment line card in today's networks delivers Voice, Video, and Data services to the subscriber. For this example, a 48-port VDSL card provides 55 Mb/s downstream and 2.3 Mb/s upstream, or 57.3 Mb/s (rounded to 60 Mb/s) per subscriber in the worst case. As a result, the Traffic Manager must shape and arbitrate traffic to each port in the downstream direction and to the shared access network bandwidth in the upstream direction. The total aggregate bandwidth (upstream + downstream) is 48 * 60 or 2.88 Gb/s.

For video, because there are never more than 1000 channels in any known scenario, there must be 1000 multicast groups. Each video channel consumes at least 2 Mb/s. Because worst-case subscribers would have 16 different video stream endpoints on their network, the multicast groups need to have 48 * 16 or 768 members per multicast group, where only 768 are active at any one time. Thus the worst-case bandwidth consumed by video is 768 * 2 Mb/s or 1.536 Gb/s. There is one queue for video per port, and the shaper must be updated for the rate each time a channel is added or removed. IGMP control packets are received per stream per port to control the channel selection. The IGMP packets must be captured and sent to the local host CPU, which would in-turn update the video multicast group and the rate for the given port.

Voice traffic is real-time. Although it is very latency sensitive, it does not consume much bandwidth. In the worst case, assume each voice service consumes 8 Kb/s and there are 32 calls at most per port. This case leads to 8K* 32 * 48 or 13 Mb/s of total voice traffic. Again, there should be one queue for all voice traffic per port with the rate controlled accordingly. Voice traffic is bidirectional, so the amount of bandwidth consumed for both directions totals 26 Mb/s.

The rest of the bandwidth is given to Data for each direction for best effort delivery. Assuming that each port can have 128 users, each user should have at least one application queue (arbitrary number) that results in 128 * 48 or 6K queues per direction, and 12K total.

Figure 8 shows the Traffic Manager architecture.

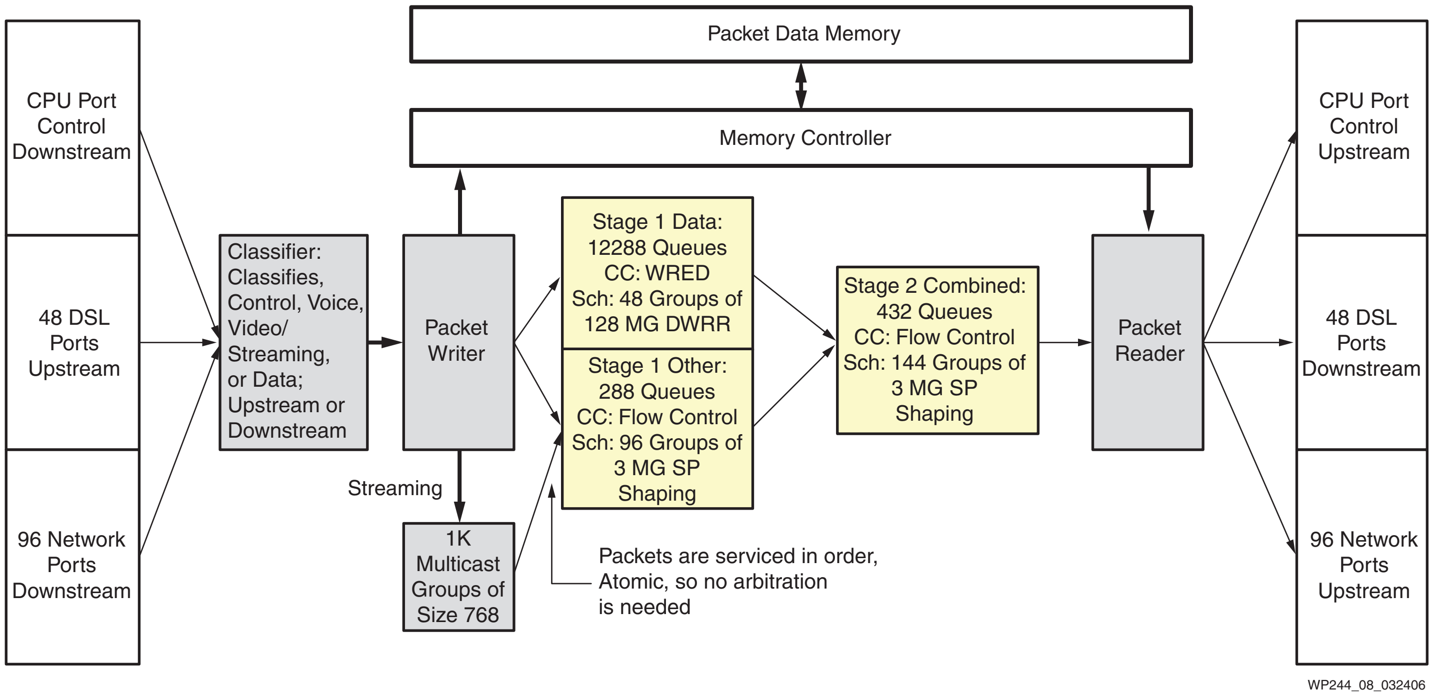

Figure 8: **VDSL Line Card Traffic Manager**

On the switch side of the line card, it is assumed that 96 channels are received. This arbitrary number can be higher or lower. The packets are received from this interface, passed to the classifier, and then passed to the Packet Writer. The Packet Writer segments the packet and stores it in external memory. A descriptor is written for the packet and is handed off to the queuing stage in accordance with the queue number provided on the Packet Writer interface. Streaming video packets are handed off to the multicast stage in which the descriptor is replicated to each unicast queue assigned to the group. The Stage 1 that is not data has 96 groups of 3 queues arbitrated by strict priority: 48 groups for downstream and 48 groups for upstream. Video is sent to the Priority 3 queues. Control data from any of the ports is directly written to the Priority 1 queues. Voice or any real-time data is written to the Priority 2 queues.

The Stage 1 associated for Data has 96 groups of 128 queues: 48 groups for downstream and 48 groups for upstream. Each group is arbitrated by a Deficit Weighted Round Robin (DWRR) scheduling algorithm. For any queues that overflow, Weighted Random Early Discard (WRED) throws packets out at a rate based on a slope defined by queue thresholds.

Stage 2 combines the Data and other (Video, Voice, and Control) into 144 groups of 3 queues that are arbitrated by strict priority. Video, Voice, and Control are given the highest priority, and Data gets the rest. There are 96 groups for the upstream network side channels and 48 groups for the downstream ports.

Shaping is provided on every strict priority queue such that bandwidths can be controlled especially if the traffic becomes *bursty* in nature. Data traffic is best effort delivery and, therefore, does not need to be limited because all other traffic is prioritized above it. This allows the maximum data to be delivered to the subscriber.

Super queuing structures can be created by putting queuing stages in parallel.

This type of design can fit in a Xilinx Virtex™-4 LX40 FPGA with some DDR2 SDRAM and DDR2 CIO SRAM.

Metro Equipment Line Card

In this application, a given Metro Equipment line card can service more than one access network. Assuming that a given access network services at least 32 nodes and that each network is at least 10G, it is reasonable for a Metro Equipment line card to service 20G or 2 access networks of 32 nodes. Each node has a mix of Voice, Video, Data, and Control traffic. The Traffic Manager on the line card, at a minimum, needs to enforce the QoS per Class of Service and allocate bandwidth to each node.

Given that streaming video traffic is to be distributed and due to the potential bursty nature of video when set to a variable bit rate (VBR), each channel has to be individually shaped.

Figure 9 shows the Traffic Manager architecture.

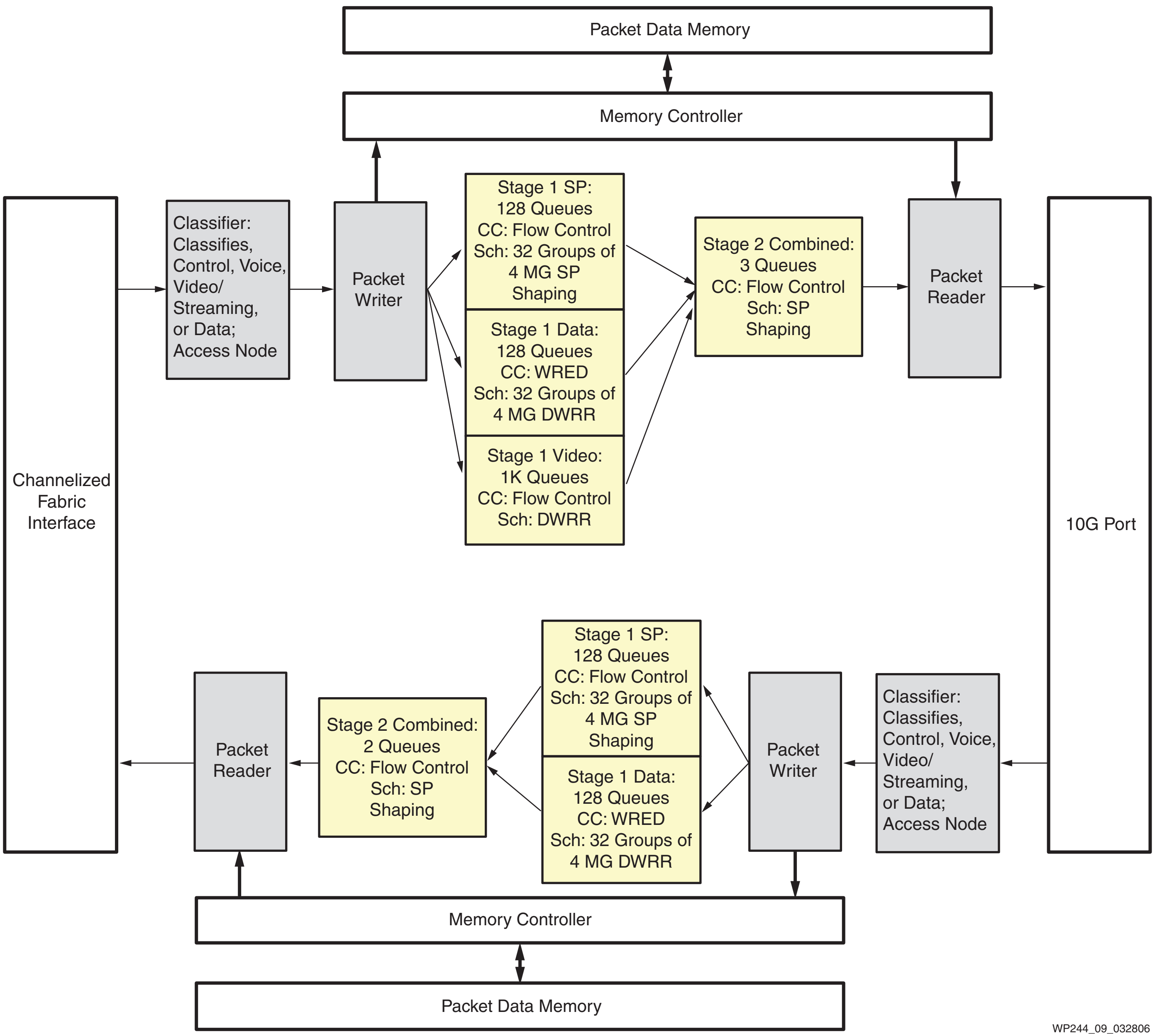

Figure 9: **Metro Equipment Line Card Traffic Manager**

For the downstream direction, packets are received from the fabric interface with the access node address and classified as Voice (really real-time), Video, Control, or Other. The classifier provides the queue ID for the packet. There are three parallel stages:

- The top stage provides a strict priority scheduler for Control and Voice traffic for each access node, flow control ensures that no packets are dropped if the queues become congestion, and shaping controls the rate such that no one queue can block the others.
- The middle stage provides data traffic management of four different priorities per access node, arbitrates via DWRR, and uses WRED to discard packets as the queues become full.
- The bottom stage, there for video or streaming traffic, provides 1K queues, 1 per channel. The purpose of these queues is to control the rate of the video on a per channel basis and ensure fair arbitration using DWRR.

The second stage combines each of the three stages into a priority queue. Packets are dequeued based on the priority setting. The priority order is:

1. Control and Voice (highest)
2. Video
3. Data (lowest)

Because the egress port is a 10G single port, the packets are dequeued one at a time.

For the upstream direction, data is received from the 10G port and is classified according to the same criteria as the ingress. The first stage is split in two. The top stage queues the Control, Voice, and any other high-priority traffic. The bottom stage queues the data and provides fair arbitration between all data streams. In the second stage, the packets are arbitrated via strict priority between the two first stages. The data traffic is the lowest priority. The last stage can also be equivalent to the number of channels on the fabric interface.

This type of Traffic Manager can fit into a Virtex-4 LX80 FPGA with external RLDRAM II and QDRII SRAM.

UMTS Base Station

In 3GPP release 5, HSDPA is a new data distribution service for wireless subscribers. This service, the first "broadband" type of service to be standardized in 3GPP, brings a whole new set of problems to the base station. Bandwidth usage, for mostly voice service, is relatively low and well controlled. The voice service scenario allows the bandwidth per subscriber to be very predictable and controllable with less sophistication built into the equipment. When providing a "broadband" data service, this consistency changes. Subscribers are not all accessing the same application at the same time. The bandwidth utilization per subscriber can be bursty, hence, less predictable and less controllable. A sophisticated Traffic Manager is needed to ensure fairness across all subscribers and to provide the best utilization of the wireless shared medium.

HSDPA provides a shared set of channels that can continuously be reassigned to a different set of subscribers. There are 15 code channels available that can be reassigned every 2 ms. This Traffic Manager provides two main functions: prioritized statistical multiplexing for classes of service per subscriber and assignment of radio channels every 2 ms for the subscribers.

Considering there are 15 code channels, each channel can support a maximum of 2 Mb/s throughput, requiring 30 Mb/s of total throughput to be managed. Assuming there are 128 subscribers with 4 classes of service, 512 queues are required scheduled in groups of 4 using strict priority. Next, shaping needs to be done per subscriber to limit the rate to the subscription rate (if required). Multicasting is convenient when any kind of streaming media is being delivered to a subscriber.

Figure 10 shows the architecture.

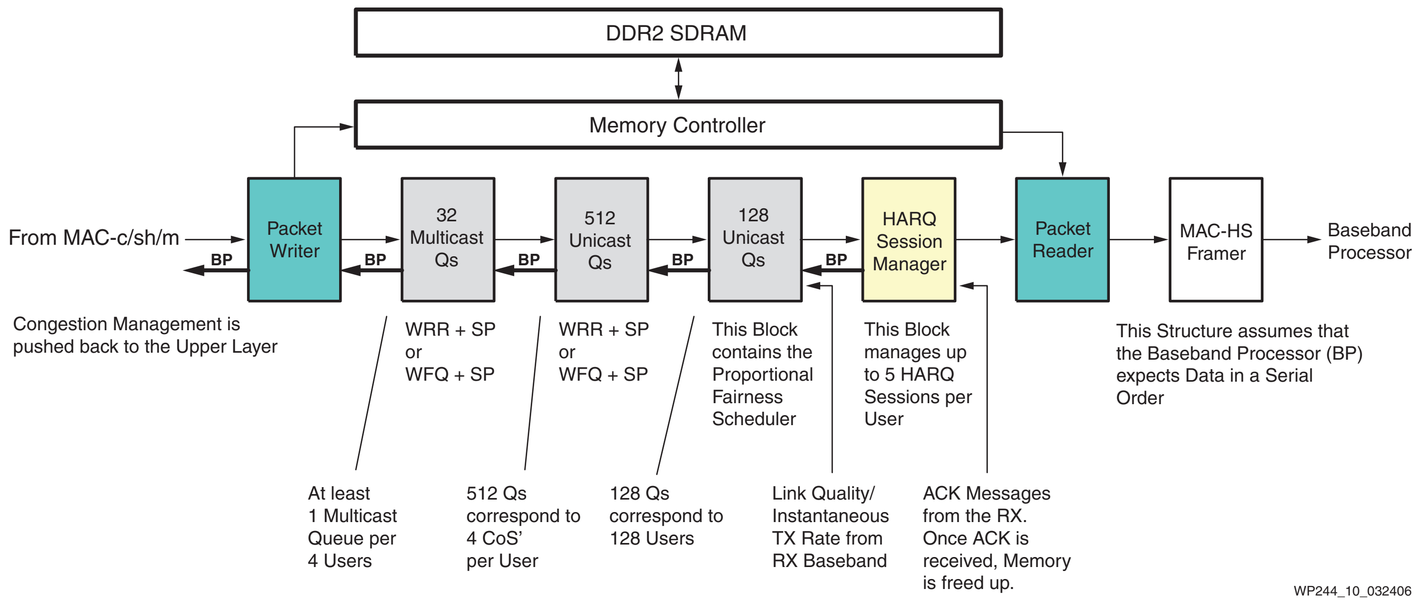

Figure 10: **HSDPA MAC (MAC-HS) + Traffic Manager**

These functions are very similar to the previous configurations except for a few things. The third stage has a different or non-standard Traffic Manager scheduling algorithm that is optimized for this wireless application. Proportional fairness or other scheduling algorithms can be provided. The HARQ Session Manager provides a Stop-and-Wait protocol per user. A window of data is sent to the subscriber, and then the session manager waits until an ACK is received from the subscriber. Multiple windows can be active per subscriber. If no ACK is received in a programmable time frame, the window of data is retransmitted.

There are several advantages to using the Traffic Manager in this application. The Traffic Manager ensures that each subscriber gets a fair share of bandwidth and that each class of service is ensured its quality level. In addition, the scheduling or assignments of channels to the subscribers are guaranteed to be deterministic and have predictable performance because the functions are in hardware. In addition, the HSDPA specification provides for 2 ms updates to the channel assignment. In congestion scenarios, where there are more subscribers than available channels, the average available bandwidth drops significantly as the number of subscribers grow. This Traffic Manager implementation in hardware can easily provide for scheduling the channels in 1 ms or even 1 µs ranges. If the subscriber equipment can accommodate the increased rate of channel assignment changes, the average bandwidth per subscriber can easily be doubled or tripled as the congestion grows, allowing for the number of subscribers to grow for a given set of channels.

This type of Traffic Manager can be implemented in a Spartan™-3 XC3S2000 FPGA or a Virtex-4 XC4VLX25 FPGA with some external DDR2 SDRAM.

Conclusion

In the growing field of Internet-based distribution of multimedia services, traffic management is required at all levels of the network. The Xilinx Traffic Manager is uniquely positioned to address all levels of the network based on the scalability, flexibility, and performance. ASSP and Von Neumann/processor-based architecture do not have the flexibility and scalability required to compete.

More information is available at http://www.xilinx.com/qos.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/10/06	1.0	Initial Xilinx Release.