Solving the Big Dilemma of Big Data

shaping tomorrow with you

FUJITSU

Our thirst for information and communication is, it seems, insatiable. This is particularly apparent in the online world, where data sets are getting steadily larger. These massive data sets, rightly nicknamed "big data," are not only ballooning in size but are also being transmitted over communications networks with increasing frequency. While scientific research, such as atmospheric modeling and astronomical exploration, continue to be the bellwethers of the big data phenomenon, the healthcare professions are not far behind them, especially with the advent of health information exchanges and the growing frequency with which large, high-definition medical images are transmitted electronically.

Ultimately, we will all be following the lead of the scientific and healthcare fields by transmitting larger and larger chunks of data and bigger files. And as the size of the files grows, it becomes more difficult to push them through the network. In the case of scientific applications, a single data file of terabyte size can take hours, days, or even longer to traverse the network. Extended transmission times of this magnitude are obviously not conducive to the productivity of the people waiting for the data to arrive. Long transmission times for huge files can also exert pressure on an entire network, slowing other traffic and reducing the efficiency of everyone using it. Think of what happens on a major freeway when multiple lanes are occupied by big trucks.

The Additional Challenge of Real-Time Data

The transmission of huge data files is not the only aspect of the overall "big data dilemma." Real time data transmission is also becoming increasingly commonplace. Real time transmission effectively translates to low latency and sufficient throughput; the organizations using the data require minimal delay and reliable throughput to avoid interruptions in service. Some examples of these services include VoIP, streaming audio, and video on demand. High-frequency stock trading is another example. In typical real time situations, the correct sequence of the actual data packets is as critical as the transmission speed. Otherwise the packets are not useful and are simply dropped.

In the case of both "big data" and "real time" data, the common thread of transferring this information is sufficient throughput. However, throughput is a more complex topic than it might at first seem; it is not just a question of the physical link speed. Other factors also affect throughput such as latency, error rates, and window size. This paper explores how these other factors impact throughput and proposes a network architecture that helps to mitigate these factors, while still providing the flexibility of a packet network.

Throughput: It's Not Just Link Speed

There is no doubt that link speed plays a critical part in the throughput equation. The slowest link speed in the path sets the upper bound of the throughput of the network, effectively determining the maximum possible throughput. However, latency, errors, and window size also play a related role and can reduce throughput significantly below the link speed.

Window Size for Flow Control

Window size for use with flow control is usually associated with the TCP protocol, since it provides a window of transmission before return acknowledgements are required. However, window sizes are not only associated with TCP. Although they are not as obvious an example as the TCP window, other protocols may also utilize a similar flow control mechanism. Although UDP, for instance, does not have direct acknowledgements, the application layers that use UDP as a transport layer may have some type of acknowledgements to indicate whether or not transmission was successful. The maximum TCP window can easily be changed to adjust for the particular network. In the case of other application-based protocols, however, the window size may be fixed.

The window size is critical to network throughput, and it must be optimized for the specific network—neither too big nor too small. If the window size is too small, the protocol loses throughput while waiting for acknowledgements, so a portion of the transmission pipe is left empty. On the other hand, if the window size is too large, there is the danger of overflowing memory buffers, which could result in lost packets which would affect throughput. Another downside of a window that is too large is that errors may not be corrected for a significant amount of time, which has a detrimental effect on real time traffic. This applies to both TCP window sizes and other protocols that utilize a window construct.

Latency

The real culprit for reduced throughput is latency. Essentially, throughput is affected by the interaction between latency and window size. Since latency impacts responsiveness to the acknowledgement of packets and the resolution of errors, it is the key to maximizing throughput, once high link speeds are established.

FUJITSU NETWORK COMMUNICATIONS INC.
2801 Telecom Parkway, Richardson, Texas 75082-3515
Telephone: (972) 690-6000
(800) 777-FAST (U.S.)
us.fujitsu.com/telecom

There are certain aspects of latency that cannot be avoided due to such things as the speed of light through a fiber. But the total latency comprises multiple components. Other causes of latency are queuing delays and processing delays. Processing latency can be minimized by setting up predetermined paths with the packets being switched via embedded labels so that processing or lookup time is minimized. However, queuing delays may still play a significant role in latency. With most types of modern packet networks, explicit routes and priority queuing may be set up to minimize queuing delays for particular services. However, this typically requires special engineering, adds complexity, and may require capacity upgrades on the traversed routers and links to meet the latency requirements. This could result in additional expense for the network.

Errors

As stated earlier, errors impact throughput due to the need to correct the corrupted or missing packet and the time that it takes to do this. Two types of errors are possible, a spurious error that occurs during transmission and a discarded packet error.

Spurious errors are typically single or several bits that have been misinterpreted by the receiver due to noise or some other type of impairment. If the error cannot be corrected, then the entire frame is discarded. Other discarded packets occur as a result of traffic congestion and involve the loss of the entire packet. These lost packets can usually be avoided by proper traffic engineering, and some spurious errors might be corrected if the links support Forward Error Correction capabilities; however, not all errors can be avoided. The diagram shown in Figure 1 shows the effect of packet loss and latency on throughput.

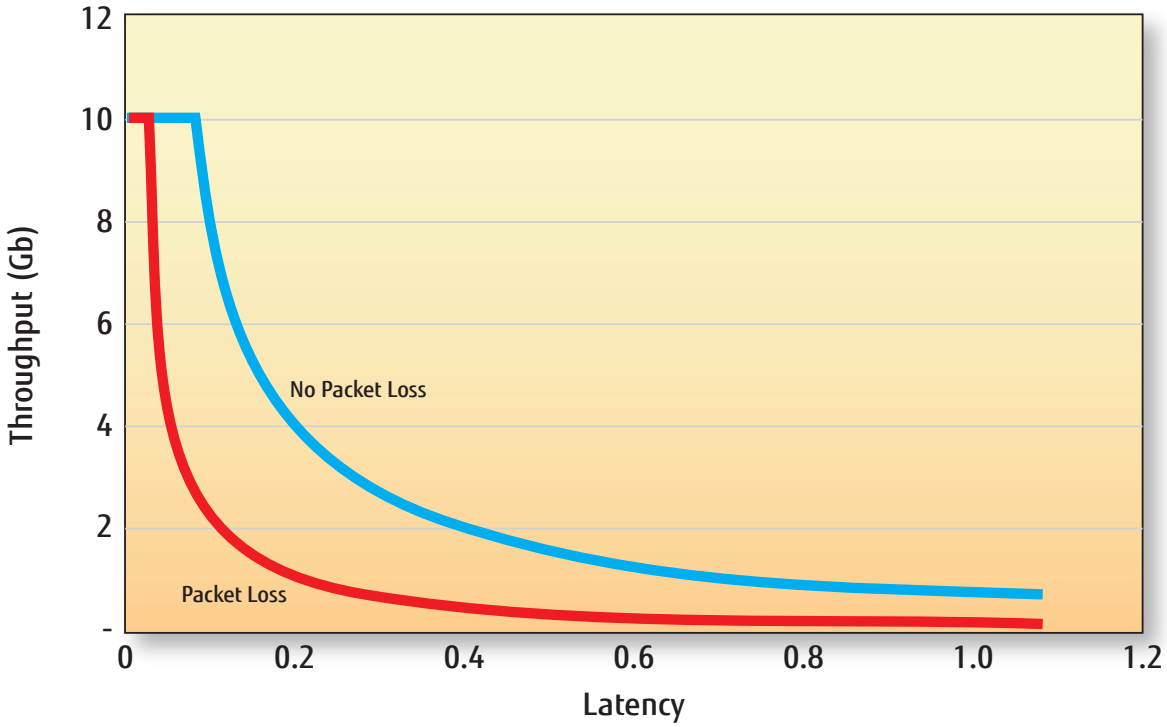

Figure 1: Throughput With and Without Packet Loss

Protocols

Protocols affect throughput based on the elements already described, namely latency, errors and window size. Different protocols exert different influences on throughput. Protocols that utilize TCP, such as FTP and HTTP, are affected by the maximum TCP window size, for example. Other protocols that instead use UDP might have better throughput capabilities for streaming data, but might result in packets arriving out of sequence. The use of newer protocols such as SCTP might help with multiple streams. The important point is that the protocol used can very much affect the throughput and should be carefully considered when transferring large files or real time data.

Aspects of Network Latency

Multiple aspects of a network can cause latency. These exert their effects at different layers of the OSI stack.

Span distances, a Layer 1 function, are not easily changeable and can contribute significantly to latency. Span distance deals with the time it takes for light to traverse the span through the fiber optic cable. This can be considered a "primary" latency effect.

Other components of latency are the protocol used and the queuing latency, so these might be considered "secondary" latency effects. These are higher layer functions at Layers 2, 3 and 4 of the OSI stack.

Errors could have a similar type of label associated with them, as spurious errors are a nature of the physical conditions of the line (SNR), so they might be considered a "primary" element of latency and impose it at Layer 1. Discarded packets due to congestion, however, are a function of the queuing structure and the traffic within the queues of the switches/routers. Therefore these might be considered a "secondary" effect, imposing delays at Layers 2, 3 and 4. Although errors do not directly affect latency, they indirectly cause delays by injecting retransmissions of errored packets and subsequent packets sent before the error was detected. The net effect of numerous retransmissions is that throughput is reduced.

Link speed is another aspect of throughput or latency that can be considered a "primary" effect, imposing latency at Layer 1.

The point of this is that some aspects of latency can't be changed due to the physical nature of the latency. These are the primary or Layer 1 effects of latency. However, other aspects of latency can be managed and perhaps changed because they are the result of higher layer protocols instead of fundamental physical effects. The effects of packet loss, queuing delays, and protocols would all be considered in this category. These are the secondary effects of latency, originating at the higher layers of the OSI protocol.

Getting to Maximum Throughput

The physical nature (Layer 1) of a network dictates the lower bound of the latency and, when combined with the windowing factors, determines the resulting network throughput. Once the span distances are determined and the link speeds understood, the lower limit of the latency can be calculated. In an ideal world, this would be the entire latency of this ideal network.

However, since this is not an ideal world, other elements of latency need to be considered. This is where throughput gets complicated as it depends on the number of routers traversed, the queuing and traffic structure through the routers, the protocols being used, and the number of errors (spurious errors that are not corrected and dropped packets) that are being encountered.

If these secondary latency contributors could be minimized, then it would be possible to approach the ideal network throughput. For example, if there were no traffic congestion (queuing delays) and no chance for packets to be dropped, then throughput could be maximized. Essentially if the network could act as a single fiber span with no routers between the two endpoints, this would be the ideal network.

Although this is not a reasonable expectation given the amount of packet traffic that is in the network today, certain network architectures might lend themselves to act like this ideal network more than others. A network that is essentially a set of routers connected together over fiber allows for high flexibility, but it might not allow for maximum throughput unless each path within it was engineered for low latency. Although this is certainly possible with any modern connection-oriented packet network, it might result in higher complexity and higher costs as additional capacity would likely be needed. On the other hand, a network that provides switchable Layer 1 functionality, such as OTN, would approach the ideal network as there are neither queuing delays nor packet losses. However, a pure OTN network may not provide all of the flexibility that is needed for transporting packet traffic.

It is important to note that, even in an ideal network with minimal delay and a low BER, throughput is still dependent on a transfer protocol that will allow large enough windows to keep the throughput at maximal levels. This is strictly a client-based dependency and not associated with the network architecture.

Combining OTN and Packet to Optimize Throughput

A good compromise that provides both the flexibility of a packet network and the low latency of a Layer 1 OTN network would be a hybrid of the two technologies. Packet networks are superior at converging small flows into larger aggregate flows and conversely, distributing large flows into smaller flows, whereas OTN networks are superior at transporting the aggregate flows over wide networks with minimal latency and maximal throughput. A network that allows for both technologies can be flexible enough to support edge-based aggregation with a packet network and then utilize OTN as its backhaul/transport mechanism.

Packet-based aggregation on the edge of the network allows small flows to be grouped together. Typically an edge network involves a relatively small number of destinations for the traffic, perhaps no more than a handful. For example, in a residential area, traffic might be broadband Internet traffic that is destined for an Internet PoP. Or it might be VoIP traffic destined for a VoIP gateway. In a business park, traffic might be destined for a company VPN. The cellular traffic in either a residential or business area might consist of two to three mobile carriers whose traffic is destined for their specific MTSO. The packet aggregation functionality can group all of the traffic together that is heading for one of the handful of destinations. In an edge application, as an example, there might be eight aggregate flows heading to eight different destinations. If the area of this packet aggregation is small enough, typical packet degradation such as packet loss, queuing delays, and excessive jitter can be minimized or avoided completely without excessive complexity or costs. See Figure 2 for an illustration of packet-based aggregation at the network edge.

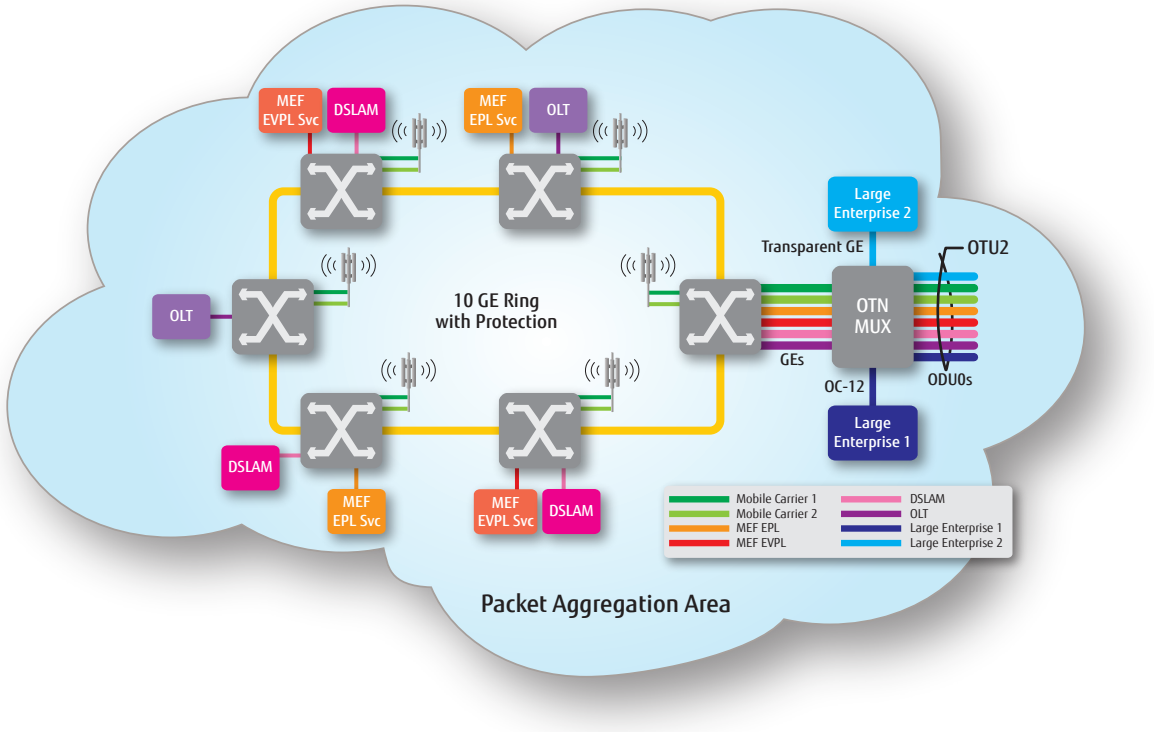

Figure 2 - Edge-Based Packet Aggregation. Aggregation can scale to higher speed pipes as needed

FUJITSU NETWORK COMMUNICATIONS INC.
2801 Telecom Parkway, Richardson, Texas 75082-3515
Telephone: (972) 690-6000
(800) 777-FAST (U.S.)
us.fujitsu.com/telecom

Once the flows are established, they can be mapped into OTN and then switched or steered to their proper destination at a Layer 1 level. Utilizing this Layer 1 technology allows for very low latency while still allowing flexible paths to the destination (Figure 3). The OTN layer also allows for correction of some spurious errors by utilizing the forward error correction capability that is a fundamental part of the OTN layer. This essentially improves the SNR to the links, thereby reducing the number of lost packets.

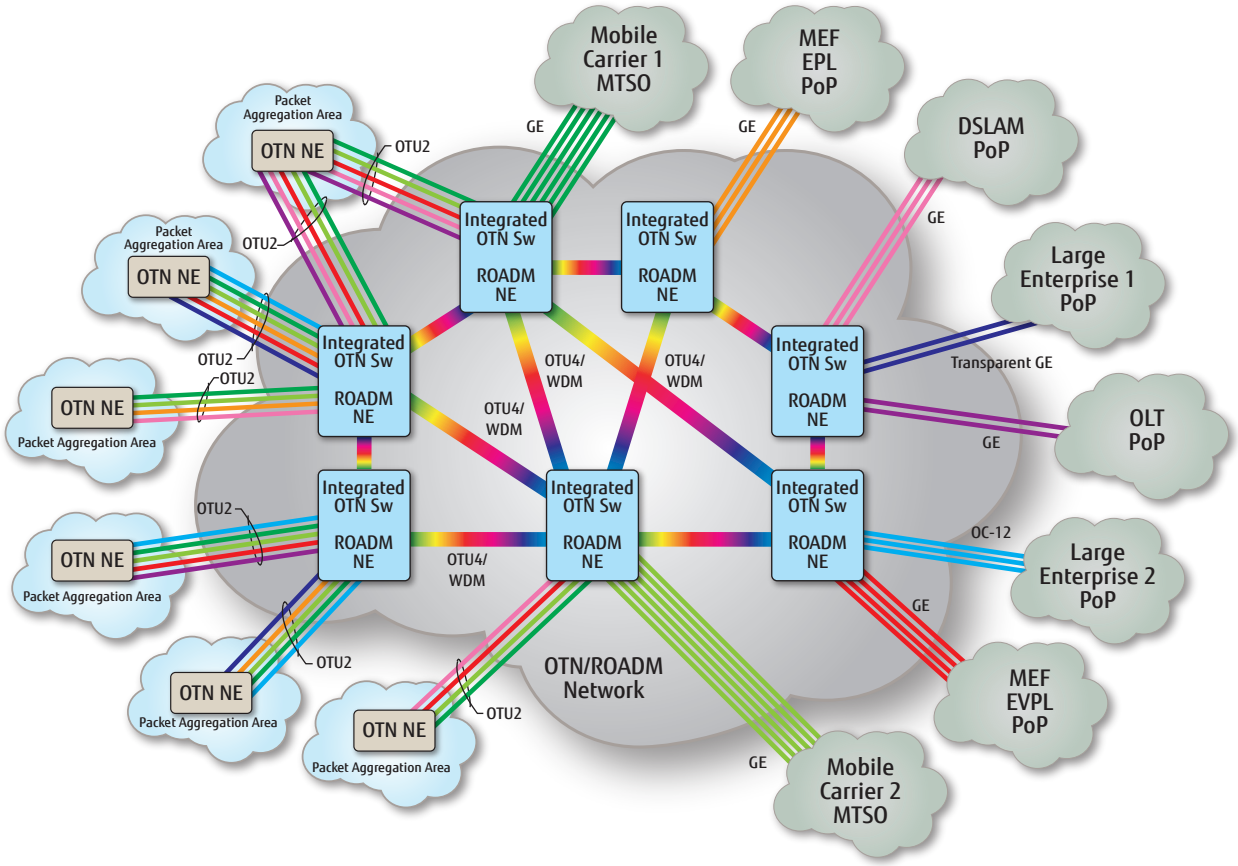

Figure 3 - OTN/ROADM Core with Packet Edge

Once the flows that are encapsulated in OTN reach their destination from the different packet edge areas, further packet aggregation can take place to build bigger flows, if necessary, to hand off to the ultimate endpoint.

Solving the Dilemma by Combining Throughput with Flexibility

In a network that mostly transmits small file sizes, throughput may not be an issue, since it is largely irrelevant to the end users if it takes a file 10 or 50 ms to traverse a network. However, the advent of big data is set to test the limits of people's patience and efficiencies by imposing transmission times of hours or even days. This is when throughput will become very important to end users and ultimately, to network designers.

There are some aspects of latency that are Layer 1-based, and these inherently define the lower bounds of the delay (i.e. the minimum possible latency). However, other aspects of latency can be managed and reduced by having the right kind of network. Note that even with an optimized network, it is still incumbent upon the end user or client to use a protocol that supports an adequate flow control mechanism to sustain maximum throughput.

A network that provides for packet aggregation on the edge allows for flexibility of the packet data, in that the traffic can be aggregated with other traffic that is heading to the same destination. This limited but sufficient packet capability also protects against traditional large packet network issues such as dropped packets, queuing delays, and excessive jitter, while still providing the packet bandwidth flexibility.

The switchable OTN network in the core allows for expediting and steering the traffic with a Layer 1 technology so that latencies are minimized and yet traffic can be easily directed to its proper destination. The OTN network is the closest thing to an ideal network that can be achieved without having a point-to-point fiber link for every flow.

Typically, combining multiple OSI layer technologies is frowned upon because this is perceived as contrary to the utopian vision of the converged network. However, in the case of "big data", which is very much on the horizon, the two technologies are synergistic with each other as they both bring complementary value to the network. Throughput without flexibility, as might be the case with a pure OTN network, may not provide the aggregation capabilities that are needed. Conversely, flexibility without throughput, which might be the case with a pure packet network, may not support cost-effective or efficient transport of big data or real time data. The solution to the dilemma is to have both: an edge packet network with a switchable core OTN network. This will provide the flexibility and the throughput needed for the coming "big data" storm.

Acronyms

Term	Definition
BER	Bit Error Rate
DSLAM	Digital Subscriber Line Access Multiplexer
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
MTSO	Multitechnology Switching Office
Mux	Muxponder
NE	Network Element
OSI	Open Systems Interconnect
OTN	Optical Transport Network
PoP	Point of Presence
ROADM	Reconfigurable Add/Drop Multiplexer
SCTP	Stream Control Transmission Protocol
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice Over Internet Protocol
VPN	Virtual Private Network

FUJITSU NETWORK COMMUNICATIONS INC.
2801 Telecom Parkway, Richardson, Texas 75082-3515
Telephone: (972) 690-6000
(800) 777-FAST (U.S.)
us.fujitsu.com/telecom