A

Seminar report
on

FAST AND SECURE PROTOCOL

***Submitted in partial fulfillment for the award of the* degree of**

MASTER OF COMPUTER APPLICATIONS

Submitted by

S HARIKRISHNA

(Reg. No. 19F65F0008)

Under the esteemed guidance of

Mrs.P.SUKANYA,MCA.

(Asst. Professor, Department of MCA)

Department of Master of Computer Applications

SIDDHARTH INSTITUTE OF ENGINEERING & TECHNOLOGY(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuram)(NAAC Accredited with 'A' Grade, NBA Accredited Institution)

Siddharth Nagar, Narayanavanam Road, Puttur-517583, Andhra Pradesh.

(2020-2021)

CONTENTS

INTRODUCTION

Aspera's *fasp* transfer technology is an innovative software that eliminates the fundamental bottlenecks of conventional file transfer technologies such as FTP, HTTP, and Windows CIFS, and dramatically speeds transfers over public and private IP networks. The approach achieves perfect throughput efficiency, independent of the latency of the path and robust to packet losses. In addition, users have extra-ordinary control over individual transfer rates and bandwidth sharing, and full visibility into bandwidth utilization. File transfer times can be guaranteed, regardless of the distance of the endpoints or the dynamic conditions of the network, including even transfers over satellite, wireless, and inherently long distance and unreliable international links. Complete security is built-in, including secure endpoint authentication, on-the-fly data encryption, and integrity verification.

Transport characteristics

- **Maximum speed**
  - Enables large data set transfers over any network at maximum speed, source disk to destination disk, regardless of network conditions or distance.
    - Over gigabit WANs with 1 second RTT and 5% packet loss, achieves 700-800 Mbps file transfers on high-end PCs with RAID-0 and 400- 500 Mbps transfers on commodity PCs.
    - Even with extreme packet loss and latency (30%/1 second RTT), sustains over 100 Mbps WAN transfers, and fills a satellite

transponder.

- Transfers large data sets of small files with the same efficiency as large single files.
- Very lightweight. Does not require specialized or powerful hardware to maintain high speeds.

- **Extraordinary bandwidth control**
  - Provides precise rate control (pre-set and on-the-fly) for guaranteed transfer times.
  - Uses an automatic adaptive rate control to fully utilize available bandwidth while remaining fair to other traffic.
  - Provides fast, automatic discovery of the bandwidth capacity between the source and destination.
  - Supports on-the-fly configurable bandwidth sharing policies. Users may pre- set and change individual transfer rates and finish times as needed.
  - Supports perfect progressive-style transfers, e.g. for media playout. Transfer speeds do not degrade with congestion or distance, ensuring smooth, immediate processing of the incoming data.
- **Complete security**
  - Includes complete, built-in security using open standard cryptography for user authentication, data encryption and data integrity verification.
- **Software only**
  - Uses standard, unmodified IP networking and is implemented in software as an application protocol. Requires no changes to the operating system or driver

installation on the file transfer endpoints, no new appliances, and no network changes.

- **Robust**
  - Provides end-to-end transfer progress reporting and detailed performance statistics for monitoring and billing, as well as custom pre- and post-transfer processing.
  - Automatically resumes partial transfers and retries failed transfers.
- **Flexible open architecture**
  - Supports interoperable file and directory transfers between all major operating systems and provides a complete, modern software API to build upon.

Need for fasp

In this digital world, fast and reliable movement of digital data, including massive sizes over global distances, is becoming vital business success across virtually every industry. The Transmission Control Protocol(TCP) that has traditionally been the engine of this data movement, however has inherent bottlenecks in performance(fig 1), especially for networks with high round-trip time and packet loss, and most pronounced on high-bandwidth networks. It is well understood that these inherent "soft" bottlenecks are caused by TCP's Additive

Increase Multiplicative Decrease(AIMD) congestion avoidance algorithm, which slowly probes the available bandwidth of the network, increasing the transmission rate until packet loss is detected and then exponentially reducing the transmission rate. However, it is less understood that other sources of packet losses due to physical network media, not associated with network congestion equally reduce the transmission rate. In fact, TCP AIMD itself creates losses, and equally contributes to the bottleneck. In ramping up the transmission rate until loss occurs, AIMD inherently overdrives the available bandwidth. In some cases, this self-induced loss actually surpasses loss from other causes (E.g. Physical media) and turns a loss free communication

"channel" into an unreliable "channel" with an unpredictable loss ratio.

The loss-based congestion control in TCP AIMD has a deadly impact on throughput: Every packet loss leads to retransmission, and stalls the delivery of data to the receiving application until retransmission occurs. This can slow the performance of any network application but is fundamentally flawed for reliable transmission of large "bulk" data, for example file transfer, which does not require in-order (byte-stream delivery).

This coupling of reliability (retransmission) to congestion control in TCP creates a severe artificial throughput penalty for file transport, as evident by the poor performance of traditional file transfer protocols built on TCP such as FTP, HTTP over wide area networks. Optimizations for these protocols such as TCP acceleration applied through hardware devices or alternative TCP improve file transfer throughput to some degree when round trip times and packet loss rates are modest, but the gains diminish significantly at global distances,

Furthermore as we will later see, parallel TCP or UDP blasting technologies provide an alternative means to achieve apparently higher throughputs but at tremendous bandwidth costs. These approaches retransmit significant, sometimes colossal amounts of unneeded file data, redundant with data already in flight or received, and thus take many times longer to transfer file data than is necessary, and cause huge bandwidth cost. Specifically, their throughput of useful bits excluding retransmitted data packets—"goodput"—is very poor. These approaches deceptively appear to improve network bandwidth utilization by filling the pipe with waste, and transfer times are still slow!

For the narrow network conditions under which TCP optimizations or simple blasters do achieve high good data throughput, as network-centric protocols, they run up against further soft bottlenecks in moving data in and out of storage systems.

Transporting bulk data with maximum speed calls for an end-to-end approach that fully utilizes

available bandwidth along the entire transfer path – from data source to data destination - for transport of "good data" – data that is not in flight or yet received. Accomplishing this goal across the great range of network round-trip times, loss rates and bandwidth capacities characteristic of the commodity internet WAN environments today requires a new and innovative approach to bulk data movement, specifically, an approach that fully decouples reliability and rate control. In its reliability mechanism, the approach should retransmit only needed data, for 100% good throughput. In its rate control, for universal deployment on shared internet networks, the approach should uphold the principles of bandwidth fairness, and congestion avoidance in the presence of other transfers and other network traffic, while providing the option to dedicate bandwidth for high priority transfers when needed.

Aspera fasp is an innovative bulk data transport technology built on these core principles that is intended to provide an optimal alternative to traditional TCP-based transport technologies for transferring files over public and private IP networks. It is implemented at the application layer, as an endpoint application protocol, avoiding any changes to standard networking. Fasp is designed to deliver 100% bandwidth efficient transport of bulk data over any IP network – independent of network delay and packet loss – providing the ultimate high-performance next generation approach to moving bulk data.

MAXIMUM TRANSFER SPEED	
Maximum end to end transfer throughput independent of WAN conditions	Transfer speed is independent of network latency and is robust to packet loss, ensuring maximum speed over even the longest and most difficult WANs(1 sec RTT/30% packet loss+).
Lightweight implementation	Does not require specialized hardware for high throughputs. Maximum throughput is obtained through a single transfer stream – multiple connections are not needed for high speeds.
Equally fast for transfers of sets of small files	Transfers large data sets of small files with the same efficiency as large single files.
SCALABILITY AND EFFICIENCY	
No theoretical throughput limitation and optimal efficiency where accelerators fail	Unlike TCP-based protocols and UDP-based blasters that lose efficiency with network delay and packet loss. fasp achieves nearly 100% good data throughput.
Higher efficient server scaling to hundreds of concurrent transfers per server host	Unlike UDP based blasters, supports hundreds of concurrent transfers on commodity hardware. Aggregate throughput scales nearly linearly with number of flows, and concurrent flows maintain fair and stable bandwidth sharing.
Automatic resume of partially or failed transfers	Aspera transfers effectively continue where partial or failed attempts left off, increasing effectiveness and

	encrypted envelope per recipient, aspera products support the ability to leave transferred files encrypted in storage. The crypto library is FIPS 140-2 compliant, and the cipher is pluggable allowing for new ciphers to be applied as stae-of-the-art standards evolve in the future.
Data integrity verification for each transmitted block	Fasp accumulates a cryptographic hashed checksum for each datagram, appending the digest-hash to the secure datagram before it goes on the wire, to be checked at the receiver to verify message integrity, preventing a man-in-the middle attack and ensuring data integrity.

TRANSFER MODES

TRANSFER PARADIGMS	
Client-Server	Upload or download large files or collections of files between clients and servers over any network and distance, at maximum speed, securely, and with exceptional bandwidth control. Aspera's server and client software runs on all major OS's, and offers many other interface types
Site-to-site transfer	Aspera software supports point to point and point to multipoint transfer capability in push and pull workflows
Multi-site synchronization	Synchronize large directories and file stores one-way, two-way or N-ways across sites.

Table 2: Transfer Modes

Person to person	Deliver large files or collection of files directly to individuals or groups. Senders can use either a browser or desktop-based interface to compose digital packages and address them to individuals via e-mail
Automated	Variety of transfer automation capabilities, including watch folders, automatic forwarding
Remotely controlled	Enterprise-wide centralized transfer management.
Local cloud	Utilize local infrastructure or seamlessly expand local servers on demand, to elastic cloud infrastructure.

Management and Automation

MANAGEMENT	
Comprehensive transfer and bandwidth management	Across the entire product line, manage transfers and apply bandwidth control in real time at the endpoint through graphical controls remotely via a centralized web interface, through pre-controlled transfer and bandwidth policies.
Server management	Comprehensive configuration of server software from a browser, GUI or command line.

Table 3: Management and Automation

AUOMOTION	Complete server side policy management. Schedule aggregate bandwidth targets by time of day.
Automatic download	Unattended automatic download of newly available files on the server via both light and regular desktop clients
Automatic forwarding	Automatically forwards downloaded files to specified targets.
Transfer queuing and reordering control	The files to be downloaded can be queued in and they can be rearranged according to the priority .

SHORTCOMINGS OF TCP TRANSFER

Transferring large data sets—big files, big collections of files—via inexpensive IP networks, instead of shipping tapes, discs, or film, promises to change fundamentally the economics of content production, distribution, and management. Under ideal conditions, data may be moved quickly and inexpensively using ordinary file transfer methods such as FTP, HTTP, and Windows CIFS copy. However, on real wide-area and high-speed network paths these methods' throughput collapses, failing to use more than a small fraction of available capacity. This is a consequence of the design of TCP, the underlying protocol they all rely on. New TCP stacks and new network acceleration devices are marketed to help, but they fail to fully utilize many typical wide-area network paths. Consequently, conventional FTP, and even new "acceleration" solutions, cannot provide the speed and predictability needed for global file transfers.

The TCP bottleneck in file transfer

The transmission control protocol (TCP) that provides reliable data delivery for conventional file transfer protocols has an **inherent throughput bottleneck** that becomes more severe with increased packet loss and latency. The bar graph shows the maximum throughput achievable under various packet loss and network latency conditions on an OC-3 (155 Mbps) link for file transfer technologies that use TCP (shown in yellow). Transmission rates are defined by rate of the bitstream of the digital signal and are designated by hyphenation of the acronym **OC** and an integer value of the multiple of the basic unit of rate, e.g., OC-48. The base unit is 51.84 Mbit/s. Thus, the speed of optical-carrier-classified lines labeled as OC-*n* is *n* × 51.84 Mbit/s. The throughput has a hard theoretical limit that depends only on the network round-trip time (RTT) and the packet loss. Note that adding more bandwidth does not change the effective throughput. File transfer speeds do not improve and expensive bandwidth is underutilized. OC-3 is a network line with transmission speed of up to

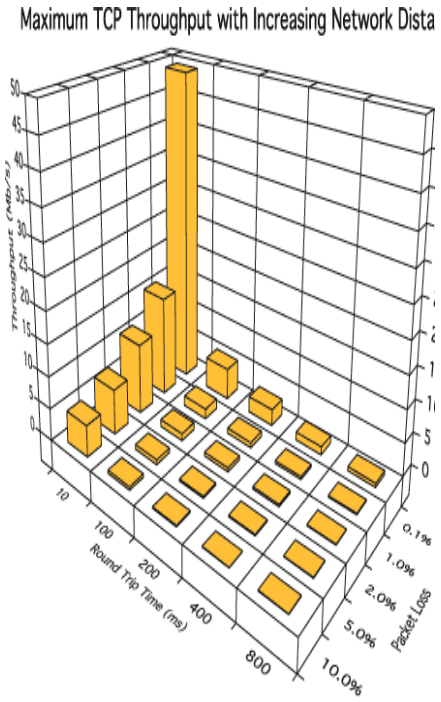


Fig 1: TCP throughput with distance

The source of the throughput bottleneck is the mechanism TCP uses to regulate its data flow rate. The TCP sender requires an acknowledgment of every packet from the TCP receiver in order to inject more data into the network. When an acknowledgment is missed, TCP assumes that it is overdriving the network capacity and enters an aggressive congestion avoidance algorithm. This algorithm reduces the data flow rate severely, and recovers the rate too slowly to keep modern pipes full. Even small variation in round-trip latency or bit errors due to the network media can cause TCP to enter congestion avoidance. Standard TCP is not equipped to distinguish these sources of packet loss from true link congestion. The consequences to file transfer are dramatic.

Consequences

TCP file transfers are slow and bandwidth utilization of single file transfers is poor. In local or campus area networks, where packet loss and latency are small but non-negligible (0.1%/10ms), the maximum TCP throughput is 50 Mbps. Typical file transfer rates are lower, 20-40 Mbps (with TCP stack tuning on the endpoints) on gigabit ethernet. Because standard

TCP halves its throughput in response to a single packet loss event, at high speeds, even a low loss percentage significantly lowers TCP throughput. Even with an abundance of bandwidth, transfer times are disappointing and expensive bandwidth is underutilized.

The bandwidth utilization problem

compounds on wide area links where increased network latency combines with packet loss. A typical FTP transfer across the United States has a maximum theoretical limit of 1.7 megabits per second (Mbps), the maximum throughput of a single TCP stream for 90ms latency and 1% loss, independent of link bandwidth. On typical intercontinental links or satellite networks, the effective file transfer throughput may be as low as 0.1% to 10% of available bandwidth. On a typical global link (3%/150ms), maximum TCP throughput degrades to 500-600 kilobits per second, 5% of a 10 Mbps link. Sometimes network engineers attempt to improve the throughput by "tuning" the operating system parameters used by the TCP networking stack on the file transfer endpoints or applying a TCP acceleration device. While this technique boosts throughput on clean networks, the improvement vanishes when real packet loss due to channel characteristics or network congestion increases.

TCP file transfers over difficult networks (with high packet loss or variable latency) are extremely slow and unreliable. TCP does not distinguish packet losses due to network congestion from normal latency variations or bit errors on some physical channels such as satellite links and wireless LANs, and severely self-throttles. FTP throughput over good satellite conditions is 100 kbps and degrades by more than half during high error periods such as rain fade. Large transfers can be extremely slow and may not complete.

TCP file transfer rates and times are unpredictable. As a window-based protocol, TCP can only determine its optimal rate through feedback from the network. TCP overdrives the network until packets are dropped by intervening routers, and in the best case, oscillates around its optimal rate, causing instabilities in the network for file transfer and other applications. Over commodity Internet links where traffic loads vary, file transfer rates may vary widely with network load. File transfers slow down and may exceed the allotted time window,. TCP acceleration devices may improve throughput and smooth the transfer rate when links are clean, but are also

window-based and subject to unpredictable back off.

Security, monitoring, and logging are deferred to external systems. In addition to suffering from TCP's throughput bottleneck and unpredictable flow rate, conventional file transfer protocols do not meet the security and manageability requirements of most business applications. FTP may require external security mechanisms to prevent content piracy or tampering. Also network performance details and transfer statistics are not available to the transfer operator for monitoring or billing.

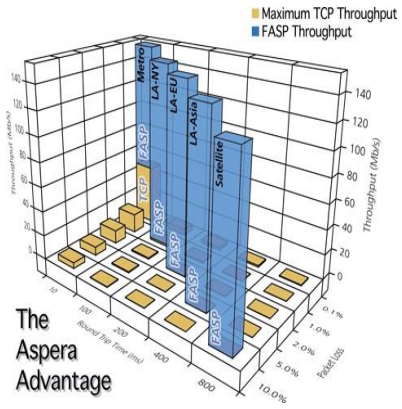
THE FASP SOLUTION

Aspera's file transfer products are built on *fasp* file transport, an innovative reliable transfer protocol designed to meet the speed, bandwidth control, and security requirements of business-critical file transfer over any IP network (LAN, WAN, satellite, and wireless).

Network independent speed

Unlike TCP throughput, *fasp* throughput is perfectly independent of network delay and robust to extreme packet loss. The bar graph below shows a comparison of throughput obtained by *fasp* versus the maximum throughput achievable for TCP-based file transfers on an OC-3 (155 Mbps) link under various latency and packet loss conditions. *fasp* transfer times are as fast as possible (up to 1,000x standard FTP) and highly predictable, regardless of network conditions. The maximum transfer speed is limited only by the resources (typically disk throughput) of the endpoint computers.

- **Delay and loss independent with near-zero overhead**. *fasp* uses standard UDP and achieves reliability in the application layer through a theoretically optimal approach that retransmits



precisely the real packet loss on the channel. At 10% packet loss, fasp achieves 90% of the targeted throughput with less than 1% redundant data overhead.

- **Single transport stream**. Unlike brute-force parallel TCP or similar protocols that divide a single file into constituent chunks and attempt to improve overall throughput through a parallel transmission, *fasp* achieves ideal efficiency with a single stream. In contrast, parallel TCP techniques exhaust system

resources, don't work for small files, reduce speed on packet loss, and hog bandwidth from single stream flows.

- **Perfect efficiency for large sets of small files**. Using a novel file streamlining technique, *fasp* achieves the same ideal efficiency for transfers of large numbers of small files. For example, one thousand 2 MB files can be transmitted from the US to New Zealand with an effective transfer speed of 155 Mbps, filling an entire OC-3. As a result, *fasp* enables replication of large data sets, regardless of file sizes, at line speed on even the longest, fastest wide area networks.
- **Distance neutral, fast, smooth progressive transfer**. *fasp* supports progressive-style transfers where incoming data is processed during the transfer, such as feeding a media player. The transfer speed is maximum (for the earliest possible start of playback), and will not change with variable network conditions, in contrast to progressive download applications using HTTP that are slow to start playback, and may be plagued by glitches and interruptions due to congestion on the link.

Automatic Adaptive Rate Control

While *fasp* can fill any available bandwidth, *fasp* also includes an intelligent adaptive rate control mechanism that allows transfer rates to throttle down for precision fairness to standard TCP traffic, but automatically ramps up to fully utilize unused bandwidth (see Figures 1 and 2).

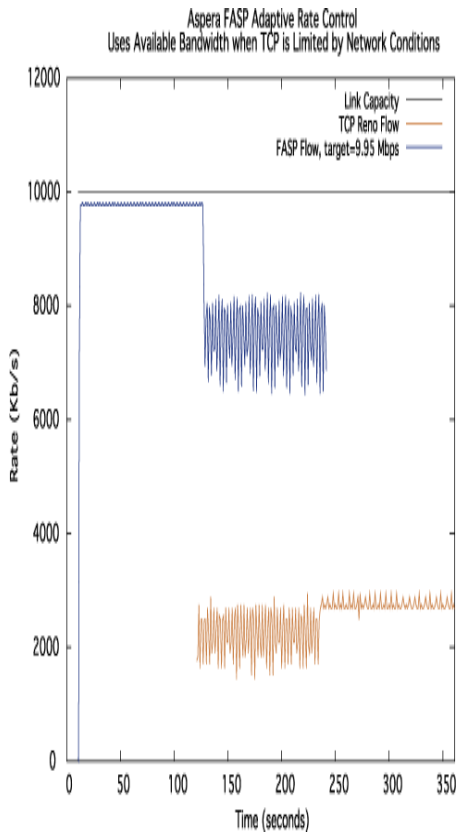


Fig 3: Efficiency of fasp

Single *fasp* and TCP flows sharing a 10 Mbps bottleneck bandwidth (100 ms RTT). A *fasp* flow is started with a target rate at link capacity (9.95 Mbps) and runs steadily. After 120 seconds, a TCP flow starts up and stabilizes at 2.3 Mbps, TCP's self-limited rate. *fasp* detects the presence of TCP traffic and immediately reduces its own rate to use the remaining available bandwidth (7.7 Mbps), without inhibiting TCP. When *fasp* finishes two minutes later, TCP continues at about the same rate. The above illustration shows that running a fasp connection and another TCP connection simultaneously does not affect the TCP connection adversely. Fasp does not hog the whole network bandwidth. It constantly monitors the available bandwidth and adapts to the prevailing network conditions. Thus it claims any bandwidth which is left unused but at the same time when congestion arises it brings its rates down so as to accommodate other connections thereby not overdriving the network.

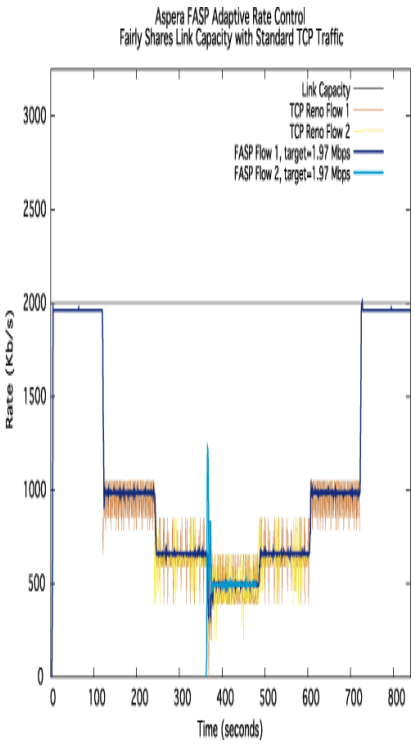


Fig 4: Fairness of fasp

Multiple *fasp* and TCP flows sharing a 2 Mbps bottleneck bandwidth (50 ms RTT). In this example a single *fasp* flow is started at the link capacity (1.97 Mbps). *fasp* 1 runs steadily at the target until the first TCP flow is started at two minutes. *fasp* 1 detects TCP 1 and immediately reduces rate to equally share the link at 1 Mbps. After two more minutes, a second TCP flow is started. Now *fasp* 1, TCP 1, and TCP 2 all equally share the link bandwidth at 660 kbps. Two minutes later, a second *fasp* flow starts up at a target of 1.97 Mbps. *fasp* 2 shoots up to the target but immediately detects the presence of the other flows and adjusts down as the other flows reapportion to equally share the link. Now all four flows are at 500 kbps. For the remainder of the test, one flow is terminated every two minutes. As each flow exits, the other flows reapportion the link bandwidth equally. The fasp flows zero in on the natural TCP rate at every interval, fairly sharing with TCP but with superior stability (less oscillation).

- **Automatic adaptation to the full, available bandwidth and fair**. The adaptive rate control algorithm is an original equation-based approach. When TCP's rate is self-limited on an uncongested link, *fasp* detects the unclaimed bandwidth and ramps up to fill it (Figure 1). When congestion builds up, *fasp* reduces rate to the TCP rate and equally shares the link with multiple TCP flows (Figure 2). This approach has fundamental benefits over the

window-based flow control algorithm used by standard TCP and even new accelerated or "high-speed" TCP's.

***fasp* adaptive rate control compared to TCP / Accelerated TCP:**

- **Loss tolerant**. Reacts only to true congestion, while remaining immune to inherent channel loss.
- **TCP fair**. Quickly stabilizes at a TCP-friendly rate when links are congested without squeezing small traffic.
- **Perfectly efficient**. Ramps up to fill unclaimed bandwidth, regardless of latency and packet loss.
- **Stable**. Zeroes in on the available bandwidth, and runs "flat" without oscillation, for stability.

Complete Security

The *fasp* protocol provides complete built-in security without compromising transfer speed. The security model, based solely on open standards cryptography, consists of secure authentication of the transfer endpoints using the standard secure shell (SSH), on-the-fly data encryption using strong cryptography (AES-128) for privacy of the transferred data, and an integrity verification per data block, to safeguard against man-in-the-middle and anonymous UDP attacks. The transfer preserves the native file system access control attributes between all supported operating systems, and is highly efficient: With encryption enabled, fasp achieves WAN file transfers of 40-80 Mbps on a laptop computer; 100-150 Mbps on a P4 or equivalent single processor machine; and 200-400 Mbps+ on dual-processor or duo-core workstations.

- **Secure endpoint authentication**. Each transfer session begins with the transfer endpoints performing a mutual authentication over a secure, encrypted channel, using SSH ("standard secure shell"). SSH authentication provides both interactive password login and public-key modes. Once SSH authentication has completed, the *fasp* transfer

endpoints generate random cryptographic keys to use for bulk data encryption, and exchange them over the secure SSH channel. These keys are not written to disk, and are discarded at the end of the transfer session.

- **On-the-fly data encryption**. Using the exchanged keys, each data block is encrypted on-the-fly before it goes on the wire. *fasp* uses a 128-bit AES cipher, re-initialized throughout the duration of the transfer using a standard CFB (cipher feedback) mode with a unique, secret nonce (or "initialization vector") for each block. CFB protects against all standard attacks based on sampling of encrypted data during long-running transfers.
- **Integrity verification**. *fasp* accumulates a cryptographic hashed checksum, also using 128-bit AES, for each datagram. The resulting message digest is appended to the secure datagram before it goes on the wire, and checked at the receiver to verify message integrity. This protects against both man-in-the-middle and re-play attacks, and also against anonymous UDP denial-of-service attacks.

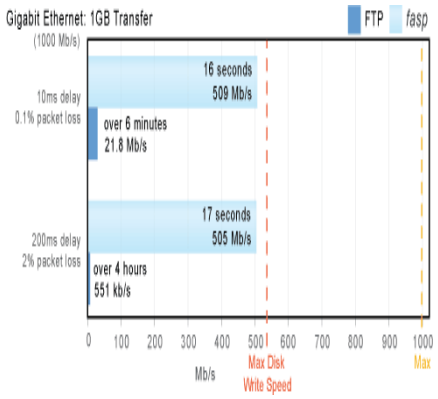
FASP VS FTP

The following graphs compare file transfer throughput and transfer time for fasp file transfer and FTP file transfer in typical network scenarios. All fasp and FTP benchmarks are for file transfer tests run within the Aspera labs. In each test, a 1-gigabyte file was transferred between commodity Pentium-4 computers running Debian Linux, using a standard Debian Linux implementation of FTP and Aspera Scp for fasp file transfer. A nistnet network emulator was used to simulate network round-trip latency and packet loss conditions typical on the Internet. Actual FTP throughputs will depend on the particular implementation of FTP used, the operating system, and the particular network loss pattern, but the results shown are typical.

***fasp* vs. FTP on gigabit metropolitan and wide area networks**

Conventional TCP file transfer technologies such as FTP dramatically reduce the data rate in response to any packet loss, and cannot maintain long-term throughputs at the capacity of high-speed links. For example, the maximum theoretical throughput for TCP-based file transfer under metropolitan area network conditions (0.1% packet loss and 10 ms RTT) is 50 megabits per second (Mbps), regardless of bandwidth. The effective FTP throughput is even less (22 Mbps). In contrast, *fasp* achieves 100% utilization of high-speed links with a single transfer stream.

Fig 5: Fasp vs FTP on WAN

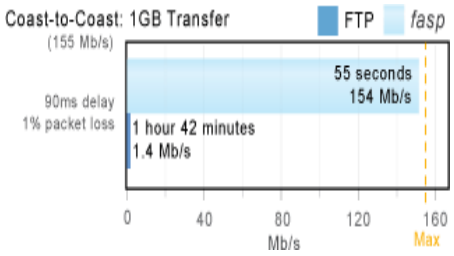


In the particular test shown, the *fasp* throughput on a gigabit ethernet MAN (509 Mbps) presses the disk read/write speed limits of the endpoint computers. Perhaps more important, *fasp* maintains this throughput even as latency and packet loss increase (505 Mbps at 200 ms/2%). FTP throughput degrades to about 550 Kbps under the same conditions. While this 1000X speed advantage over traditional TCP transfers is only evident on the fastest long-haul networks, it illustrates the difference in the *fasp* approach.

fasp vs. FTP on cross-continental links

Aspera fasp sustains the highest possible end-to-end file transfer rates on cross-continental and intercontinental file transfers where latencies are high and packet loss is variable. An FTP file transfer from LA to New York (90 ms) will achieve 5-6 Mbps when loss is low (0.1%). As congestion on the link increases (1%), FTP dramatically reduces its rate to 1.4 Mbps. In contrast, fasp transfers data at link capacity. On a 155 Mbps link with 90 ms/1%, fasp transfers at 154 Mbps, 100 times faster than FTP. Using a more typical 45 Mbps link, the transfer is still 30 times faster than FTP.

Fig 6: Fasp vs FTP on continental networks



fasp vs. FTP on intercontinental links

fasp's throughput advantage over FTP is more pronounced on an intercontinental transfer. At a packet loss rate of 2% and latency of 150ms, an FTP file transfer between continents runs at 700 kbps, and can drop to a crawl during periods of high congestion. fasp maintains stable throughput at link capacity in around-the-world file transfers. Using fasp file transport, a 1-gigabyte data transfer on a 10 Mbps link at 2% loss will run consistently at 9.9 Mbps, and

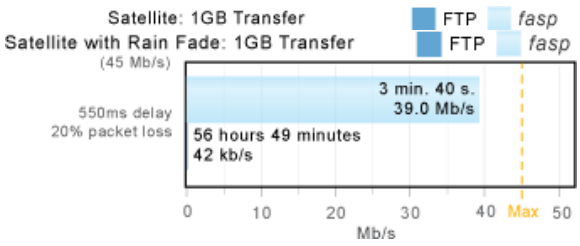
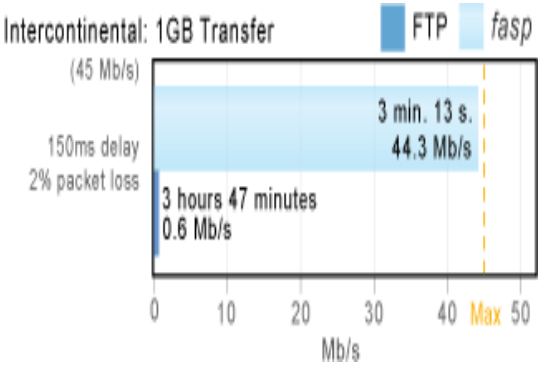


Fig 8: Fasp vs FTP over satellite

finish in under 15 minutes, regardless of distance. On a 45 Mbps link, the transfer finishes in

3.3 minutes. An FTP transfer under the same conditions takes several hours, and may terminate prematurely.



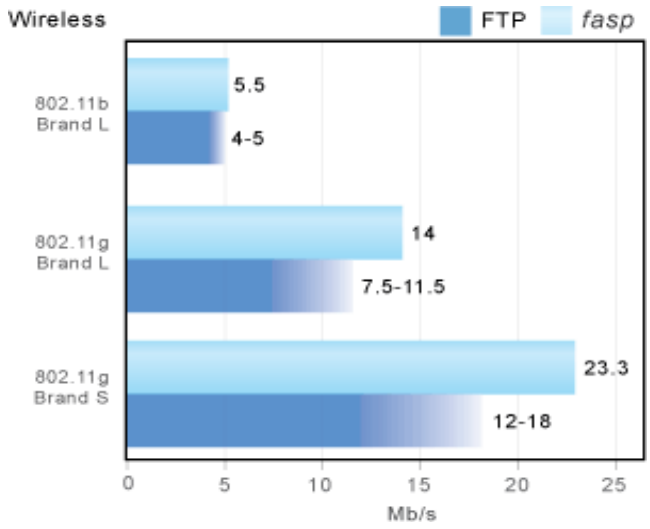
***fasp* over high-delay satellite links**

The high latency and bit error rates of satellite links severely hamper FTP file transfers, making large data set distribution or file upload over satellite impractical. fasp file transfer rates are immune to the distance and loss characteristic of single and multiple satellite hops. For example, a single fasp file transfer can fill a full transponder bandwidth (e.g. 45 Mbps) and will gracefully tolerate even the most extreme packet loss (over 30%), while FTP runs at 100 kbps or less and may not complete.

***fasp* vs. FTP on 802.11 Wireless**

802.11 Wi-Fi and fixed wireless networks are proliferating. However the promise of uploading/downloading personal data, large email attachments, or images from a mobile endpoint is limited today by the poor performance of TCP-based file transfers over wireless links. File transfers using conventional FTP or HTTP over wireless achieve a fraction of the specified capacity and are prone to wide variations in throughput and early terminations, depending on the link quality. Early experiments show that *fasp* improves the throughput of file transfers over standard 802.11 a/b/g connections by a factor 1.5 -2 and is more robust to the changes in link quality due to roaming or other traffic. As shown in the graph below, FTP throughputs and transfer times vary by +/-20% while *fasp* runs faster and at a steady rate. The *fasp* throughput and reliability advantage over standard file transfer may become even more necessary as wideband wireless enters the mainstream.

Fig 9: Fasp vs FTP over wireless network



FASP TRANSFER TIMES

Actual transfer time data by bandwidth, for typical network conditions

Table 4: Fasp Transfer Times

Metro Area Network

delay: 10ms, loss: 0.1%

	10 Mb/s	45 Mb/s	100 Mb/s	500 Mb/s
1GB	14m	3m 12s	1m 24s	16s 800ms
10GB	2h 18m	32m	14m	2m 48s
100GB	23h 24m	5h 18m	2h 18m	28m

High Speed WAN

delay: 200ms, loss: 2%

	10 Mb/s	45 Mb/s	100 Mb/s	500 Mb/s
1GB	14m 24s	3m 18s	1m 24s	17s 300ms
10GB	2h 24m	33m	14m 24s	2m 54s
100GB	24h 6m	5h 30m	2h 24m	28m 54s

Coast-to-Coast

delay: 90ms, loss: 1%

	10 Mb/s	45 Mb/s	100 Mb/s	500 Mb/s
1GB	14m 12s	3m 12s	1m 24s	17s
10GB	2h 24m	32m 24s	14m 12s	2m 48s
100GB	23h 36m	5h 24m	2h 24m	28m 18s

Intercontinental

delay: 150ms, loss 2%

	10 Mb/s	45 Mb/s	100 Mb/s	500 Mb/s
1GB	14m 24s	3m 18s	1m 24s	17s 300ms
10GB	2h 24m	33m	14m 24s	2m 54s
100GB	24h 6m	5h 30m	2h 24m	28m 54s

Satellite

delay: 550ms, loss 5%

	10 Mb/s	45 Mb/s	100 Mb/s	500 Mb/s
1GB	15m 12s	3m 30s	1m 30s	18s 200ms
10GB	2h 30m	34m 36s	15m 12s	3m
100GB	1d 1h 12m	5h 48m	2h 30m	30m 18s

CONCLUSION

The market need for a flexible, next generation file transfer technology is pervasive and exploding as file sizes increase, network capacities grow, and users increasingly exchange data and media. As an all-software platform deployable on any standard computing device, *fasp*™ is capable of filling the next-generation need, providing optimal data transfer in an application, over any Internet network, from consumer broadband to gigabit core.

As a result, fasp eliminates the fundamental bottlenecks of TCP and UDP based file transfer technologies such as FTP and UDT, and dramatically speeds transfers over public and private IP networks. Fasp removes the artificial bottlenecks caused by imperfect congestion control algorithms, packet losses, and the coupling between reliability and congestion control.

REFERENCES

http://www.asperasoft.com/en/technology_sections

- http://www.technologyreview.in/web/24526/
- http://nextbigfuture.com/2010/02/aspera-fasp-air-uploader-application.html
- http://en.wikipedia.org/wiki/Fast_And_Secure_Protocol
- Computer Networking, 5/e ,James F. Kurose ,Keith W. Ross ISBN: 0-13-607967-