Securing Grid Data Transfer Services with Active Network Portals

Onur Demir[1] **Michael R. Head**[2] Kanad Ghose[3]
Madhusudhan Govindaraju[4]

Department of Computer Science
Binghamton University (SUNY)
`{onur`[1]`, mike`[2]`, ghose`[3]`, mgovinda`[4]`}@cs.binghamton.edu`

The 8th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing 2007

*This work supported by in part by the NSF under award numbers EIA 9911099 and CNS 0454298

Outline

1. Motivation
   - Grid Data Transfer Services
   - Threats to GridFTP
   - Previous Work

2. Our Solution
   - Architecture/Implementation
   - Experiments
   - Results

Outline

1 Motivation
- Grid Data Transfer Services
- Threats to GridFTP
- Previous Work

2 Our Solution
- Architecture/Implementation
- Experiments
- Results

Requirements Driving Grid Data Transfer Services

- Higher bandwidth infrastructure
  - TCP and FTP as-is are unsuitable to connections with a high bandwidth delay product
  - Network bandwidth can outperform raw disk access
- Enormous data files
  - Output from high energy physics experiments
  - Large databases: protein sequencing databases, human genome project
- Need to authenticate and authorize in a globally scalable manner

GridFTP's Answers to Grid Data Requirements

GridFTP. . .

- Supports high bandwidth-delay-product infrastructure
  - Allows tunable TCP window sizes
  - Supports multiple parallel streams
- Supports striping to increase disk bandwidth
- Supports reliable resumption of canceled or dropped transfers
- Integrates with Globus's GSSAPI authentication

There are other solutions, but GridFTP is available everywhere

Outline

1 Motivation
- Grid Data Transfer Services
- Threats to GridFTP
- Previous Work

2 Our Solution
- Architecture/Implementation
- Experiments
- Results

Threats to Grid Services

Working assumption: As Grid services become more broadly available, they will increasingly become targeted. . .

- (Distributed) Denial of Service
- “Flash crowding” (not strictly an attack)
  - Services should prepare for near instantaneous explosions in (legitimate) activity
- Abusive users (with greater or lesser degrees of competence/intent)
  - This is hard to solve without service-specific solutions
  - Also ultimately requires some heuristics to classify abusers

Hypothesis: Prioritizing requests for different file sizes can improve performance for classes of users while maintaining overall throughput, even under attack.

Outline

1. Motivation
   - Grid Data Transfer Services
   - Threats to GridFTP
   - Previous Work

2. Our Solution
   - Architecture/Implementation
   - Experiments
   - Results

Previous Work

- A number of solutions supporting QoS and differentiated service involve packet inspection in the server
    - We propose inspection in an intelligent router
- A number of solutions specifically for DDoS exist
    - We also help protect against load attacks
- Our own Grid 2005 work
    - This work involves testing new policies and new results
    - Specifically, differentiating service for different classes of requests

Outline

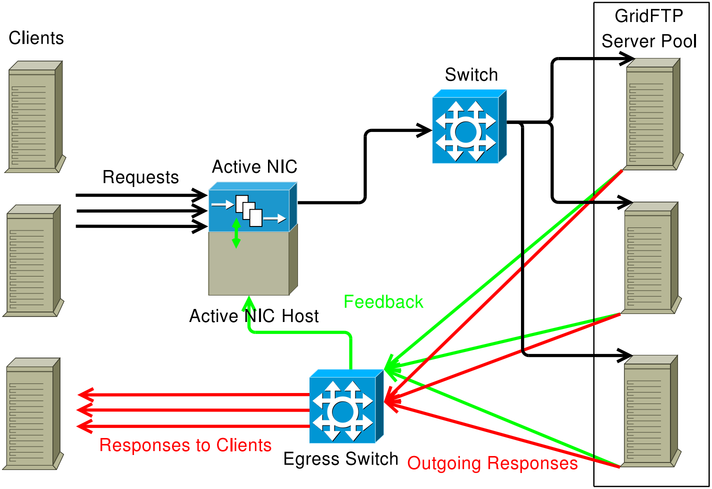
1. Motivation
   - Grid Data Transfer Services
   - Threats to GridFTP
   - Previous Work

2. Our Solution
   - Architecture/Implementation
   - Experiments
   - Results

Architectural Outline

- Connections from clients to GridFTP servers are mediated by an Active NIC (programmable gateway)
- Gateway examines packets and performs destination NAT and balances connections across the GridFTP server pool
- When attack conditions are detected may implement different policies to drop packets from clients or distribute connections differently

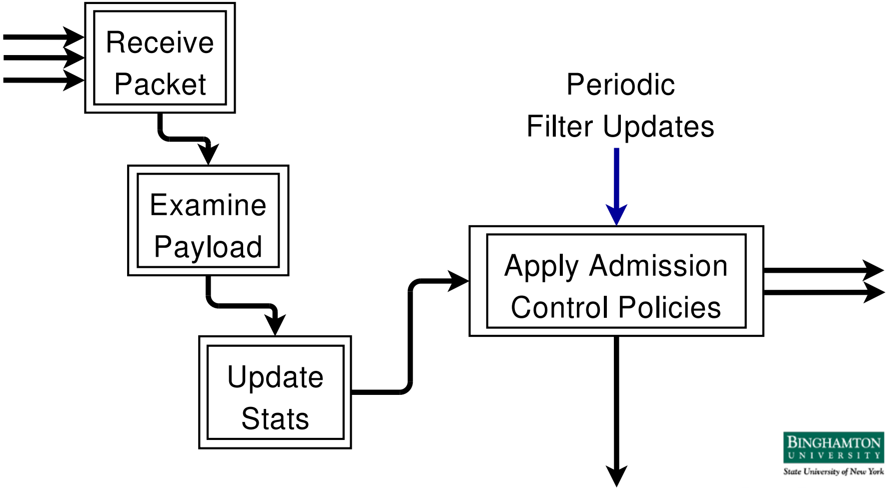
Server Architecture



Component Responsibilities

- Active NIC
    - Receive packets, rewrite them for the servers
    - Implement filtering and load balancing rules
- Active NIC Host
    - Collect load information from servers
    - Implement policy by transforming load data into filtering rules for the NIC
    - Upload rule updates to the NIC
- Grid FTP Servers
    - Implement the GridFTP protocol
    - Transfer load information to the Active NICÂ Host
    - Collect information about known good clients

Processing Logic Inside the Gateway



Client Categories

We define three classes of clients...

- **Green Addresses**: Clients currently engaged in legitimate service use
- **Red Addresses**: Clients that are unknown to the server in recent time
- **Preferred Addresses**: A pre-configured set of known good addresses

Requirements for the Gateway Device

- Packet processing rate should match the servers' packet handling capacity
- Low latency link to servers
- Must react quickly in response to attack

We've chosen a Ramix PMC 694 PCI card:

- Dual 100Mbit Ethernet ports
- Two autonomous DMA controllers
- 233Mhz PowerPC CPU w/32MB of RAM

Gateway Performance Issues Addressed

- Packet filter is BPF+ with some optimizations for the PowerPC data cache
  - Implemented in the TCP/IP stack atop the IP layer
- Packets are forwarded between interfaces with a zero-copy regime
- Load balancing is round-robin by default, but may change based on policy
- Packets are read in a (fast) polling loop (not interrupt driven)
- Trie data structure is used for efficient IP address lookup

Policies Examined

We examined the results of privileging certain classes of requests during attack

- *Small Requests Favored*: Clients requesting 48KB size files have priority
- *Medium Requests Favored*: . . . 2MB . . .
- *Large Requests Favored*: . . . 64MB . . .

Note: $(64MB * 10^6)/(10^8 bps/8)$

$=$5.120s to transfer 64MB over a 100Megabit link

Implications

- It is possible to favor a file class by modifying thresholds.
- Depending on the file type character of the server, completion rate of transfer can be improved.
  - Small files can be favored for servers that hold source code, small images
  - Large files can be favored for servers that hold multimedia files, large data files, . . .
- The system itself is dynamic and allows custom policies.
- The server feedback support provides implementation of new policies based on other application level criteria.

Outline

1. Motivation
   - Grid Data Transfer Services
   - Threats to GridFTP
   - Previous Work
2. Our Solution
   - Architecture/Implementation
   - Experiments
   - Results

Experimental Design

Five cases . . .

1. Base (“NORMAL”) case
2. Attack, no policy
3. Attack, small favored
4. Attack, medium favored
5. Attack, large favored

Note: plotting the results of running a completely unprotected server pool is uninteresting

Test cases

Comment this out before talk

- “Attack” means a simulated DDoS attack is under way
- Base case involves running the clients when the server is not under attack
- “ATTACK, no policy” means an attack is under way, and the server is protected with no particular favoritism policy (equivalent to Grid2005 protection)
- “ATTACK, small favored” means that requests for medium and large files are limited to a quota – similarly for the other two cases
- The quota is determined by making an increasing number of concurrent requests for each file class and waiting until clients start experiencing connection failures. We calculate the number of requests per second at this point, multiple that rate by K=2, then apply that rate to some time window size. The gateway then drops new connections for a given class when the incoming requests exceed the quota for its quota.

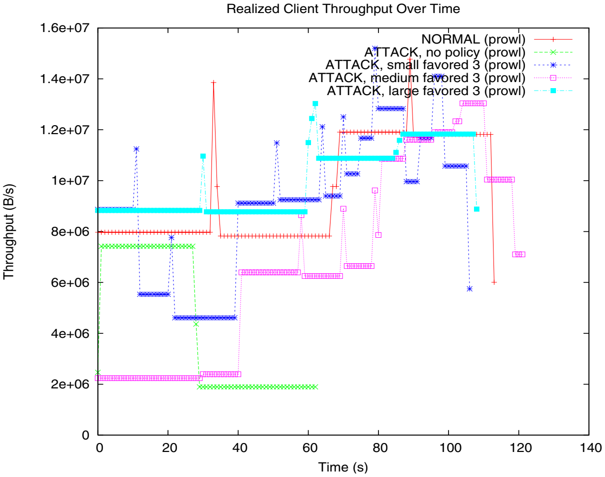
Test Scripts

- Each script instance repeatedly requests file several time (using `globus-url-copy`)
  - Uses “Extended Block Mode” and four parallel streams
- Scripts requesting a given file size all run on a specific client machine
- 20-50 script instances per file size class

Outline

1. Motivation
   - Grid Data Transfer Services
   - Threats to GridFTP
   - Previous Work
2. Our Solution
   - Architecture/Implementation
   - Experiments
   - Results

Effect of Policies on Large File Class

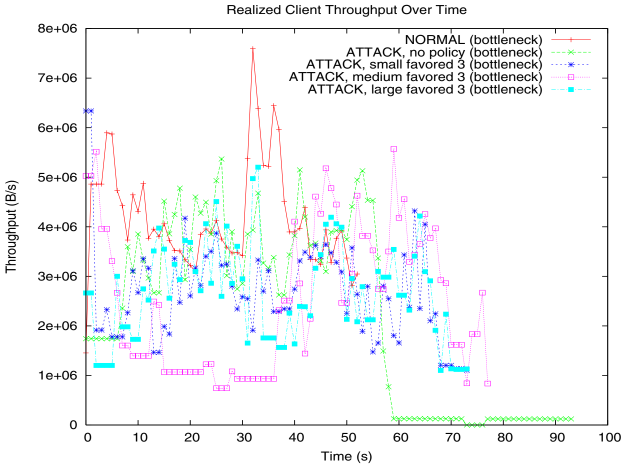


Talking Points on Large File Clients

comment out this slide

1. This graph should be trimmed to the first 60 seconds.
2. Note that when the large file class is favored, it overcomes even the normal case
3. When smaller files are favored (blue line), large files do better than when no defense policies is in place
4. Why is the pink line so bad?

Effect of Policies on Medium File Class

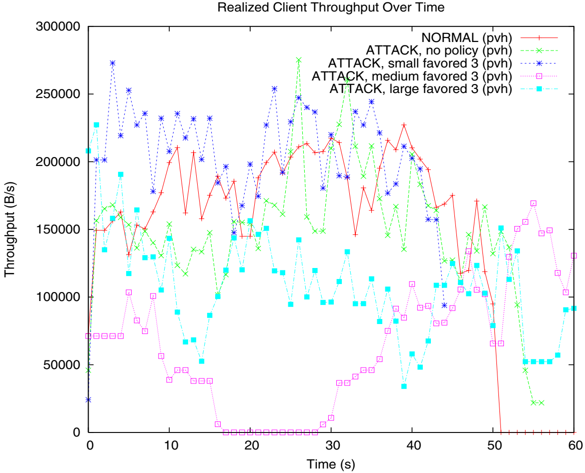


Talking Points on Medium File Clients

comment out this slide

1. Umm... why is the pink line low here?
2. Also, as in the large file cas, only the first 60 seconds of the plot is interesting
3. There must have been some problem with either the medium favored run or the in the analysis scripts!
4. Consider tossing this slide or use it to show there was a problem with the experiment?

Effect of Policies on Small File Clients

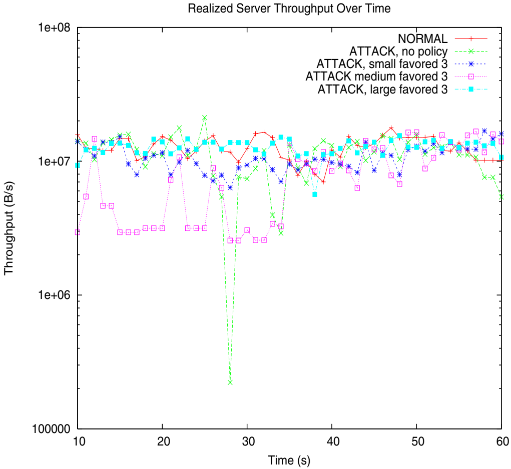


Talking Points on the Small File Clients

comment out this slide

1. Again, pink has problems???
2. Note that as with the large file class, the small file clients perform better than under the normal case
3. Under attack with no policy beats the cases when medium and large file clients are preferred. Recall that the overall throughput was not hurt and was more consistent overall, though.

Overall Server Throughput



Discussion Points on Server Throughput

comment out this slide

- Each line represents the overall server throughput as seen by the clients (the sum of all data transferred to all clients)
    - The client-realized throughput for a given client is calculated by noting the start and top times for each file transferred and then dividing the size by the time taken. The average is entered for each second of the experiment. This leads to some noisiness in the data due to the possibility for two consecutive transfers in a given script stopping and starting in the same second since one will have their averages summed for that second – this is needed to compute the overall server throughput for each second.
- Server throughput is overall more consistent with any of the three policies in place than with no policy in place
    - The dip in the pink line is due to the a large file transfer having a particularly low average transfer rate

Limitations

- Currently uses static knowledge about the requests to determine the size of the file associated with the request
    - May need to decrypt the control stream in the Active NIC (expensive operation)
- Data about ongoing connections are from GridFTP logs
    - May need to write a GridFTP module or otherwise modify GridFTP to provide more detailed connection information
- Constants for the quota multiplier and window size were determined experimentally
- Only shapes incoming packets – outgoing packets (and GridFTP data connections) go through separate egress switch

Summary

- Adaptive traffic management improves server throughput
- Active NIC based gateway serves as an unobtrusive mechanism for classifying requests and shaping incoming traffic
- Implements fast response to attacks
- Careful choice of thresholds essential
  - Identify through experimentation