Effects of Processing Delay on Function-Parallel Firewalls

Ryan J. Farley and Errin W. Fulp

WAKE FOREST UNIVERSITY
Computer Science
Network Security Group
nsg.cs.wfu.edu

US Department of Energy
Office of Science
MISC Division

IASTED PDCN
February 15, 2006

Abstract

- Firewalls filter packets between networks.
- Unfortunately, they introduce significant delay to a system.
- Given issues with current high speed networks, how will firewalls cope with future networks?
- This presentation will introduce a parallel firewall system that can:
  - Maintain integrity of original system.
  - Mitigate Denial of Service.
  - Provide High Scalability.
  - Maintain Quality of Service.

Modeling Precedence

- A **rule** is an ordered tuple and an associated action.
- A **policy** is an ordered set of rules.
- In a Policy DAG Vertices are rules, edges are **precedence relationships**.
  - Rules intersect if their every tuple of their set intersection is non-empty.
  - Edge exists between r_i and r_j, if $i < j$ and the rules intersect.
- If two rules intersect, then the order is significant.

Accept Sets

- An **accept set** *A* is the set of all possible unique packets which a policy will accept.
- A **deny set** *D* is the set of all possible unique packets which a policy will deny.

- A **comprehensive** policy *R* is one where $\overline{D} = A$.
- R and R′ are **equivalent** if $A = A'$.
- If *R′* is a modified *R* then **integrity** is maintained.

Data Parallel

- A system is Data parallel (load-balancing) if:
  - Distributes packets evenly to all firewall nodes.
  - Duplicates original policy to each firewall node ($R_i = R$)
- Maintains integrity since $A_i = A$.
- Better throughput than traditional designs.
- Does not allow for Quality of Service or state.
- Benefit is related to load, when enough traffic exists to split.
- Does not directly focus on reducing processing delay.

Function Parallel with Gate

- A system is Function parallel (with gate) if:
  - Duplicates packets to all firewall nodes.
  - Distributes local policy R_i to each firewall node, where

$$\bigcup_{i=1}^{m} A_i = A$$

- A **gate** coordinates local policy results.
- Incoming packets are also duplicated to the gate.
- Multiple nodes may find an accept match for the same packet if: $A_i \cap A_j, i \neq j$
- A gate node is needed to preserve precedence.

Function Parallel with no Gate

- If the nodes could be designed to act independently then the gate could be removed.
- A system is Function parallel, and does not require a gate if:
  - Duplicates packets to all firewall nodes.
  - Distributes a local policy R_i to each node, where both

$$\bigcup_{i=1}^{m} A_i = A \qquad \bigcap_{i=1}^{m} A_i = \emptyset$$

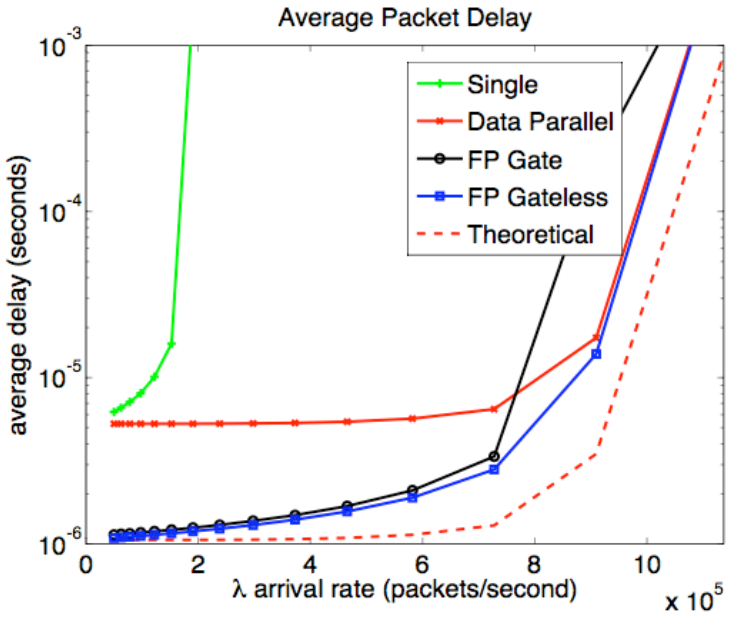
- Since no accept sets intersect, only one node will find an accepting match.

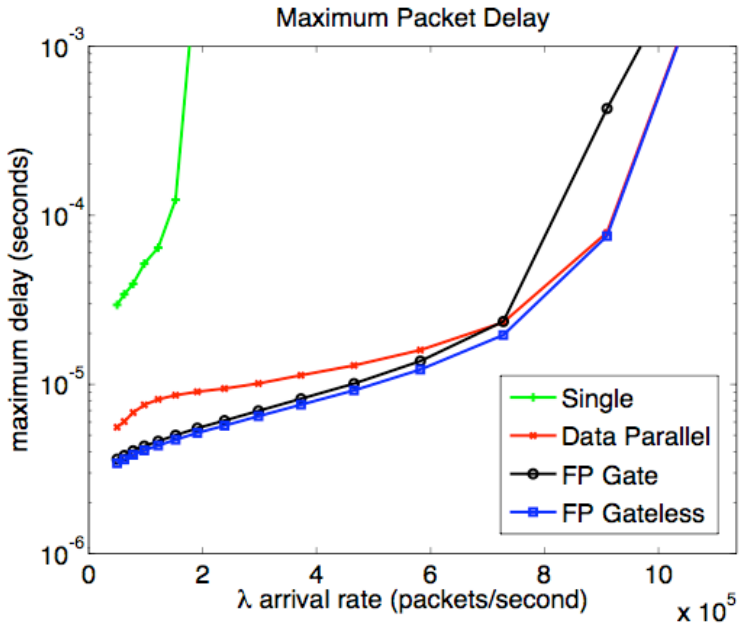
Simulation Comparison

- Assumptions:
  - Each node could process 6 x 10^7 rules per second.
  - Inter-arrival rate scheduled on **Poisson** distribution.
  - Rule match probability according to **Zipf** distribution.
  - No additional delay for Data Parallel packet distribution.
  - Constant gate delay for Function Parallel with Gate
- Cases were ran to determine the performance of:
  - Increasing arrival rates.
  - Increasing policy size.
  - Increasing number of nodes.

Delay vs Arrival Rate

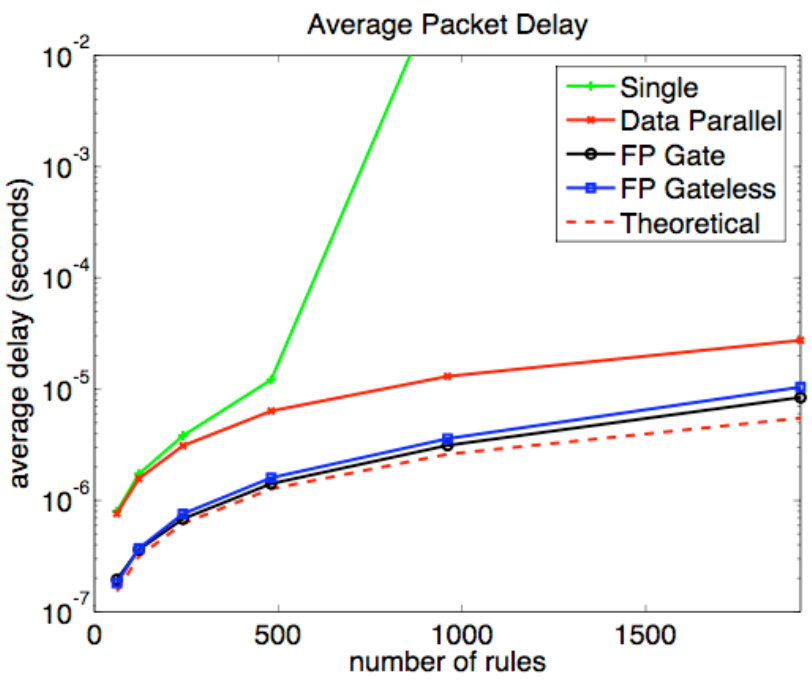
- Parallel systems consisted of 5 nodes.
- Policy size was 1024 rules.
- Arrival rate was varied from 300 Mbps up to 6 Gbps.

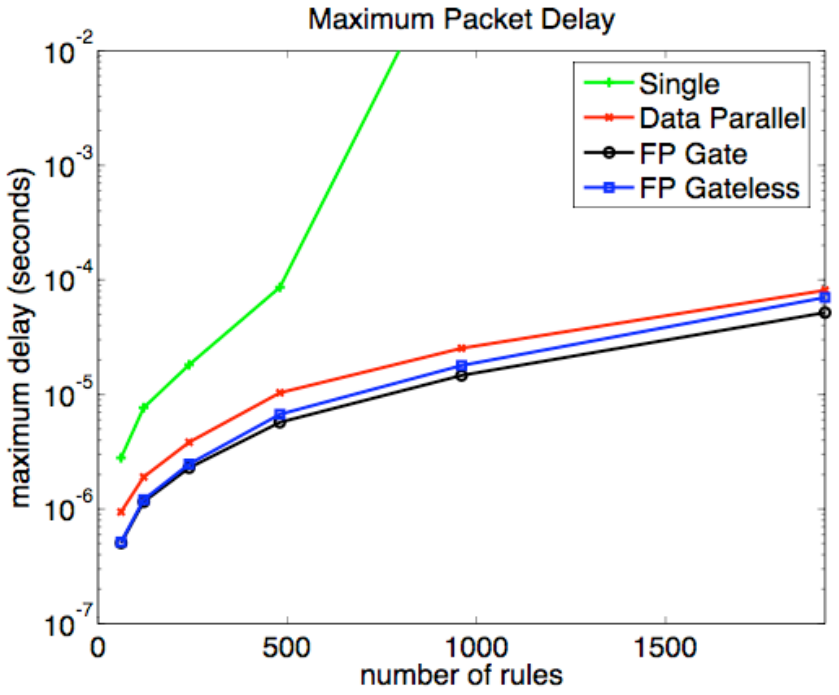




Delay vs Policy Size

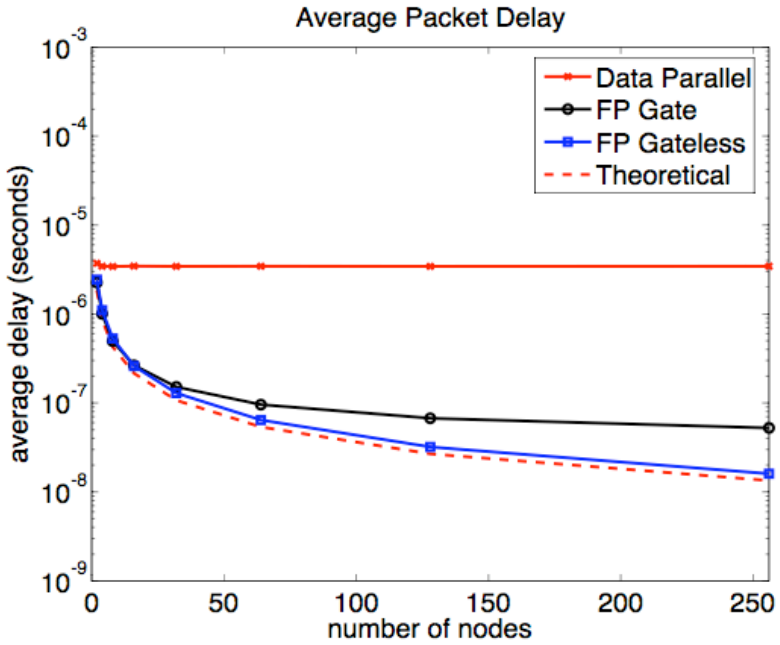
- Parallel systems consisted of 5 nodes.
- Arrival rate was established at 650 Mbps.
- Policy size was incremented from 2 to 2048.

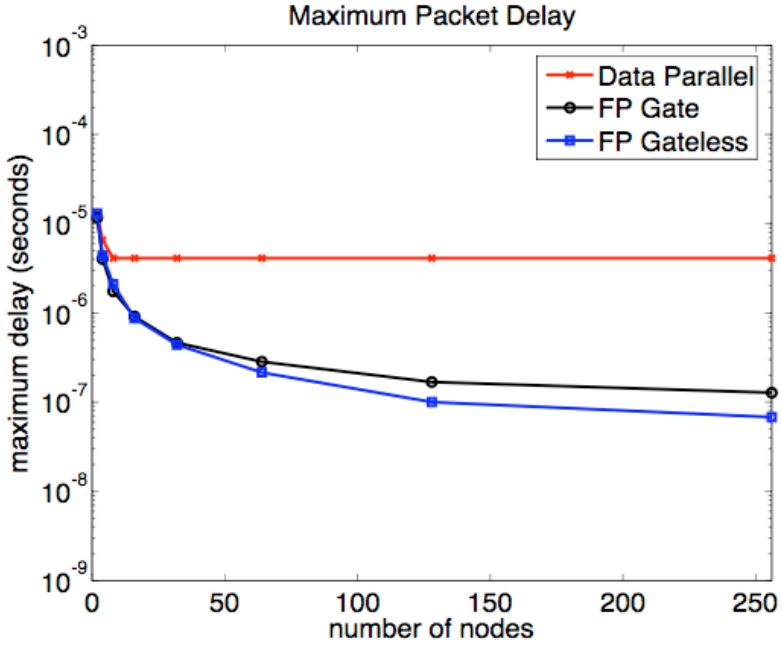




Delay vs Number of Nodes

- Arrival rate was established at 650 Mbps.
- Policy size was 1024 rules.
- Parallel systems varied number of nodes from 2 to 256.





Summary of Simulations

- Illustrates advantage of parallelism.
- Reducing **processing time** is more advantageous than reducing **arriving traffic load**.
- Removing the gate delay helps function parallel approach theoretical rates.

Conclusions

- It is important that a firewall acts transparently to users.
- Unfortunately, firewalls quickly become bottlenecks.
- Particularly in **High Speed Networks**.
- Improving implementations and hardware is not as scalable as needed.
- Enter **Parallel firewalls**.
- **Data parallel** does not address processing delay.
- **Function parallel with gate** is flexible, but has the added gate delay.
- **Function parallel** with no gate solves scalable processing delay issues.

Future Direction of Work

- Extend rule distribution and optimization methods for Function parallel with no Gate.
- Incorporate Distributed IDS/IPS.

- New Start-up company
  - Great Wall Systems. Winston-Salem, NC, USA.
  - Basis is two patents created through research from DOE grant.
  - Dedicated to High Speed Networking Devices for IDS/IPS systems.

Candidate for Parallelism

- Several solutions for improving firewall performance:
  - Optimize algorithms.
  - Optimize rules.
  - Parallelize system.
- Improvements to the single firewall design are temporary.

- Can divide load two ways:
  - Data Parallel - divide data processed.
  - Function Parallel - divide work of processing.

How the Gate Works

- Firewall nodes do not execute an action.
  - Send decision as a **vote** to the gate.
  - Vote consists of at least the rule number and action.
    - **No match** is a valid response.
    - Matches in state would have uniformally lower values.
- The gate caches the packet until a decision can be made.
- First match method is accomplished by executing the action of the vote with the lowest rule number.

Other Considerations

- Redundancy can be provided as long as accept sets are not violated.
- Gate can use knowledge of DAG to remove necessity of some votes.
- Processing the traffic asynchronously would increase work efficiency.
- Removal of need for the gate would eliminate associated processing delay.

Theoretical Comparison

- Standard formula for delay of a cascading system is:

$$E(T) = \sum_{i=1}^{q} \frac{1}{\mu_i - \lambda_i}$$

- Data parallel is:

$$E_d(T) = \frac{1}{\frac{x}{n} - \frac{\lambda}{m}}$$

- Function parallel is:

$$E_f(T) = \frac{1}{\frac{m \cdot x}{n} - \lambda}$$

- Relationship of delay is:

$$\frac{E_f(T)}{E_d(T)} = \frac{1}{m}$$