

Optimisation of Policy-Based Internet Routing using Access-Control Lists

V. Grout, J. McGinn
Centre for Applied Internet Research (CAIR)
University of Wales, NEWI Plas Coch Campus
Mold Road, Wrexham, LL11 2AW
UK
{v.grout, j.mcgin}@newi.ac.uk

Abstract

This paper considers an optimisation problem encountered in the implementation of traffic policies on network routers, that of attempting to minimise the time taken to process a sequence of rules in an Access Control List (ACL). The problem is formulated and shown to be NP-complete. Exact and heuristic solution methods are introduced and compared and computational results given. Additional complications and extensions are considered in conclusion. The full paper can be found in [1]

Keywords

Traffic packet policies, Access Control Lists (ACLs), Rule order
Optimisation, Approximation algorithms

1. Introduction: Access Control Lists

This paper considers packet-based traffic filters in the form of *Access Control Lists (ACLs)*. ACLs are used for a variety of purposes in applying security and other policies on routers and other key network devices. An ACL is an ordered sequence of *rules*, each rule seeking to permit or deny any packet that it matches. A typical rule, using Cisco IOS notation [2] might be

```
access-list 101 deny icmp any 10.0.0.0 0.255.255.255 echo-reply
```

with each rule attempting to match an incoming packet for source or destination addresses, protocols and/or other characteristics. The packet will either be permitted or denied by the first rule it matches with a final, implicit `deny all` rule rejecting packets not otherwise matched. Two rules are *dependent* if it is possible for a single packet to match both rules. The order of dependent rules must be preserved. With some rules taking longer to process than others (due to more complex characteristics) and some matching more packets (for the current traffic flow), there will be an

optimal ordering of the rules of an ACL to minimise expected processing time, subject to the order constraints of dependent rules.

2. The Problem and its Complexity

In [1] a formal development is given of a *list* L , implementing a *policy* Z in *traffic* T . If two rules, i and j , are dependent then set $d_{ij}=1$; otherwise $d_{ij}=0$. Defining the *latency*, $\lambda(r_i)$, of a rule r_i to be the time taken to (independently) process r_i , the *cumulative latency*, $\kappa(r_i(L))$, of r_i at position i in a list L , is the time taken to process r_i and all rules preceding it in L .

$$\kappa(r_i(L)) = \sum_{\varphi=1}^i \lambda(r_{\varphi}(L)). \quad (1)$$

The *expected latency*, $E(L,T)$, of a list L , in traffic T , is then given by

$$E(L,T) = \sum_{i=1}^n h(r_i(L),T) \kappa(r_i(L)) = \sum_{i=1}^n h(r_i(L),T) \sum_{\varphi=1}^i \lambda(r_{\varphi}(L)). \quad (2)$$

For a given traffic flow, T , we require to find (or approximate) the list, L , implementing a policy, Z , that minimises $E(L,T)$. The following theorem is given.

THEOREM: SEQUENCING TO MINIMISE EXPECTED LATENCY (SMEL) is *NP-complete*

PROOF: Transformation to SEQUENCING TO MINIMIZE WEIGHTED COMPLETION TIME (SMWCT) [3].

A direct mapping from SMEL to SMWCT is achieved by setting

<u>SMEL</u>		<u>SMWCT</u>
Z	to	N
r	to	t
D	to	$<$ by taking $t_i < t_j \Leftrightarrow (i < j) \wedge d_{ij} = 1$
$\lambda(r)$	to	$l(t)$
$h(r)$	to	$w(t)$

for any given flow, T (using the notation from [1] and [4]). \square

It follows that (unless $P=NP$) guaranteed exact solutions are not reasonably to be expected for large values of n .

3. Algorithms and Results

Optimisation algorithms may be divided broadly into two classes: exact and approximate. The paper discusses exact solutions to SMEL using exhaustive search, dynamic programming, linear programming and branch and bound techniques. Such methods are appropriate only for small ACLs or, using more powerful processors than generally available on network routers, as benchmarks with which to compare approximate solutions. The paper then proceeds to describe heuristic approaches, in particular the *k-Opt* [5] and Lin & Kernighan [6] algorithms. These are both examples of *local search* optimisation methods in which small changes, *perturbations*, are applied to a current solution in a search for improvement [7]. In this case, the improvement will be a decrease in expected latency and the natural initial solution to which to apply the first set of perturbations will be the policy list as set by the network administrator. Such approximations will be necessary for larger lists running on production routers in live networks. The paper compares results of simulated test instances of 10, 25, 50 and 100 rule ACLs solved firstly exactly, then using a constrained 2-Opt and finally the commonest version of Lin & Kernighan [6]. It is noted from these results that:

- optimisation is extremely worthwhile since significant improvements in expected latency (approaching 50% for $n=100$) are to be achieved over non-optimised lists,
- a simple 2-Opt approach gives excellent results (typically within 5% of optimality for $n=100$) and runs quickly, and
- the more sophisticated LK method gives marginally better results but at the expense of a significant increase in run time (approximately 50%) more than for 2-Opt.

It is therefore concluded that, for production routers in environments where minimal latency is a high priority, the simple 2-Opt is the most appropriate solution to adopt. 2-Opt provides effective solutions quickly whereas any more sophisticated method is likely to add to, rather than reduce, packet latency.

4. Further Conclusions and Future Work

The paper finishes by considering extensions and variants, in particular:

- the need to accommodate traffic shaping, queuing and prioritisation, and
- issues relating to variable traffic flows and timing.

In particular, it asks:

- As the hit rate of a rule is known only for its current position, how can the gain in expected latency be calculated for a potential rule swap?
- How frequently should (re-)optimisation of the ACL rule order be performed?

A complete solution to these problems, in the form of a real-time simplification to 2-Opt, is given in [8], where the significance of related work, such as [9], [10], [11], [12], [13] & [14], is also discussed.

References

- [1] V. Grout & J. McGinn, "Optimisation of Policy-Based Internet Routing using Access Control Lists," <http://www.newi.ac.uk/groutv/Papers/ACLs/ACLs.pdf>. May 2005.
- [2] A. Colton, "*Cisco IOS for IP Routing*", Rocket Science Press, 2002.
- [3] E.L. Lawler, "Sequencing Jobs to Minimize Total Weighted Completion Time Subject to Precedence Constraints," *Ann. Discrete Maths*, Vol 2, pp75-90, 1978.
- [4] M.R.Garey & D.S. Johnson, "*Computers and Intractability: A Guide to the Theory of NP-Completeness*," W.H. Freeman, New York, 1979.
- [5] C. Rego, "Local Search and Metaheuristics" in *The Traveling Salesman Problem and its Variations*, G. Gutin & A. Punnen (eds.), Kluwer Academic, 2002.
- [6] S. Lin & B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem", *Operations Research*, Vol. 21, pp972-989.
- [7] E. Aarts & J.K. Lenstra, "*Local Search in Combinatorial Optimisation*", Princeton University Pres, 2003.
- [8] V. Grout & J. McGinn, "Reducing Processing Latency in Network Packet Filters", *Proc. 5th International Network Conference (INC 2005)*, Samos Island, Greece, 2005 (to appear).
- [9] E. Al-Shaer & H. Hamed, "Modeling and Management of Firewall Policies", *IEEE Trans. on Network and Service Management*, Vol. 1-1, 2004.
- [10] Cisco, "*ACL Optimizer and Hits Optimizer*", Cisco Systems, 2003.
- [11] B. Hari, S. Suri & G. Parulkar, "Detecting and Resolving Packet Filter Conflicts", *Proc. 19th Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM00)*, 2000, pp1203-1212.
- [12] A. Mierluti, "A Rule Cache for IPTables in Linux", *Proc. RoEduNet International Conf.*, Iasi, Romania, 2003, pp108-114.
- [13] F. Bukhatwa, "High Cost Elimination Method for Best Class Permutation in Access Lists" *Proc. IADIS WWW/Internet International Conference (W3I 2004)*, Madrid, Spain, 6th-9th October 2004, pp287-294.
- [14] I. Stoica, "Rook Lookup and Packet Classification, *Tech. Rep. no. CS268*, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, USA, 2001.

Biographies

Vic Grout is a Reader in Computer Science and leads the Centre for Applied Internet Research (CAIR) at the University of Wales, Wrexham, UK, directing research into network algorithms. John McGinn is a Lecturer in Computing in CAIR at the University of Wales, Wrexham, UK, researching visualisation of intrusion detection.