

A New Distributed Link-State Routing Protocol with Enhanced Traffic Load Balancing

Vic Grout, John Davies, Mike Hughes and Nigel Houlden

Centre for Applied Internet Research, University of Wales, NEWI Wrexham, UK
e-mail: v.grout@newi.ac.uk

Abstract

This paper details proposals for an improved link-state interior routing protocol. The advantages and disadvantages of OSPF, a dominant existing protocol, are discussed and enhancements suggested in outline form. A guiding theory of optimal routing is then developed to which practical constraints are introduced. A two-part, distributed algorithm is presented providing network partitioning and traffic-balanced path calculation. Results, limitations and possibilities for future work are discussed in conclusion.

Keywords

Routing protocols, Optimisation, Partitioning, Path determination

1. Introduction: Interior Link-State Routing and OSPF

Routing protocols in networks or internetworks allow routers to build routing tables. These routing tables permit *routed* protocols such as the Internet Protocol (IP) to function. Routing protocols may be classified as *interior* or *exterior*. Exterior Routing Protocols (ERPs), acting between domains, are essentially configurable rules of precedence defining traffic policy at border routers. This paper is concerned with the more varied Interior Routing Protocols (IRPs), operating within a domain, on a network under a common administration.

There is a broad division of IRPs into Distance-Vector Routing Protocols (DVRPs) and Link-State Routing Protocols (LSRPs). DVRPs, though easy to configure, are generally limited in scope, inefficient and slow to converge across a network to a stable state (Slattery and Burton, 2000). This paper discusses the preferred LSRP approach in which routers have a more complete view of the topological state of the network. In particular it uses, as a foundation, the Open Shortest Path First (OSPF) protocol currently in widespread use (Moy, 2000). The implementation and operation of OSPF for a large domain may be summarised as follows:

1. *Partitioning*: The Network Administrator (NA) partitions the domain into a number of areas. One of these areas must be the backbone, to which all other areas are adjacent.
2. *Information Exchange*: Routers run OSPF to exchange link-state information regarding the current topology of the network. Information is passed in full among routers in the same area but in summarised form only between routers in adjacent areas.

3. *Path Determination:* OSPF routers run Dijkstra's Shortest Path Algorithm (DSPA) (Dijkstra, 1959) to find shortest paths to all other networks. The first step in each path becomes the routing table entry for that network.

Steps 2 and 3 are implemented continuously as the network reacts to link-state changes. Figure 1 gives an example of the general structure/hierarchy. Advantages of OSPF's hierarchical area structure are:

- DSPA is of polynomial complexity and optimal on a pairwise basis.
- The summarised updates between areas reduce routing traffic in the domain.
- DSPA needs to be run less frequently as minor updates are contained within areas.
- Route summaries result in smaller routing tables, which require less processing.

However, OSPF has the following shortcomings:

- The NA's task of configuring areas is notoriously difficult and prone to error.
- These manually configured areas will be sub-optimal in that they are unlikely to take into account the underlying network topology.
- DSPA produces optimal paths between networks in isolation only. When traffic following multiple paths is overlaid on common links across the domain, the resultant routing strategy can be far from optimal.

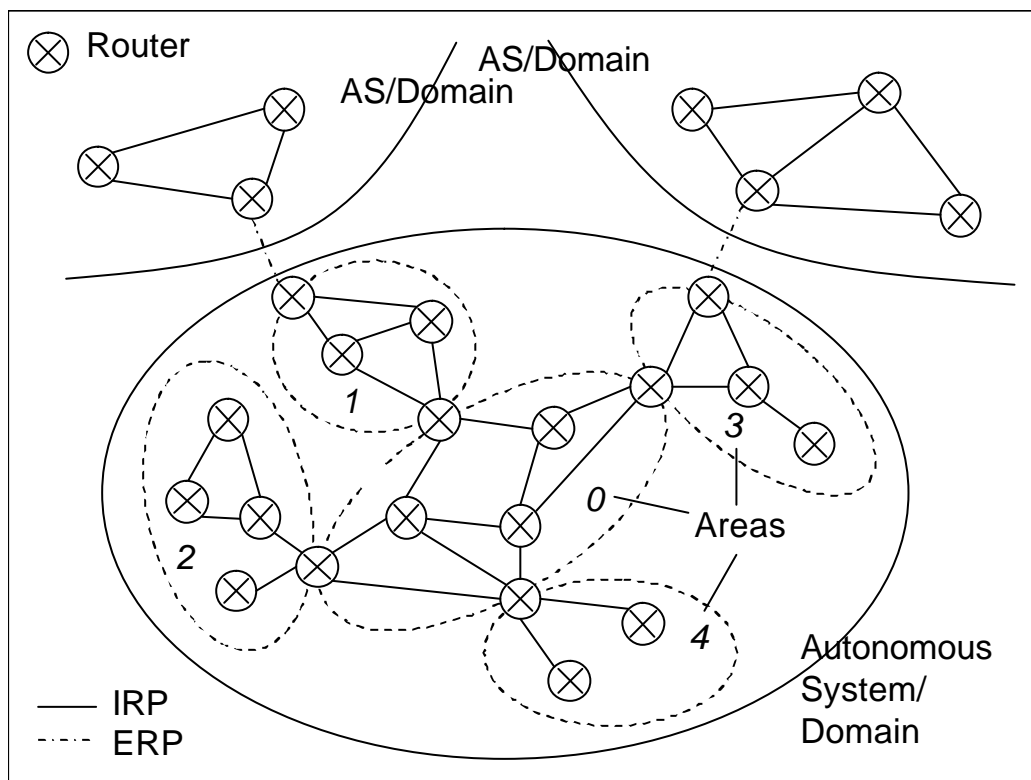


Figure 1: The routing model

This paper proposes two key extensions to OSPF in the areas of partitioning and path determination, namely:

- Automatic, and optimal, derivation of areas.
- Improved optimality of the overall domain routing.

These issues of partitioning and path determination have been addressed, separately and mainly in theory only, and simple algorithms given, in recent months (Grout, 2003 & 2004). This paper presents a unified protocol and deals with the remaining practical issues outstanding from the original work.

2. An Ideal Routing Protocol

The essential problem in the operation most non-trivial routing algorithms is that of determining the *cost* of a network connection. OSPF costs are configurable but the default standard, and by far the commonest implementation, is to take the cost of a link to be inversely proportional to its speed. Precisely, $c = 10^8 / b$, where b is line speed in bits per second. As a measure of adjacency, this works well: two routers connected by a fast link may be considered 'close'. However, as a predefined cost component in a complex objective function to be minimised, with initially unknown levels of traffic sharing the bandwidth on each link, it proves to be less satisfactory.

To deal with partitioning first, it would be desirable for routers within the same area to be connected by the faster links. This increased bandwidth will be best able to carry the full routing updates that these routers will share. The remaining slower links will then be better suited to carry the reduced traffic summaries between areas. A final practical constraint is that the NA may wish to force certain routers to be partitioned together to reflect simple geographical considerations (they share the same site/cabinet, for example). An algorithm to this effect is given in section 3.1.

The question of path determination is more complex. It may be modelled as follows. Suppose there are n routers connecting m networks in the domain (n,m) . Let b_{ij} be the speed of the link (i,j) ($1 \leq i,j \leq n$) (n not m : stub networks need not be considered in the optimisation. Also there is no need to assume full-duplex transmission: $b_{ij} \neq b_{ji}$ if required.). Then, in OSPF terms, the cost of the link (i,j) is given by $c_{ij} = K / b_{ij}$ for some suitable constant, K . ($K > 10^8$ is suggested: OSPF is already experiencing difficulties with Gigabit Ethernet in this respect.) If there is no link (i,j) then $b_{ij}=0$ and $c_{ij}=\infty$. A *domain routing*, $R = ({}^{xy}r_{ij})$, on the domain (n,m) , is defined as:

$$\begin{aligned} {}^{xy}r_{ij} = 1: & \quad \text{if traffic from network } x \text{ (} 1 \leq x \leq m \text{) to network } y \text{ (} 1 \leq y \leq m \text{) is routed} \\ & \quad \text{over the link } (i,j) \text{ (} 1 \leq i,j \leq n \text{).} \\ {}^{xy}r_{ij} = 0: & \quad \text{otherwise.} \end{aligned} \tag{1}$$

DSPA seeks to optimise path cost for each (independent) network pair (x,y) , that is to minimise

$$C_{xy} = \sum_{i=1}^n \sum_{j=1}^n c_{ij}^{xy} r_{ij} \quad (2)$$

for each (x,y) . This, however, takes no account of multiple loads sharing the available bandwidth. Assuming a reduction in bandwidth proportional to the load on a link, the true cost of the routing R across the domain (n,m) is given by

$$C = \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{x=1}^m \sum_{y=1}^m r_{ij}^{xy} . \quad (3)$$

Minimising C_{xy} for each (x,y) will not, in general, minimise C . Both procedures are of equivalent polynomial (n^2m^2) complexity so, in principle, the ‘global’ optimisation objective could be substituted for DSPA to achieve optimal routing. Section 3, however, introduces a significant operational constraint in preparation for which the domain routing is redefined from an individual network perspective.

For any given network x ($1 \leq x \leq m$), define the *network routing* $R_{(x)} = ({}^{(x)y}r_{ij})$ to be:

$$\begin{aligned} {}^{(x)y}r_{ij} &= 1: && \text{if traffic from network } x \text{ to network } y \text{ (} 1 \leq y \leq m \text{) is routed over the} \\ &&& \text{link } (i,j) \text{ (} 1 \leq i,j \leq n \text{)}. \\ {}^{(x)y}r_{ij} &= 0: && \text{otherwise.} \end{aligned} \quad (4)$$

Then, for each such network x , the function to be minimised is

$$C_x = \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{y=1}^m {}^{(x)y}r_{ij} , \quad (5)$$

which can be calculated independently for each network x . Again, evaluating and optimising C_x for every network in the domain, is of n^2m^2 complexity but n^2m for each network x . Minimising C distributes routed traffic across the domain, making best use of all available bandwidth rather than overloading high-bandwidth central paths. Minimising C_x for each network x will not, in general, minimise C across the domain but it will be an improvement over minimising C_{xy} for each pair (x,y) . The significance of these variants is discussed in the next section.

3. Algorithms and Implementation

An essential feature of a practical routing algorithm is that it should run independently on each router upon which the protocol is configured. *Centralised* algorithms requiring parameters from other routers across the domain will not implement successfully. It is for this reason, for example, that C from the previous section proves to be an unsatisfactory objective function. The two sections that follow outline *distributed* algorithms.

3.1. Partitioning

Cost, as determined by line speed, is an appropriate measure of ‘closeness’ for the purposes of partitioning. Grout (2003) gives a centralised partitioning algorithm. The following is a modified, distributed version.

The set of partitions is given by $\tilde{\mathbf{A}} = \{P_x\}$ for each network x . Initially, in the unconstrained case, $P_x = \{x\}$, that is, each network is the sole member of its own partition. However, if the NA chooses to apply partial areas as constraints, then $P_x = P_y = P_z = \dots = \{x, y, z, \dots\}$ for networks x, y, z, \dots in a common partition. The *diameter* of each partition P_x is given by $d(P_x) = 0$ if $P_x = \{x\}$ or $d(P_x) = \max_{y, z \in P_x} c_{yz}$ otherwise. The maximum diameter of a partition is d_{max} and c_{max} is the maximum cost between two adjacent routers in the same partition. c_{max} and d_{max} are configurable by the NA. The *Partitioning Algorithm (PA)*, running on each router i , for each adjacent network x , proceeds as follows:

```

PartitionsFormed := false;
repeat
   $c_{min} := c_{max}$ ;
  for  $z := 1$  to  $m$  do
    if  $P_x \neq P_z$  then
      if  $c_{xz} < c_{min}$  then
         $y := z$ ;
         $c_{min} := c_{xz}$ ;
  if  $c_{min} = c_{max}$  then
    PartitionsFormed := true
  else if  $d(P_x \dot{\cup} P_y) < d_{max}$  then
     $P_x := P_x \dot{\cup} P_y$ ;
     $P_y := P_x$ ;
  else if  $d(P_x \dot{\cup} \{y\}) < d(P_y)$  then
    if  $|P_y| > 1$  then
      for  $z := 1$  to  $m$  do
        if  $z \in P_y$  then
           $P_z := P_z - \{y\}$ ;
     $P_x := P_x \dot{\cup} \{y\}$ ;
     $P_y := P_x$ ;
  else
    PartitionsFormed := true
until
  PartitionsFormed

```

($|P_y|$ is the number of networks in partition P_y .) Each router i builds the partition set $\tilde{\mathbf{A}}$ as far as the constraints c_{max} and d_{max} permit. If a projected partition is too large then networks can instead be moved to smaller partitions if appropriate. The necessary ‘negotiation’ follows the process defined for OSPF elections and exchanges (Moy, 1998). It is implicit in this algorithm, for brevity, that c_{max} and d_{max} are such as to permit a feasible solution. If it is possible that this is not the case then a suitable filter must be added. The algorithm, running cooperatively or in isolation on each router, will derive the optimum set of partitions using an adapted hybrid version of Kruskal’s (1956) and Prim’s (1957) algorithms - greedy algorithms, each of which is known to be optimal.

3.2. Path Determination

There are $m(m-1)$ paths between networks required to define a domain routing, or $n(n-1)$ between routers, discounting stubs. For each network pair, a path of length q ($0 \leq q \leq n-2$) is possible (based on full connectivity), with the q intermediate routers being visited in $q!$ different orders. The total number of possible routings on the domain (n,m) is then

$$m(m-1) \sum_{q=0}^{n-2} q! \quad (7)$$

which is exponential in n . Grout (2004) provides a centralised approach to optimising these interdependent paths. An improved distributed version is offered here. Taking the network perspective, that is considering only network routings, reduces (7) to

$$(m-1) \sum_{q=0}^{n-2} q! \quad (8)$$

Exhaustive search is thus not a practical approach and greedy algorithms are known to fail (generally badly) for this type of problem. The best solution is likely to be a form of *local search* - that is a process of applying small perturbations to a current solution to look for improvement. Although not optimal in itself, the best starting solution is likely to come from DSPA acting on the individual network pairs. The following *Path Determination Algorithm (PDA)*, running on each router i , for each adjacent network x , will iterate until no further improvements are to be found.

```

for  $y := 1$  to  $m$  do
  find  $R_{(x)} = ({}^{(x)}yR_{ij})$  such that  $C_{xy} = \min_{x'y'} C_{x'y'}$  {using DSPA}
repeat
  MaximumImprovement := 0;
  for  $y := 1$  to  $m$  do
    for  $i := 1$  to  $n$  do
      for  $j := 1$  to  $n$  do
        if  $C_x - C_x(i@j:y) > \text{MaximumImprovement}$  then
          MaximumImprovement :=  $C_x - C_x(i@j:y)$ ;
           $y' := y$ ;
           $i' := i$ ;
           $j' := j$ ;
        if MaximumImprovement > 0 then
           $R_{(x')} := R_{(x)}(i'@j':y')$ 
until
  MaximumImprovement = 0

```

C_x is calculated using equation (5). (Again for brevity,) this algorithm includes the subroutines, $C_x(i@j:y)$ and $R_{(x)}(i@j:y)$. $R_{(x)}(i@j:y)$ is the $R_{(x)}$ routing for network x with the link between i and j added to the path to y (if it is not present in $R_{(x)}$) or removed (if it is) and the alternate section of the path modified accordingly. $C_x(i@j:y)$ is the cost of this perturbed solution. (These subroutines are routine but long on code.) Links are added and removed from individual paths in an attempt to improve the overall routing cost. In deviating

from optimal DSPA paths, which minimise C_{xy} for each network pair (x,y) , it is possible to reduce C_x for each network x by distributing the traffic load. As can be seen from the final section, this also has the effect of reducing C .

4. Results, conclusions and future work

Two distributed (router-based) algorithms are given for network partitioning and path determination. Both are intended to provide enhanced performance: the first in comparison with manual configuration and the second over simple pairwise network routing. Testing has been carried out on two platforms:

- The ns2 network simulator (ns2, 2002)
- The University of Wales NEWI (UoWNEWI) NetSim package

Domains of 20, 50 and 100 networks were tested, 12 runs in each case, with topologies and bandwidths generated at random. Timings are given for the partitioning algorithm, PA and the path determination algorithm, PDA, and in comparison with DSPA. Routing cost results are given for a compound algorithm, running independently on each router (as it would in practice) in comparison with DSPA acting on randomly generated partitions. Infeasible parameter sets (eg, disconnected domains) are not included.

The central component of each algorithm is polynomial in complexity. However, both nest within an indefinite loop. With no theoretic bound on run time, empirical results are significant. Table 1 summarises simulated results. A *step* is a single high-level language instruction. *Convergence* is the state of stable partitioning and/or routing across the domain.

Number of networks (m)	(Mean) number of steps to converge			% increase over ...	
	DSPA	PA	PDA	PA	PA + PDA
20	7.2×10^6	8.1×10^5	8.4×10^6	17	28
50	7.4×10^7	8.2×10^6	1.0×10^8	35	47
100	4.3×10^8	4.4×10^7	7.8×10^8	81	92

Table 1: Run times

Number of networks (m)	(mean) $\sum_{x=1}^m C_x$			(mean) C		
	% improvement of PA + PDA over DSPA					
	Minimum	Maximum	Mean	Minimum	Maximum	Mean
20	14	69	37	5	65	20
50	23	88	43	19	69	32
100	26	72	44	25	75	37

Table 2: Routing costs

The percentage increase in run time of PA and PDA over DSPA increases approximately linearly with the size of the domain.

Table 2 compares C_x , (equation 5) summed over all networks x , and C (equation 3) for DSPA with PA and PDA. Although PA and PDA seek to improve C_x across the domain, savings in C also result.

Both algorithms offer significant improvements in routing/loading efficiency at the expense of a modest (in complexity terms) increase in run time.

In conclusion, it is proposed that it may be possible to increase this improvement in C still further by optimising on domain routings rather than network routings. It may not be necessary to resort to a centralised algorithm for this purpose. Legge and Baxendale (2003) report success, on a limited scale, in using ant-colony algorithms for adaptive network routing solutions. Although untested for larger domains, it may be expected that these techniques, in conjunction with the partitioning and path determination concepts presented in this paper, could lead to further advances.

5. References

- Dijkstra, E.W., (1959), "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, Vol. 1, pp269-271.
- Grout, V., (2003), "Towards an Optimal Routing Strategy", *Proceedings of IADIS WWW/Internet 2003*, Algarve, Portugal, 5th-8th November, pp903-906.
- Grout, V., (2004), "A Self-Partitioning Link-State Routing Protocol", *Proceedings of IEE/IEEE ICN'04*, Gosier, Guadeloupe, French Caribbean, 1st-4th March (to appear).
- Kruskal, J.B., (1956), "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem", *Proceedings of the American Mathematical Society*, Vol. 7, pp48-50.
- Legge, D. and Baxendale, P., (2003), "An Agent-Managed Ant-Based Network Control System", *Centre for Telecommunication Networks, School of Engineering, University of Durham, UK*, www.dur.ac.uk/telecoms.networks/public/pdfs/AAMAS-III.pdf
- Moy, J.T., (1998), "OSPF: Anatomy of an Internet Routing Protocol", Addison Wesley.
- Moy, J.T., (2000), *OSPF Complete Implementation*, Addison Wesley.
- ns2, (2002), "The Network Simulator – ns – 2", www.isi.edu/nsnam/ns
- Prim, R.C., (1957), "Shortest Connection Networks and Some Generalizations", *Bell System Technical Journal*, Vol. 36, pp1389-1401.
- Slattery, T. and Burton, B. (2000), *Advanced IP Routing in Cisco Networks*, Osborne McGraw-Hill.