7-1-2010

Some Notes and Results on Bandwidth-based Routing and Implicit Load Balancing

Vic Grout
Glyndwr University, v.grout@glyndwr.ac.uk

Nigel Houlden
Glyndwr University, n.houlden@glyndwr.ac.uk

Recommended Citation

Grout, V. & Houlden, N., "Some Notes and Results on Bandwidth-based Routing and Implicit Load Balancing". Proceedings of the Eighth International Network Conference (INC 2010), University of Plymouth, 6-8 July 2010.

Some Notes and Results on Bandwidth-based Routing and Implicit Load Balancing

Abstract
The Maximum Bandwidth Path Algorithm (MBPA) is introduced as an alternative to Dijkstra's Shortest Path Algorithm (DSPA) for Internet routing. The two are compared and differences noted. Of particular interest is the extent to which each algorithm achieves Implicit Load Balancing (ILB) – the principle of effective link usage for traffic across the network as a whole for non-equal paths and without the use of explicit routing variance. Although MBPA may prove to be more efficient than DSPA generally (further work is required), it is shown, through extensive simulation, that it produces better levels of ILB for real, as opposed to artificial, network scenarios.

Keywords
Internet routing; Implicit load-balancing; Dijkstra, Shortest path algorithm, DSPA, Minimum bandwidth routing

Disciplines
Computer and Systems Architecture | Digital Communications and Networking | Hardware Systems | Systems and Communications

Comments
This paper was presented at the Eighth International Network Conference (INC 2010), July 6-8 2010, which was held in Heidelberg, Germany. It was published by the University of Plymouth, and the symposium proceedings are available at http://www.cscan.org

Some Notes and Results on Bandwidth-based Routing and Implicit Load Balancing

Vic Grout, Nigel Houlden

Centre for Applied Internet Research (CAIR)
Glyndŵr University, Wales
Plas Coch Campus, Mold Road, Wrexham
LL11 2AW, UK
v.grout@glyndwr.ac.uk
n.houlden@glyndwr.ac.uk

Abstract: The *Maximum Bandwidth Path Algorithm* (*MBPA*) is introduced as an alternative to *Dijkstra's Shortest Path Algorithm* (*DSPA*) for Internet routing. The two are compared and differences noted. Of particular interest is the extent to which each algorithm achieves *Implicit Load Balancing* (*ILB*) – the principle of effective link usage for traffic across the network as a whole for non-equal paths and without the use of explicit routing variance. Although MBPA *may* prove to be more efficient than DSPA generally (further work is required), it is shown, through extensive simulation, that it produces better levels of ILB for real, as opposed to artificial, network scenarios.

1 Introduction: DSPA and Internet Routing

Dijkstra's Shortest Path Algorithm (*DSPA*) [Di59] plays a prominent role in routing traffic over the Internet. The *Open Shortest Path First* (*OSPF*) [Mo98] and *Intermediate System to Intermediate System* (*IS-IS*) [Or90] protocols, for example, both use DSPA to calculate routing paths through a network. Modelling a network as a *graph* comprising *n vertices* (or *nodes*) and *m edges* (or *arcs* if direction is significant), DSPA is a simple algorithm that, starting from a *root* vertex, expands a *spanning tree* across the graph until all vertices are connected to the root via the shortest possible *path*. The essential operation of DSPA is described in Figure 1. Full implementation details can be found in [BP05 P384].

An example of DSPA's operation is given in Figure 2 with $n=6$ and $m=10$. Costs are given for each edge present; non-existent edges can be considered to have infinite cost. The process of generating a spanning tree, rooted on A is demonstrated. (In practice, a similar process is effected for all nodes.) In Steps 1 to 5, the nodes D, B, C, E & F are included by DSPA being, in turn, the new node to be reached via the shortest path from A to that node calculated as the sum of the (shortest) path to a currently spanned node and the cost of the link from that node to an unspanned node. (Shortest distances to each node are shown in brackets.) The final cost from A to F, for example, is *11*. This is an efficient process. A crude implementation of DSPA can be achieved in $O(n^2)$ steps, which can be reduced to $O(n\log n + m)$ with greater finesse and some increase in space complexity [BP05 P386].

```
DSPA:
INPUT:     $G$: graph with vertex set $V = \{0,\dots,n-1\}$ & edge set $E$
           $r$: vertex in $V$
           $\underline{c}$: cost matrix $[0,\dots,n-1;0,\dots,n-1]$ on $E$
OUTPUT:    $T$: shortest path tree on $G$ rooted at $r$
           $\underline{d}$: distance array $[0,\dots,n-1]$
INITIALISATION:  $T = \{r\}$
                 $d[r] = 0$
IMPLEMENTATION:  while $V-T \neq \emptyset$ do
                     find edge $(u^*,v^*)$ minimising $d[u]+c[u,v]$
                                              for all $u \in T$, $v \in V-T$
                     add vertex $v^*$ and edge $(u^*,v^*)$ to $T$
                     $d[v^*] = d[u^*]+c[u^*,v^*]$
```

Figure 1: Dijkstra's Shortest Path Algorithm (DSPA) (adapted from [BP])

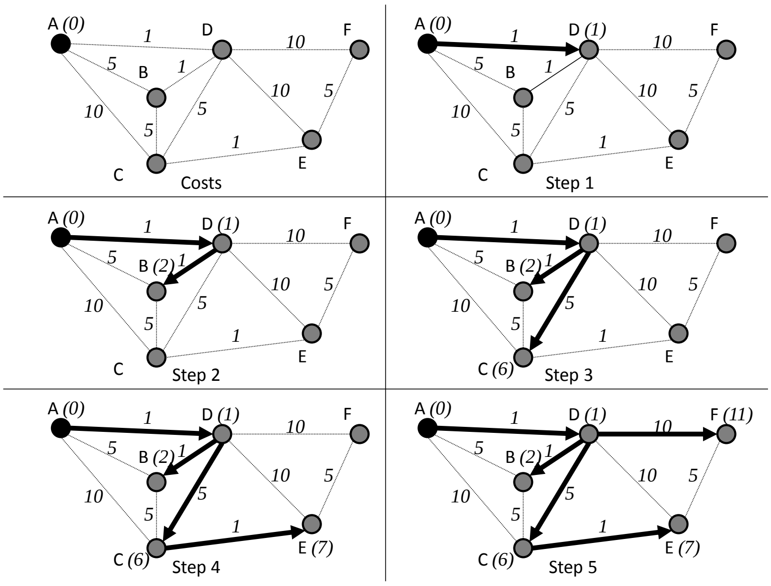


Figure 2: DSPA example

In the Internet, vertices/nodes are *routers* or *switches* and edges/arcs the *links* connecting them. In its practical implementation, DSPA edge costs are typically calculated inversely on the basis of link *bandwidth*. OSPF, for example, defines the cost of a link (u,v) to be $c_{(u,v)} = 10^8 / b_{(u,v)}$, where $b_{(u,v)}$ is the bandwidth of the link (u,v) (although the constant is largely irrelevant). Once topological (*link state*) information has been exchanged, running DPSA independently on each router allows a path to be determined to each remote target and this information (the *route*, the first step in the path) entered in the router's *routing table*.[1]

2 A Possible Criticism of DSPA in the Real World - and an Alternative

The notion of applying a fixed cost to each edge of a graph, as described in Section 1, is largely a theoretical one, inherited from the original underlying graph theory [Wi96]. In the real world of Internet algorithms and protocols, this is often problematic [GCH08] and sometimes ludicrous [GCP07]. The judgement for DSPA applied to Internet routing is a mixed one. On the one hand, some concept of the cost of a link, in terms of its attractiveness as a routing option, seems appropriate; on the other, the assumption that the cost of a path should be calculated as the sum of its individual link costs is less than axiomatic. In fact, this is clearly *not* accepted by many other routing protocols – see the discussion on Cisco's *Enhanced Interior Gateway Routing Protocol* (*EIGRP*) in [Ho06] for example.

The problem, in particular here, stems from the, possibly inappropriate, manner in which this summation of link costs is related to the initial determination of these costs. Taking $c_{(u,v)} = 1 / b_{(u,v)}$ (ignoring the constant), for the link (u,v), and summing link costs for a path, P, gives a path cost of $C(P) = \Sigma_{(u,v)\in P}\, c_{(u,v)} = \Sigma_{(u,v)\in P}\, 1/b_{(u,v)}$. Calculating the cost of a path as the sum of the inverses of the bandwidth of its individual links is hardly intuitive!

```
MBPA:
INPUT:    $G$: graph with vertex set $V = \{0,…,n-1\}$ & edge set $E$
          $r$: vertex in $V$
          $\underline{b}$: bandwidth matrix$[0,…,n-1;0,…,n-1]$ on $E$
OUTPUT:   $B$: maximum bandwidth tree on $G$ rooted at $r$
          $\underline{m}$: maximum array$[0,…,n-1]$
INITIALISATION: $B = \{r\}$
                $M[r] = \infty$
IMPLEMENTATION: while $V-B \neq \emptyset$ do
                   find edge $(u^*,v^*)$ maximising
                       min($m[u],b[u,v]$) for all $u \in T$, $v \in V-T$
                   add vertex $v^*$ and edge $(u^*,v^*)$ to $B$
                   $m[v^*] = min(m[u^*],b[u^*,v^*]$
```

Figure 3: The Maximum Bandwidth Path Algorithm (MBPA)

The textbook argument for DSPA [He02] maintains that OSPF is configured so as to favour links of high bandwidth since, intuitively, these will be able to carry traffic with less risk of congestion. However, if the primary aim of the routing process is to avoid

---

[1] In reality, routers/switches calculate (and hold in their routing tables) the distance to each remote *network*, not other routers/switches. However, in this paper, for simplicity, and without loss of accuracy/detail, we consider paths between *nodes* only and use the term *network* to mean what could otherwise be called the *domain*. See [GH05] for a more traditional treatment.

bottlenecks (which is a wholly laudable, if not unique, objective), then it could be argued that it would be advantageous to choose paths in which the *minimum* bandwidth (of the worst link) was as large as possible since this is precisely where any bottleneck issues are likely to occur. In this case, the link costs of the other components of a path are irrelevant and should not contribute to the cost of the path. (literally: *"a chain is only as strong as its weakest link"* - Anon.) This suggests an alternative to DSPA, the *Maximum Bandwidth Path Algorithm* (*MBPA*), outlined in Figure 3.

At each step of building its spanning tree, MBPA chooses the link that *maximises the minimum bandwidth* of the path connecting the newly-spanned node to the root rather than minimising the summed cost as with DSPA. The complexity of MBPA, subject to implementation, is the same as for DSPA. An example of the operation of MBPA is given in Figure 4.

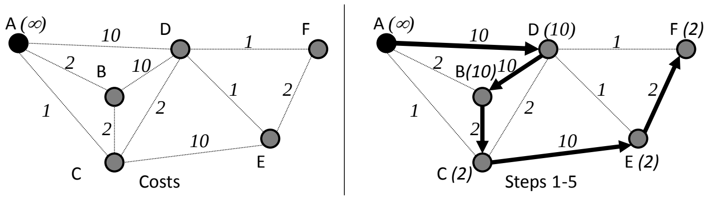


Figure 4: MBPA Example

This example is the MBPA equivalent of DSPA in Figure 2 with $b = 10/c$ (or more precisely, $c = 10/b$; since cost is calculated *from* bandwidth in DSPA, we should regard the first part of Figure 2 as having been derived *from* the first part of Figure 4). Here, the figures in brackets indicate the minimum bandwidth along the best path leading to each node (theoretical in the case of A).

Note that, at Step 3 in Figure 4 (implicit) – with A, D & B currently spanned, MBPA has two equally good choices. The next step could be to add the link (B,C) *or* the link (D,C). (B,C) has been shown selected in Figure 4 to demonstrate that differences *can* occur between DSPA and MBPA. In fact, if MBPA employed a *Minimum Hop Tie-breaker* (*MHT*), selecting the path with the fewest links, then (D,C) would be chosen instead.[2] However, the integers in Figures 2 and 4 have been chosen for simplicity and convenience. If the bandwidth of (D,C) in Figure 4 is changed to *1.8*, for example, so that the cost of (D,C) in Figure 2 becomes approximately *5.5*, then a difference in behaviour between MBPA and DSPA is ensured. Section 3 extends this analysis of the variation between the two algorithms.

A final note for this section is that the operation of MBPA is *effectively* the same as EIGRP if the EIGRP *k-values* (distance coefficients) are changed from the default $\{k_1=1, k_2=0, k_3=1, k_4=0, k_5=0\}$ to $\{k_1=1, k_2=0, k_3=0, k_4=0, k_5=0\}$ [Ho06]. (That is, the resultant routing tree will be the same; the algorithmic derivation of the tree will follow a

---

[2] Note, in preparation for the next section, that DSPA can also be configured to employ MHT – that is, to choose the minimum-hop path if summed bandwidths are tied.

different course.) Partly due to observations such as this, it is widely held that alternatives such as MBPA should provide more effective routing than DSPA.

However, in truth, although the argument against DSPA is a fashionable one [Li06], there is little empirical data to suggest that summing bandwidth over a path is substantially inferior to any other approach. Most other evidence is essentially circumstantial; EIGRP, for example, outperforms OSPF [GZ94] for a variety of reasons; the method of aggregation of bandwidth may or may not be a factor. Further research is needed in this direction; however a different consideration is discussed in the next section.

3 A Different Comparison of DSPA and MBPA – *Implicit Load Balancing*

So far, this paper has introduced MBPA as an alternative to DSPA and noted that:

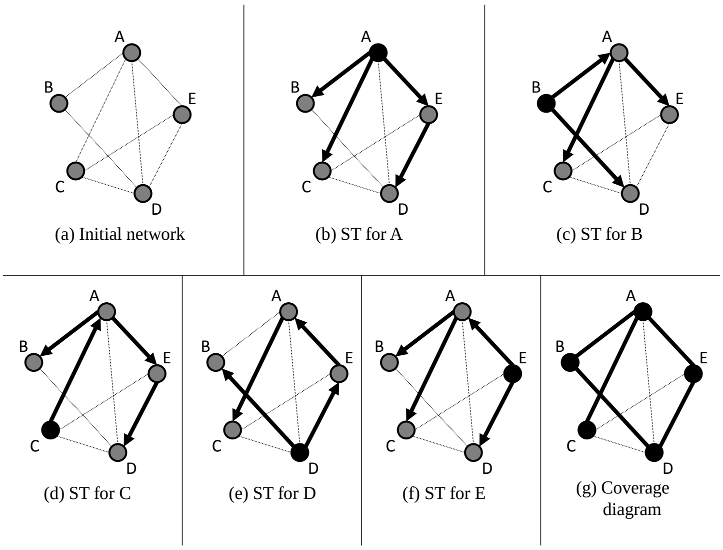
1. DSPA and MBPA can produce different routing strategies under certain circumstances, and
2. In terms of routing efficiency, there is little evidence of one being markedly superior to the other.

The paper now considers one particular feature of a routing protocol: the extent to which it achieves *Implicit Load Balancing* (*ILB*), to be defined below.

Conventional load balancing is supported by most existing routing protocols [Ha00]. The essence of load balancing, performed independently on each router, is that more than one route to a remote destination can be held in the router's routing table and traffic to that destination shared among the different routes available. This will happen automatically with all protocols with load balancing capabilities (which is most) if the cost of two paths is equal. It can also be forced, in some, by the use of tolerance settings; the `variance` command in EIGRP [RWS00], for example, accepts into the routing table all routes with costs within a given ratio of the lowest cost. In all cases, load balancing only takes place on a router-by-router basis (so that the overall effect of traffic interaction over the whole network is ignored) and works only for equal or 'artificially made equal' routes.

Implicit Load Balancing (*ILB*) is a consideration of how *all* routes, equal and unequal, share links in the network, not by design but naturally, through the overlaying of paths to and from all sources and destinations onto the same network. A network in which most links are used by at least one route is a well-balanced one; one in which only a few arterial links are used by most routes is not. Balanced networks are known to have a variety of advantages such as being able to survive failure better [Bh98], reconfigure/reconverge faster [PA07], tolerate higher levels of changeable/bursty traffic [Re07] and deal with a number of other issues better than unbalanced ones [Gr07].

Consider the example in Figure 5. Five nodes are connected by a partial mesh in 5(a); only the links (B,C) and (B,E) are absent. 5(b)-(f) show possible routing spanning trees for nodes A, B, C, D & E. Overlaying these trees on a single graph gives the *Coverage Diagram* (*CD*) in 5(g).[3]



(a) Initial network (b) ST for A (c) ST for B (d) ST for C (e) ST for D (f) ST for E (g) Coverage diagram

Figure 5: Spanning Trees and Coverage Diagram

Define the *Implicit Load Balancing Index* (*ILBI*) to be the ratio of covered links to actual/available links across the whole network. In the example in Figure 5, this gives an ILBI of *5/8* or *0.625*. In general, the higher the ILBI, the better (implicitly) balanced the routing for the underlying network. As a further illustration, if all spanning trees are calculated for the network in Figures 2 and 4, DSPA yields an ILBI of 0.7 whilst MBPA gives a figure of 0.6.[4]

Different combinations of network topology and link costs can yield very different ILBIs for DSPA and MBPA as the carefully selected example in Figure 6 shows. 6(a) shows an initial network with (for convenience) $c = b = 10$ (which is possible if $c = 100/b$). The coverage diagram for both DSPA and MBPA is given in 6(b); both have an ILBI of *1*. 6(c) gives an alternative initial network for the same topology but with different bandwidths. If, this time, $c = 315/b$ (again chosen for convenience to yield integer results), then the DSPA costs are as in 6(d). Once again, the CD for DSPA is that of 6(b) with an ILBI of *1* once more. However, the CD for MBPA is as shown in 6(e), this time with an ILBI of $6/10 = 0.4$ – a considerable difference.

Although the example in Figure 6 is clearly a contrived one, it amply demonstrates that DSPA and MBPA can produce quite difference levels of implicit load balancing. The next section delivers simulations and experimental results on these differences.

---

[3] In this case, direction has been ignored since each used (*covered*) link is used (covered) in each direction by at least one routing tree. A proper distinction between the *symmetric* and *asymmetric* cases is made in the next section.

[4] The evaluation is left to the reader.

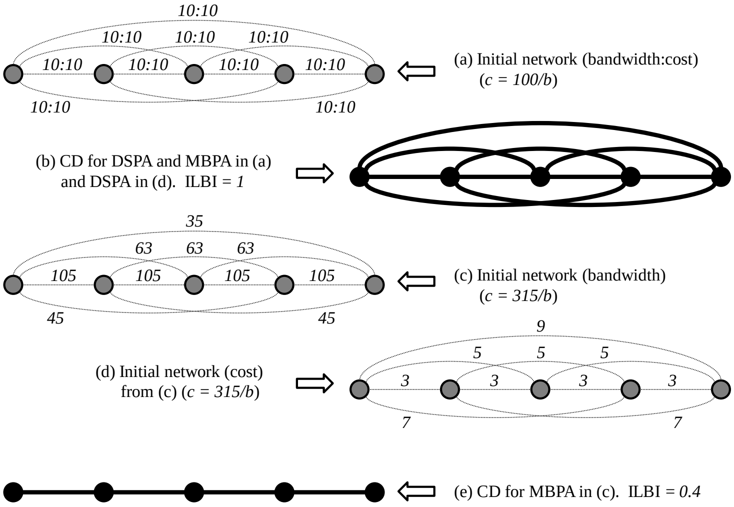


Figure 6: Differences in ILB from DSPA and MBPA

4 Simulations and Experimental Results

This section describes a large number of simulations carried out for various initial network configurations. The number of nodes in these test networks varied from *10* to *200*. A number of runs were completed for each experiment, ranging from *1000* for *10* nodes to *50* for *200* – as the latter took considerably longer. On this basis, of course, it could be argued that the results for larger values of *n* are more susceptible to statistical error. However, as Figures 7-11 show, it is precisely for these larger networks that ILBI values tend to their limits. In all cases, the MHT separator (Section 2) was used for paths of equal length or minimum bandwidth.

Firstly, complete (full-mesh) initial network instances were generated with link bandwidths randomly and uniformly distributed between *10* and *100*. (Taking *c = 1000/b*, gives a similar range for DSPA costs.) There are two cases to consider:

- ‘Asymmetric’: Bandwidths are assigned independently in each direction for a link. In the resultant CD, a link is considered covered if it used by an ST in the direction in question;
- ‘Symmetric’: Bandwidths are the same in both directions for a link. A link is considered covered if it is used by an ST in either direction.[5]

The results of this first simulation are given in Figure 7. Although, in all sets of results (in Figures 7-11), there are instances where either DSPA or MBPA gives a better ILBI

---

[5] In fact, for a symmetric configuration, a link used in one direction *will* be used in both directions; Figure 5 is an example – though not a proof!

than the other, DSPA outperforms MBPA in terms of ILB on average and in the majority of runs. There is little difference between the asymmetric and symmetric cases here with the two completely coinciding for DSPA and almost so for MBPA.

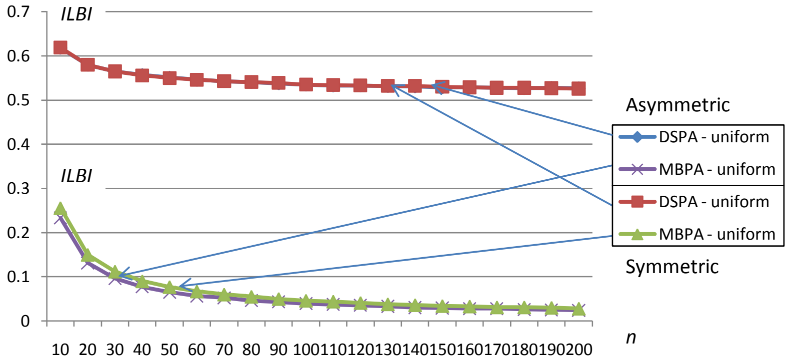


Figure 7: ILBI for DSPA and MBPA with Uniformly Distributed Bandwidths

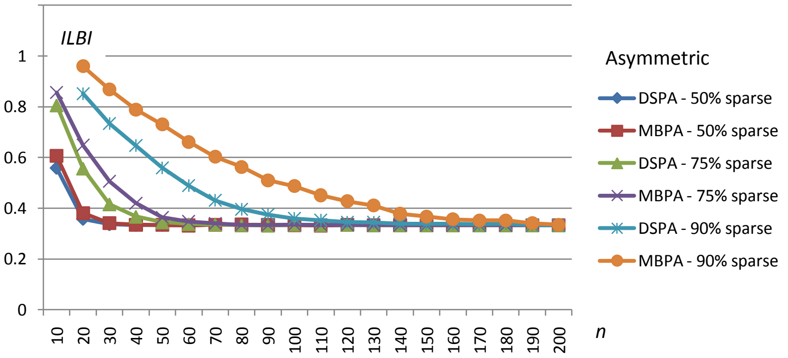


Figure 8: ILBI for DSPA and MBPA with Sparse Asymmetric Bandwidths

Figures 8 and 9 show the asymmetric case for sparse (partial-mesh) initial networks with more realistic bandwidths. In Figure 8, networks were generated with, in turn, a *50%*, *25%* and *10%* probability of there being a link between any two nodes (the networks thus being *50%*, *75%* and *90%* sparse respectively). For those links present, bandwidths were randomly, uniformly assigned as *100Mb/s*, *10Mb/s* and *1.544Mb/s* (T1) representing a balance of LAN and WAN connections.[6] Taking the conventional OSPF calculation of $c = 10^8/b$, gives DSPA costs of *1*, *10* and (approximately) *65* respectively. Here the change is remarkable; MBPA gives a better ILBI measure than DSPA in all

---

[6] True, the use of *10Mb/s* speeds and not *1000Mb/s* is somewhat dated but only *relative* differences are important and these figures are easier to work with. (The *100/1000* calculation for Gigabit Ethernet, for example, causes ‘issues’ for the integer arithmetic of some router operating systems!)

cases, the difference being more marked the more sparse (and consequently more realistic) the initial networks become.[7] [8]

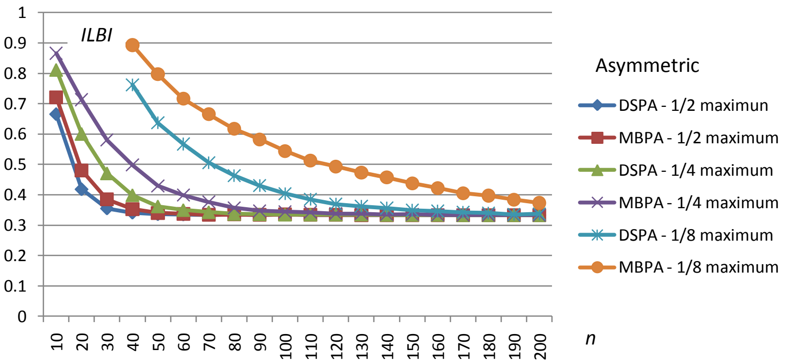


Figure 9: ILBI for DSPA and MBPA with Asymmetric Bandwidths and Maximum Link Lengths

Figure 9 repeats the simulations of Figure 8 with some geographical considerations taken into account. Here, nodes are randomly placed within a unit square; links are only permitted for nodes within, in turn, one half, one quarter and one eighth of the maximum possible distance in the unit square (so $\sqrt{2}/2$, $\sqrt{2}/4$ and $\sqrt{2}/8$ respectively). For the eighth-maximum case, bandwidths are allocated as in Figure 8 and similarly for the quarter- and half-maximum cases but with the 'local' networks constrained by these distances being *25%* and *50%* sparse respectively. Once again, MBPA outperforms DSPA (on average) in each case, the more so for sparser, more realistic initial networks.

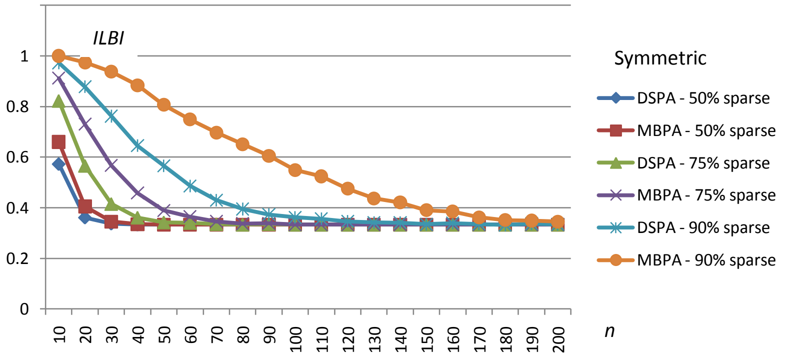


Figure 10: ILBI for DSPA and MBPA with Sparse Symmetric Bandwidths

---

[7] Not all simulation curves start at *n = 10* since most or all initial networks are disconnected for these combinations of parameters and routing trees do not exist.

[8] All simulation curves tend to a value of ILBI = *1/3* in these tests since, as networks become larger and (relatively) more path choices are available, the highest bandwidth links (one third of all present in these simulations) will be increasingly available and thus used. Different configurations yield different limits but the relative performance of DSPA and MBPA remains the same.

In comparison, Figures 10 and 11 give the equivalent results for Figures 8 and 9 for the symmetric case. There is more variation between the symmetric and asymmetric cases for these realistic initial configurations but, once more, MBPA provides the superior ILB.

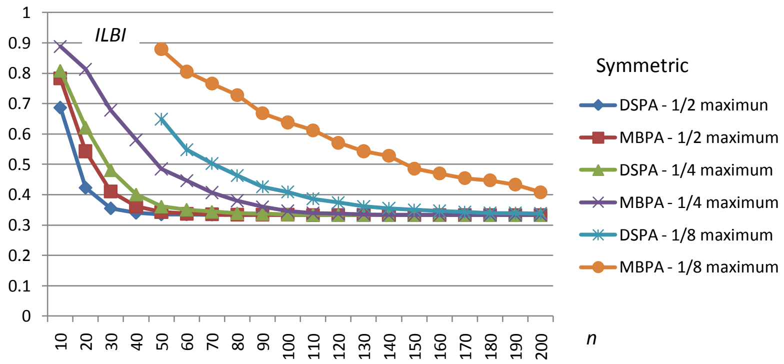


Figure 11: ILBI for DSPA and MBPA with Symmetric Bandwidths and Maximum Link Lengths

Finally, the ILBI evaluation for DSPA and MBPA was conducted on three known real-world network fragments of *35*, *38* and *96* nodes respectively.[9] In each case, the initial network structures consisted essentially of a number of components connected by high-speed links with one or or more edge nodes in each component connecting to edge nodes in other components via slower links.[10] (All were symmetric.) Link bandwidths ranged from 1.544Mb/s (T1) to 1000Mb/s (Gigabit Ethernet). The results are given in Table 1. Although these results are at some variance with the equivalent simulated results, once more, MBPA is superior.

	Sparseness	DSPA ILBI	MBPA ILBI
n = 35	*0.67*	*0.435*	*0.572*
n = 38	*0.77*	*0.420*	*0.488*
n = 96	*0.82*	*0.411*	*0.558*

Table 1: ILBI Comparison of DSPA and MBPA for Three Real-world Networks

5 Conclusions and Future Work

There are a number of issues raised here and some comments are necessary before the paper concludes. Firstly, there is no clear evidence that DSPA or any of its bandwidth-based alternatives are significantly superior than any others in terms of routing (traffic carrying) efficiency [Li06, GZ94]. Although this might be seen as unfortunate, it is hardly surprising. With typical real-world bandwidths, and consequently DSPA costs,

---

[9] Further details must unfortunately remain confidential for commercial reasons.

[10] A fairly typical configuration in line with the *small world* or *scale-free* paradigm to be found in [Mi09 P227-270].

usually orders of magnitudes apart, it takes a potentially long path for DSPA or an alternative to choose different routes. When this does happen, the change in overall efficiency may yet be small; however, it can have a considerable effect on balancing as the examples in this paper show. What is clear is that a more concerted attempt to measure relative efficiencies is long overdue [Ns08, Ns09, Op09].

Balancing itself is a different matter. A balanced network is known to have several advantages beyond simple efficiency [Bh98, PA07, Re07, Gr07]. Leaving aside, forced balancing methods, ILB is a useful consideration as it deals with the level of balancing inherent in a given algorithm. This paper demonstrates clearly that MBPA, a simple alternative to DSPA, produces better levels of ILBI for realistic forms of initial network configuration

There are some unresolved issues, however. An interesting question with which to begin is whether the application of the MHT is valid. Although the selection of shorter (hop) paths in the case of tied (bandwidth) distances seems intuitive, the effects are less obvious. In fact, although it could be argued that shorter paths are likely to increase efficiency and be less error prone (fewer links give less chance of link failure), it should be clear that longer paths provide better balancing (more links in each path will, on average, imply more links used overall). In this case, the very appropriateness of the balancing objective itself has to be questioned. Is balancing really a secondary objective or one of a number of metrics to be considered in an EIGRP manner [Ho06]? Should it be considered independently? Further testing on the combined effect of these settings on real-world networks is necessary.

Also, the issue of direction in the consideration of link and network coverage is subtler than might be realised. Although the symmetric case is simple (links used in one direction by one spanning tree will be used in the opposite direction by another[11]), the asymmetric case requires more careful consideration. It is not difficult to construct a network configuration in which, between two particular nodes say, traffic is well-balanced in one direction but poorly so in the other. An average ILBI could be argued to produce a somewhat distorted impression in this case and a finer tuning of the balancing calculation might be desirable, which leads to the final observation …

Because it must be acknowledged that the binary notion of link coverage, described here, is limited in scope. Marking a link merely as covered or not is a fairly primitive measure of the level of balancing in a network routing strategy. Better would be to keep a tally of the *number* of times a link is used, in each direction if necessary. It would then be possible to calculate a correlation between link usage and link bandwidth across the network or between link usage and traffic levels if known. In essence, this is a reasonably straightforward extension. However, this paper is already too long so this is left for another day.[12]

---

[11] A proof is still needed for this!

[12] Watch this space!

References

[Bh98] Bhandari, R.: Survivable Networks: Algorithms for Diverse Routing. Springer, 1998.

[BP05] Berman, K.A.; Paul, J.L.: Algorithms: Sequential, Parallel and Distributed. Thomson, USA, 2005.

[Di59] Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik, 1, 1959: P 269-271.

[GCH08] Grout, V.; Cunningham, S.; Hebblewhite, R.: An Illustrative Discussion of Different Perspectives in Network Engineering: Proc. 7th Int. Network Conf. (INC 2008), Plymouth, UK, 8-10 July 2008: P 91-104.

[GCP07] Grout, V,; Cunningham, S.; Picking, R.: Practical Large-Scale Network Design with Variable Costs for Links and Switches: Int. Jnl. Computer Science and Network Security, Vol. 7, No. 7, July 2007: P 113-125.

[GH05] Grout, V.; Houlden, N.: Principles of Optimal Interior Routing: Proc. 1st Int. Conf. on Internet Technologies and Applications (ITA 05), Wrexham, UK, 7-9 September 2005: P 274-281.

[Gr07] Grout, V.: So Many Paths! (A stroll through the minefield separating Internet theory from practice.) (keynote address), Int. Conf. E-Activity and Leading Technologies (E-ALT'07), Oporto, Portugal, 3-6 December 2007.

[GZ94] Garcia-Luna-Aceves, J.J.; Zaumen, W.T.: Area-based, loop-free internet routing: Proc. 13th IEEE Conf. Networking for Global Communications (INFOCOM’94), Toronto, Canada, 12-16 June 1994: P 1000-1008.

[Ha00] Halabi, S.; Internet Routing Architectures. Cisco Press, USA, 2000.

[He02] Heckmann, B.; CCNP Practical Studies: Routing. Cisco Press, 2002.

[Ho06] Houlden, N. et al.: Extended End-to-End Cost Metrics for Improved Dynamic Route Calculation: Proc. 7th Int. Network Conf. (INC 2006), Plymouth, UK, 11-14 July 2006: P 89-96.

[Li06] Liang, G.; Lee, W.C.; Metke, A.; Bansal, D.: Intra-Domain Routing Topology Bandwidth Management Architecture: Proc. 15th Int. Conf. Computer Communications and Networks (ICCCN 2006), Arlington, VA, USA, 9-11 Oct. 2006: P 25-32.

[Mi09] Mitchell, M.; Complexity: A Guided Tour, Oxford University Press, 2009.

[Mo98] Moy, J.: OSPF Version 2. RFC 2328 (http://tools.ietf.org/html/rfc2328), April 1998.

[Ns08] ns-2; The Network Simulator – ns-2: http://www.isi.edu/nsnam/ns/ (2008).

[Ns09] ns-3; The ns-3 Network Simulator: http://www.nsnam.org/ (2009).

[Op09] OpNet; The OpNet Network Simulator: http://www.opnet.com (2009).

[Or90] Oran, D.: OSI IS-IS Intra-domain Routing Protocol. RFC 1142 (http://tools.ietf.org/html/rfc1142), February 1990.

[PA07] de Paula Herman, W.; de Almeida Amazonas, J.R.; Hop-by-hop Routing Convergence Analysis Based on Paths Algebra: Proc. IEEE Conf. Electronics, Robotics and Automotive Mechanics (CERMA 2007), Morelos, Mexico, P 9-14.

[Re07] Rezaul, K.M.: Estimating Long-Range Dependent Self-Similar Network Traffic: Performance Evaluation and Control", PhD Thesis, University of Wales, October 2007.

[RWS00] Retana, A.; White, R.; Slice, D.: EIGRP for IP. Addison Wesley, 2000.

[Wi96] Wilson, R.J.: Introduction to Graph Theory. Prentice Hall, 1996.