Kent Academic Repository

Waters, A. Gill (2000) *Hierarchies for Network Evolution.* In: 16th UK Teletraffic Symposium on Management of Quality of Service - the New Challenge. . 5/1-5/6. IEE

Downloaded from
https://kar.kent.ac.uk/22019/ The University of Kent's Academic Repository KAR

The version of record is available from

This document version
UNSPECIFIED

DOI for this version

Licence for this version
UNSPECIFIED

Additional information

Versions of research works

Versions of Record
If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts
If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries
If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

(Presented at the Sixteenth UK Teletraffic Symposium on "Management of Quality of Service - the New Challenge", 22-24 May 2000 at Harlow, UK)

HIERARCHIES FOR NETWORK EVOLUTION

Gill Waters

The success of the Internet is due to best-efforts delivery which allows for easy expansion, coupled with a congestion-adaptive reliable transport protocol (TCP), which serves well for the delay-insensitive traffic of Web browsing or file transfer. There are several new proposals for handling Quality of Service (QoS) for real-time and multimedia traffic, but their introduction is slow. The scale of the Internet is one of the impediments to successful QoS provision. Hierarchical structure is the key to scaling problems, but it must be provided in a way that helps evolution too. This paper reports work in progress on techniques for organising hierarchies using clustering, which could be applied to clouds of QoS capable routers or to support multicast applications.

1. Introduction

This paper describes work in progress to identify techniques which might aid the evolution of the Internet. In particular, we suggest applying clustering techniques to nodes within the network. These techniques would be used to help to identify clouds of nodes (e.g. routers or servers) with similar capabilities and to arrange such clouds into hierarchical structures such that protocols need only communicate directly with systems at the same hierarchical level.

2. Need for categorisation and hierarchies on the Internet

In general, hierarchies are needed to reduce the burden on routers of keeping large amounts of network state information and to reduce the number of signalling messages exchanged. Provided suitable hierarchies are available, full information need only be kept on systems within the same hierarchical level together with summarised information about adjoining levels. The technique is therefore particularly helpful for routing where large tables must currently be maintained. Increasingly, routers and servers on the Internet are being equipped to provide specialised services, especially those associated with QoS provision or multicasting. To evolve such services to a large user population, it is important that systems with similar capabilities can be categorised and merged to form larger systems which can then form their own hierarchies.

For multicast applications, hierarchies are useful both for routing and for transport level support. For example, scalable reliable file distribution can be achieved with a hierarchy of systems that provide error and flow recovery on behalf of others. Some applications may also need techniques to reduce the number of multicast addresses needed locally and may use hierarchical trees to achieve this

2.1 QoS on the Internet

Today's Internet offers a best effort connectionless service, in which datagrams may be lost, discarded, unpredictably delayed or mis-ordered. Fortunately, the Transmission Control Protocol (TCP) can both recover from errors and reduce injected traffic when there is congestion, so that non-delay sensitive (elastic) traffic requiring reliable delivery (e.g.file transfer, Web page retrieval) is

Computing Laboratory, University Kent at Canterbury, Canterbury, Kent CT2 7NF
E-mail: A.G.Waters@ukc.ac.uk

well served. Real time traffic, such as voice or video, which have delay and bandwidth constraints (inelastic traffic) are poorly served, but there is increasing demand for them to be carried on the Internet. Such traffic can only be carried if the network allows the negotiation of Quality of service parameters and offers resource reservation, queueing disciplines and routing together with the protocols to support them.   A number of proposals have been put forward including the Integrated Services Architecture (Intserv), Differentiated Services (Diffserv) and Multi-Protocol Label Switching (MPLS).

Intserv [RFC 1633] follows similar conventions to those of ATM networks and supports individual flows with specific QoS requirements. It is intended to support guaranteed, controlled (i.e. within some tolerance limits) and best efforts traffic. To maintain QoS and control congestion, Intserv requires admission control, routing algorithms, queueing disciplines and discard policies. A router must deal with the flows as a whole and maintain look-up tables that allow fast decisions on classification and priority queueing for each packet. Intserv is complex and has therefore been slow to be implemented. One important component, which is now implemented is many routers, is the Resource ReServation Protocol (RSVP) [RFC 2205] which can handle both unicast and multicast resource provisioning.

Because of the difficulty of introducing Intserv, an alternative technique based on service classes, Diffserv, was introduced [RFC 2475]. This is intended to aid scalability, to offer an evolutionary approach and to reduce complexity within the network. In Diffserv, the QoS is based on a field in the IP header. It offers differential levels of service for *aggregated* traffic (normally unidirectional) and uses well-defined building blocks for end-to-end agreement and hop by hop treatment, thus separating policy from forwarding. More recently, MPLS has been proposed [Callon et al]. In this, packets are forwarded in Forward Equivalence Classes (FEC) indicated by a label. Each packet is assigned to an FEC and labelled as it enters the network (based on analysis of the header). The label is used to determine QoS and forwarding policy. Because of its similarity with lightweight virtual circuits, MPLS offers easy integration with ATM and Frame Relay networks.

Because of the variety of schemes and the difficulty of introduction of the services, it is likely that QoS will initially only be offered in small clouds and most current implementations are on IP-based intranets belonging to specific organisations or consortia. As the techniques begin to be more widely accepted, it is likely that they will be made available to users on the public Internet and the problem of scaling arises. To overcome this, we need to be able to identify routers and servers which offer specific public QoS support and to group them in the most appropriate way. In order to do this, we need information on geographical location, services provided, and whether subscription and charging are supported. Some of this information is already available, but better availability would help Internet evolution regardless of the scalability mechanisms used. Intuitively, we need to group systems together geographically - hence *clustering* seems a sensible approach.

3. How can clustering help?

3.1 An introduction to clustering algorithms

There are two main types of clustering algorithm: hierarchical and non-hierarchical. See for example [Sharma]. However the term hierarchical is not used in the same sense as above. A hierarchical clustering algorithm will assign each node in turn to an appropriate cluster, gradually reducing the number of clusters and, in so doing, produce a hierarchical tree describing the way in which the clusters have been built up. Non-hierarchical clustering algorithms (e.g. kmeans) divide the nodes into a given number, k, of clusters based on a given starting set of centres and then iterate to optimise the membership of the clusters. For each type of algorithm, there are a number of options, for example, to find the nearest neighbour or to optimise cluster cohesion. One recommended technique is to determine the number of clusters by evaluating the results of a hierarchical technique and then to optimise for this number of clusters by using a non-hierarchical technique.

3.2. Applying clustering to network hierarchies

If we consider maps of Internet use, this is concentrated on major cities and is certainly not evenly spread around all possible physical locations. By arranging the system into clusters, the number of trunk interconnections can be reduced. Iterative use of clustering algorithms should also help to divide very large numbers of systems into further levels.

Clustering techniques have been applied in many fields, including image anaysis and vowel discrimination in speech. Their application to the design of backbone communication networks is described in [Cahn]. One of the first problems of applying this technique to packet networks is to decide what are the optimisation criteria? For instance, multicast applications may need a maximum fan-out at each level or may need to minimise the number of levels in the hierarchy. Note that clustering algorithms will always partition the nodes into sets and in many cases a useful grouping will be obtained even if it is not optimal.

The easiest way to perform the clustering is to use geographical closeness of like systems. Applying clustering to the actual network topology is a more complex problem. It may be that geographical clustering can be done at higher levels in the hierarchy with the topology being applied more accurately on the local scale.

4. Related Work

A number of authors have considered techniques for networking hierarchies and, although several of these discuss clusters, they do not actually use clustering algorithms to group nodes into clusters as we propose. Most proposals are related to multicasting or to Web caching.

An approach to hierarchical clustering for multicast routing based on Voronoi diagrams is discussed by Baccelli et al [Baccelli et al]. (A Voronoi diagram divides a plane into regions each based on a specified node. Within each region, any other node is closer to that region's specified node than to the specified node in any other region.) The technique divides the network into domains each fed by a core and cores are organized into a hierarchy of centre-based trees.

Although this approach looks promising, their evaluation uses a uniform distribution of nodes, which may not be realistic. The authors found that optimum tree cost can be found by having much bigger fan-outs nearer the source of the multicast tree, with successively lower fan-outs at lower levels in the hierarchy, whereas practical systems may require similar fan-outs at each level.

Another hierarchical scheme is that of Chatterjee and Bassiouni who describe a two level hierarchy [Chatterjee & Bassiouni]. Optimization at the top level (linking the clusters) uses a minimum spanning tree and at the lower levels, a broadcast tree with optimal delay within each cluster. Yallcast is another protocol-based approach to hierarchical multicast trees in the Internet [Francis].

A project to evaluate hierarchical structures for *reliable multicast distribution* is being undertaken at Carnegie Mellon University [Sripanidkulchai et al]. One such scheme is the *TRAM* protocol [Chiu et al] with a hierarchical tree structure of "repair heads" for repairing failures in transmission to a multicast group. Tree organisation uses advertisement protocols and expanding ring searches and dynamic distributed procedures.

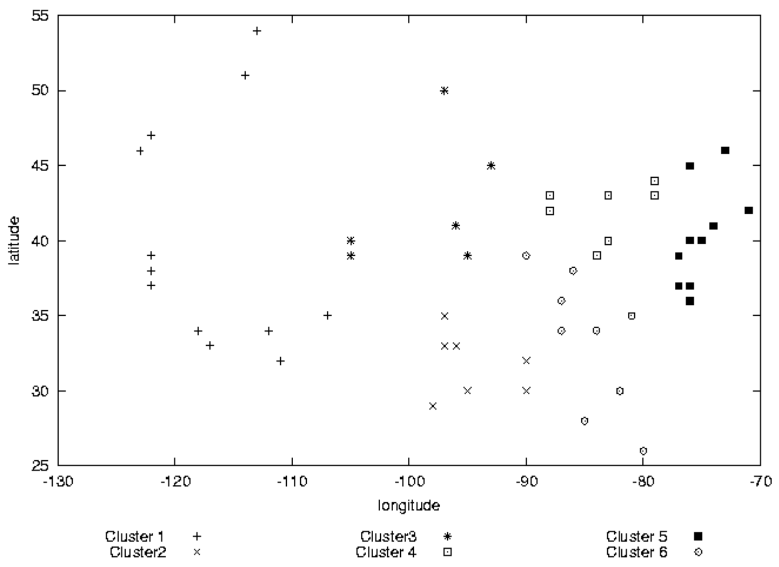
Web caching often uses hierarchical structures; these methods combine a method of detecting well-used pages with decisions about the hierarchy. For example, in the LSAM proxy cache [Touch & Hughes], their Intelligent Request Routing uses a neighbour-search operation to configure the proxy hierarchy. Another example is the Squid Proxy Cache [Squid].

We believe that it will be useful to combine distributed algorithms with the more centralised approach described here. Distribution and searching are best used for finding service-capable portions of the network and for dynamically updating existing hierarchies.

5. Preliminary results

In order to test the applicability of clustering algorithms to networking hierarchies, we have started by looking at suitable hierarchical trees for reliable multicast file distribution. A data set of the geographical location of 51 North American cities has been used.

The figures sow the effects of kmeans clustering to find six clusters in this data set. One immediately obvious effect is the influence of the initial choice of cluster centres on the eventual clusters produced. For figures 1 and 2, initial centres are distributed throughout the range of locations. This produces reasonably even-sized clusters ranging from 6 to 12 members. Figs. 3 and 4 use six initial



centres

Fig.1. Six clusters with initial centers distributed throughout the range

from the South West of the range of locations. Clusters and their centres can and have moved with this procedure, but the result is that the clusters further from the initial centres are large. Cluster sizes range from 2 to 19.

The kmeans algorithm clusters nodes nearer to the given centres than to other centres. It then recalculates the centres and performs the clustering again until a stable solution is found. For Figs 1 and 2, three iterations were needed to achieve convergence, but, because of the movement of the cluster centres for Figs 3 and 4, convergence took 12 iterations.

Figs 2 and 4 show the trees formed from the clusters. Nodes are connected to the node nearest the actual centre of each cluster. The overall centre was found from the cluster centres which were then connected to the nearest city to the overall centre. (In practice, the source of a multicast group would not be located at the overall centre.) For large clusters, sub-clustering can be achieved by using the same technique to as many hierarchical levels as necessary, e.g to maintain a maximum fan-out of six at each level.

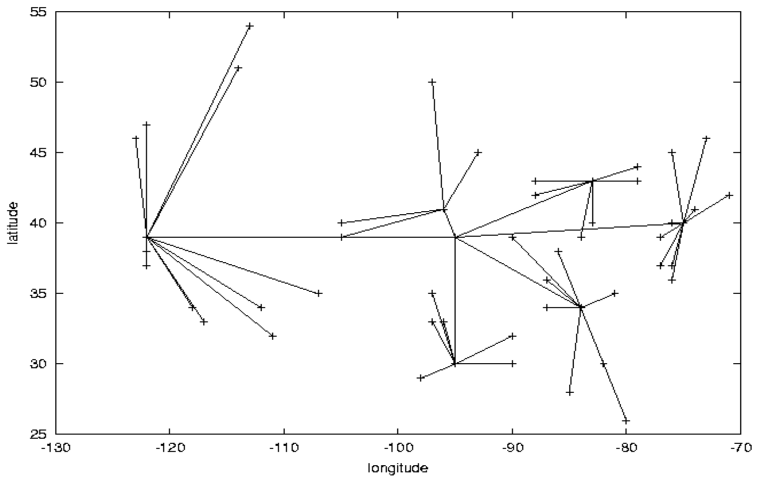


Fig 2. Two-level hierarchical multicast tree from clusters of Fig.1

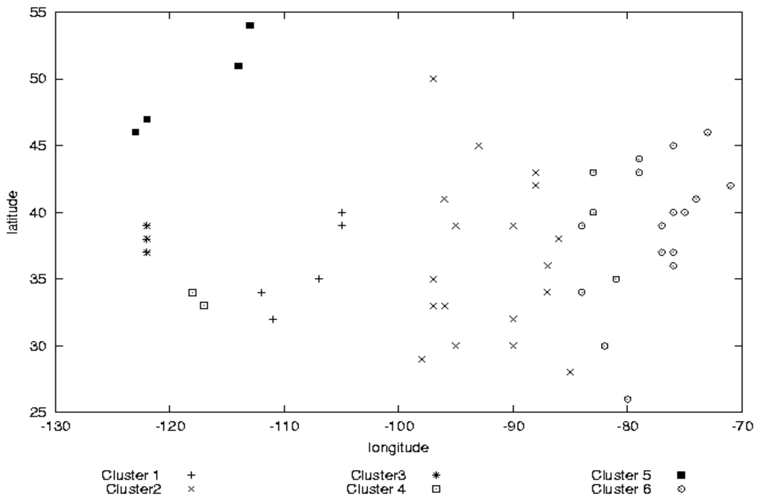


Fig 3. Cluster using six nodes in South West as sarting centres

As a measure of the performance of the hierarchies, we take the total distance covered by the tree found and compare this to the total using a single hierarchical level i.e. connecting each node directly to the overall centre. For the two level tree shown in Fig 4, the total is 262.54, compared with a single level total of 953.96.

From these preliminary results, it can beseen that clustering can be used as a way of organising hierarchies, but more work is needed on finding suitable numbers of clusters, choice of initial centres and on evaluating the results over a wide range of potential groups and multicast membership distributions.

6. Conclusions and further work

The work is beginning to give an insight into the applicability of clusters. It is likely that there will be benefits to support multicasting applications. Applying it to the more general problem of networking hierarchical structures will be investigated.

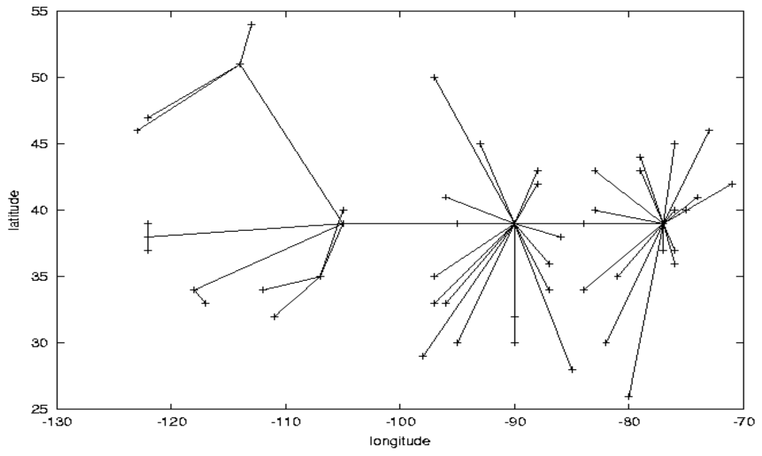


Fig. 4 Hierarchical multicast tree for clusters in Fig. 3

7. References

[Bacelli et al] Bacelli, F., Kofman D. and Rougier J.L., *Self-organising hierarchical multicast trees and their optimization,* Proc. of IEEE INFOCOM 99: Conference on Computer Communications, March 1999, pp 1081-1089

[Cahn] Cahn, Robert S., *Wide Area Network Design: concepts for tools and optimization*, Morgan Kaufmann, 1998

[Callon et al] Callon R., P. Doolan P., Feldman N., Fredette A., Swallow G. and Viswanathan A., *A Framework for Multiprotocol Label Switching*, Internet Draft September 1999. n.b. For the latest document see http://www.ietf.org/ID.html

[Chatterjee & Bassiouni] Chatterjee, S and Bassiouni, MA, *Hierarchical message dissemination in very large WANs,* IEEE Computer Soc. Press, 13-16 Sept 19992, pp 397-403

[Chiu et al] Chiu, D-M, Hurst, S, Kadansky M. and Wesley, J, *TRAM: A tree-based reliable multicast routing protocol*, Sun Microsystems Laboratories technical report SML TR-9896, July 1998

[Francis] Francis, Paul, *Yallcast: Extending the Internet Multicast Architecture*, September 1999, available from http://nttsl.mfeed.ne.jp/francis/yallcast/

[RFC 1633] Braden, R., Clark D. and Shenker S., *Integrated Services in the Internet Architecture: an Overview*, Internet Request for Comments1633 (Informational), June 1994

[RFC 2205] Braden, R., Zhang L., Berson S., Herzog S. and Jamin S., *Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification,* Internet Request for Comments 2205 (Standards Track), September 1997

[RFC 2475] S. Blake S., D. Black, D., Carlson M., Davies E., Wang Z. and Weiss W., *An Architecture for Differentiated Services*, Internet Request for Comments 2475 (Informational), December 1998

[Sharma] Sharma, Subhash, *Applied Multivariate Techniques*, John Wiley and Sons, 1996

[Squid] *The Squid Web Proxy Cache* http://squid.nlanr.net/

[Sripanidkulchai et al] Sripanidkulchai, K, Myers, M and Zhang, H., *Reliable Multicast Recovery Structures*, August 1999, see http://www.cs.cmu.edu/~kunwadee/research/rm-trees/

[Touch & Hughes] Touch, J and Hughes, A S, *LSAM proxy cache: a multicast distributed virtual cache,* Computer Networks and ISDN Systems, 30 (1888) pp 2245-2252