Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 11, pp.41062-41067, November, 2016

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

GRACEFUL GRAPHS AND ITS APPLICATIONS

*Sivaraman, R.

National Awardee for Popularizing Mathematics among masses, Associate Professor of Mathematics
D G Vaishnav College, Arumbakkam, Chennai – 600 106

ARTICLE INFO

Article History:
Received 22nd August, 2016
Received in revised form
15th September, 2016
Accepted 10th October, 2016
Published online 30th November, 2016

Key words:

Graceful Graphs, Edge Graceful Labeling, MPLS, Graceful Numbering, Forwarding Equivalence Class, Label Switched Path.

ABSTRACT

In this Paper, we see about "*Graceful Graphs*" and have discussed the important application of Graceful Graphs in the field of Computer Science, namely, MULTI PROTOCOL LABEL SWITCHING (MPLS) related to Networking. The MPLS technique has many useful applications in networking, for example, it can be effectively used in ATM network infrastructure, as the labeled flows can be mapped to ATM virtual circuit identifiers, and vice-versa. This feature signifies the existence of graceful labeling concept in Computer Networks via MPLS.

Citation: Sivaraman, R. 2016. "Graceful graphs and its applications", *International Journal of Current Research*, 8, (11), 41062-41067.

INTRODUCTION

A **graph** G = (V, E) is a set of vertices V, and a set of edges E that connect pairs of points in V. Edges are actually determined by a symmetric irreflexive relation on V x V. A labeled graph G = {V,E} is a finite series of graph vertices V with a set of graph edges E of 2-subsets of V. Given a graph vertex set $V_n = \{1, 2, \ldots, n\}$, the number of vertex-labeled graphs is given by $2^{n(n-1)/2}$. Two graphs G and H with graph vertices $V_n = \{1, 2, \ldots, n\}$ are said to be isomorphic if there is a permutation P of V_n such that $\{u,v\}$ is in the set of graph edges E(G) iff {p(u),p(v)} is in the set of graph edges E(H).

Graceful graphs

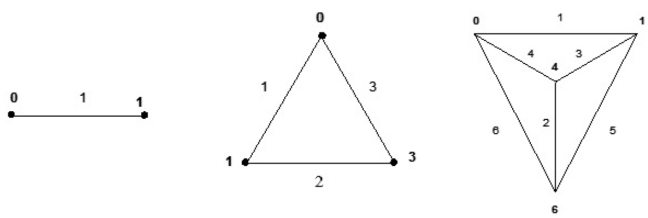
Definition

A labeled graph, which can be "Gracefully Numbered", is called a graceful graph.

In order to achieve this:

1. Label the nodes (vertices) with distinct nonnegative integers (say between 0 and *e* – both inclusive, where *e* is the no of edges)
2. Label the graph edges with the absolute differences between node values. If the graph edge numbers then run from 1 to *e*, the graph is gracefully numbered. In order for a graph to be graceful, it must be without loops or multiple edges.

The following figures provide some examples for Graceful Graphs (which are also complete graphs) that satisfy the conditions stated in our definition.



Edge-Graceful Graph

Given a graph G, we denote the set of edges by E(G) and the vertices by V(G). Let q be the cardinality of E(G) and p be that of V(G). The problem is to find a labeling for the edges such that all the labels form 1 to q are used once and the induced labels on the vertices run from 0 to p-1. In other words, the resulting set for labels of the edges should be (1, 2...q} and (0, 1, . . . , p-1} for the vertices. Once a labeling of the edges is given, a vertex *u* of the graph is labeled by the sum of the

***Corresponding author: Sivaraman, R.**
National Awardee for Popularizing Mathematics among masses, Associate Professor of Mathematics D G Vaishnav College, Arumbakkam, Chennai – 600 106.

labels of the edges incident to it, modulo p. Or, in symbols, the induced labeling on the vertex u is given by

V(*u*) = ∑E(e) mod │V(G)│

where V(*u*) is the label for the vertex and E(e) is the assigned value of an edge incident to u.

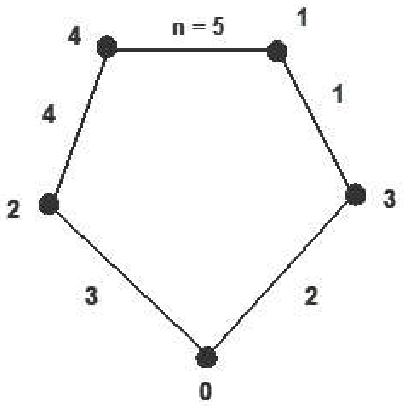
A graph G is said to be edge-graceful if it admits an edge-graceful labeling.

Example

Consider the cycle with three vertices, C_3. This is simply a triangle. One can label the edges 0, 1, and 2, and check directly that, along with the induced labeling on the vertices, this gives an edge-graceful labeling.

In general, C_m is edge-graceful when m is odd and not when m is even.

An edge-graceful labeling of C_5 is shown in the following figure:



The node number corresponding to edges numbered 2 and 3 should be 2 + 3 mod5 = 0. Similarly, the node corresponding to edges numbered 5 and 1 should be 5 + 1 mod 5 = 1. We can similarly fill the other nodes to obtain the edge graceful labeling as shown above. Lo in [ ] has found a necessary condition for any graph to be edge-graceful. By analyzing edge-graceful graphs and the relationship between the vertex labels and the edge labels he discovered that p divides $q^2 + q - \{p(p-1)/2\}$ for all edge-graceful graphs. We now prove this fact.

The sum of the values of all the vertices equals: 1 + 2 + 3 + ... + (p–1)

However, this sum is equal to twice the sum of all the values of the edges mod p, so

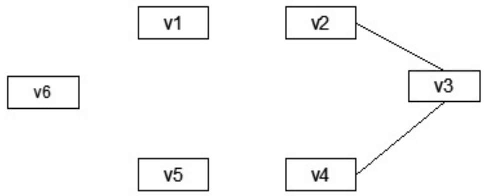
1+ 2 + 3 + ... + (p–1) = 2 (1+ 2 + 3 + ... + q) mod p
which yields p(p–1)/2 = 2(q(q+1)/2 mod p = q^2+q mod p
Therefore, $q^2 + q - \{p(p-1)/2\} = 0$ mod p or p │ $q^2 + q - \{p(p - 1)/2\}$ as desired.

Rosa's approach

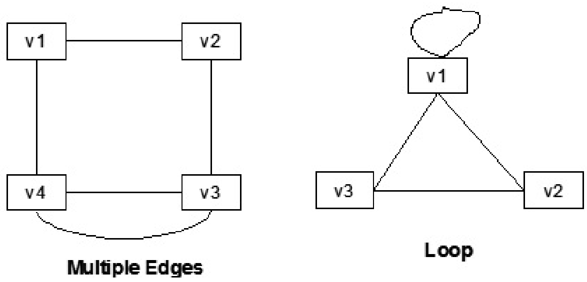
Most graph labeling methods trace their origin to one introduced by Rosa in 1967, or one given by Graham and Sloane in 1980. Rosa called a function *f* a **beta-valuation of a graph G with q edges** if *f* is an injection from the vertices of G to the set {0, 1, . . . q} such that, when each edge <*x*, *y*> is assigned the label │f(x) – f(y)│, the resulting edge labels are distinct.

Rosa has identified essentially three reasons why a graph fails to be graceful. These are shown in the following three cases.

(1) G has "too many vertices" and "not enough edges"

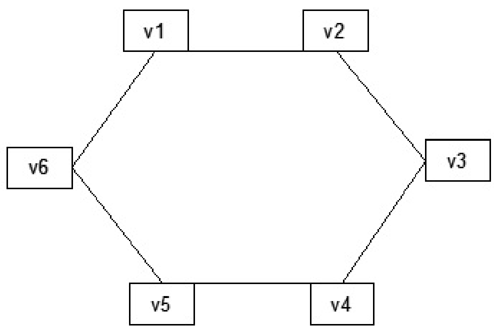


(2) G "has too many edges", that is, the graph "has loops and/or multiple edges"



(3) G Satisfies Parity condition

Rosa [ ] developed a useful parity condition for a simple graph G with e edges. He proved that if every vertex of G has even degree and if e ≡ 1 or 2 (mod 4), then G is not graceful. Rosa's parity condition serves as a sufficient condition for non-graceful graphs.

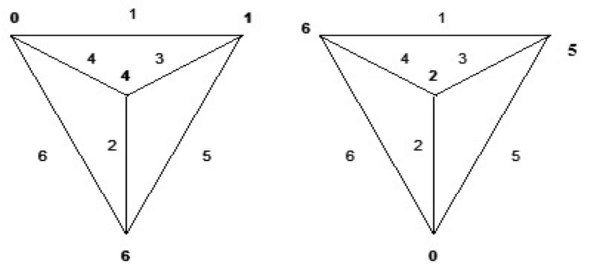


Enumerating All Graceful Labeling Graphs

To enumerate the graceful graphs on N edges, you need to iterate through all possible graceful labeling and then eliminate ones that are isomorphic to labeling already enumerated. The first of these tasks is pretty easy – you just need to count in base N-factorial. The easiest way to do this is to keep a vector of length N, in which member i is between 0 and i inclusive. Start with the all-zero vector, increment the last element until it reaches N-1, then reset it to 0 and increment the next-to-last element (until it reaches N-2, and so on). This approach is illustrated below:

(0,N)
(0, N – 1) (1, N)
(0,N – 1) (1,N – 1) (2, N)
.
.
.
(0, 1) (1,2) ... (N – 1, N)

With this approach, we see that there will be N! Graceful labeling on N edges. One example of the approach provided above is provided below.



Applications of Graceful Graphs in MPLS

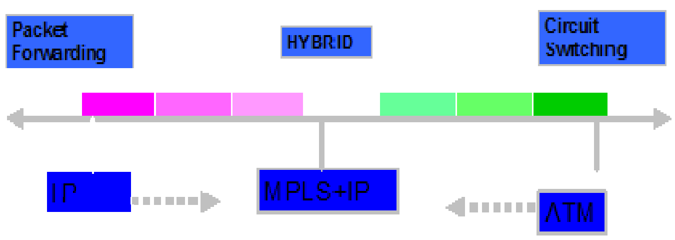
Graceful Label Numbering in Optical MPLS Networks

What is MPLS?

Multi Protocol Label Switching is a routing technique that imitate connection oriented forwarding method in a connectionless (IP) environment

- Uses label native to the media.
- Multi level label substitution transport.
- Hop-by-hop or source routing to establish labels.

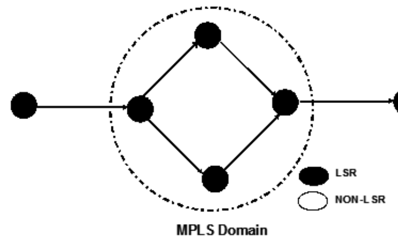
Multi Protocol Label Switching (MPLS) is a data-carrying mechanism which emulates some properties of a circuit-switched network over a packed-switched network. MPLS operates at an OSI Model layer that is generally considered to lie between data link layer and network layer, and thus is often referred to as a “Layer 2.5” protocol. It was designed to provide a unified data-carrying service for both circuit-based clients and packet-switching clients which provide a datagram service model. It can be used to carry many different kinds of traffic, including IP packets.



MPLS is one which,

- Improves the Performance and scalability of a network.
- Explicit routing and traffic engineering (managers the flow pattern of packets) is very efficient (Constraint-based Routing/QoS (based on priority)
- Is used in the Separation of Control (routing) and forwarding is a Virtual Private Network.

How MPLS works?



Carrying the label in the packet

Link Layer header	MPLS “shim” Label header	Network Layer header	Network Layer data

- Each LSR maintains a forwarding table
- typical forwarding table entry

Incoming label	Outgoing label Outgoing interface Next hop address

- Label swapping
- Label format

Label (20 bits)	Experimental (3)	Stack (1)	TTL (8)

MPLS works by prepending packets with an MPLS header contains the ‘label’.

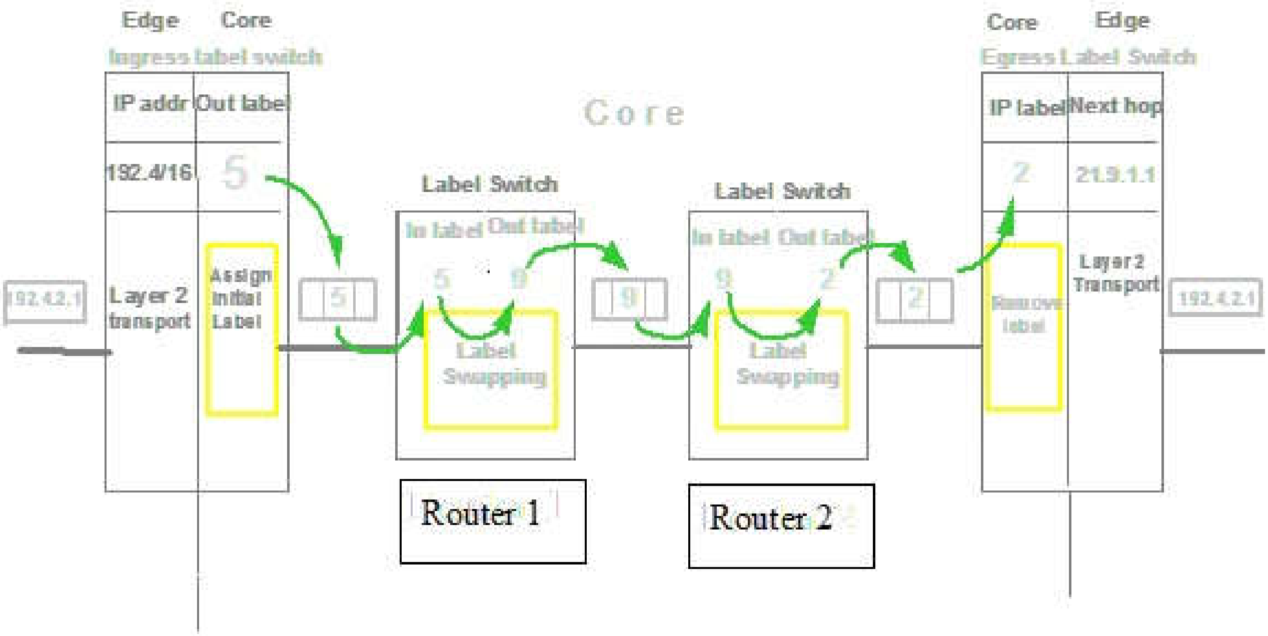
Label entry contains four fields:

- a 20-bit label value.
- a 3-bit field for QoS priority
- a 1-bit bottom of stack flag. If this is set, it signifies the current label is the last in the stack.
- an 8-bit TTL (time to live) field.

These MPLS labeled packets are switched after a Label Lookup/Switch instead of a lookup into the IP table. The exit points of an MPLS network are called Label Edge Routers (LER). Routers that are performing routing based only on Label Switching are called Label Switch Routers (LSR) Penultimate Hop Popping (PHP) is a function performed by certain routers in an MPLS enabled network. It refers to the process whereby the outermost label of an MPLS tagged packet is removed by a Label Switched Router (LSR) before the packet is passed to an adjacent Label Edge Router (LER)

Label Switched Path

The router which first prepends the MPLS header to a packet is called an ingress router. The last router in an LSP, which pops the label from the packet, is called an egress router. Routers in between, which need only swap labels, are called transit routers or Label Switching Routers. When an unlabeled packet enters the ingress router and needs to be passed on to an MPLS tunnel, the router first determines the forwarding equivalence class the packet should be in, and then inserts one (or more) labels in the packet’s newly created MPLS header. The packet is then passed on to the next hop router for this tunnel. When a labeled packet is received by an MPLS router, the topmost



label is examined. Based on the contents of the label a *swap, push* (impose) or pop (dispose) operation can be performed on the packet's label stack. Routers can have prebuilt lookup tables that tell them which kind of operation to do based on the topmost label of the incoming packet so they can process the packet very quickly. In a ***swap*** operation the label is swapped with a new label, and the packet is forwarded along the path associated with the new label. In a **push** operation a new label is pushed on top of the existing label, effectively "encapsulating" the packet in another layer of MPLS. This allows the hierarchical routing of MPLS packets. Notably, this is used by MPLS VPNs. In a **pop** operation the label is removed from the packet, which may reveal an inner label below. This process is called "decapsulation". If the popped label was the last on the label stack, the packet "leaves" the MPLS tunnel. These are usually done by the egress router, but see PHP below.

During these operations, the contents of the packet below the MPLS Label stack are not examined. Indeed transit routers typically need only to examine the topmost label on the stack. The forwarding of the packet is done based on the contents of the labels, which allows "protocol independent packet forwarding" that does not need to look at a protocol-dependent routing table and avoids the expensive IP longest prefix match at each hop. At the egress router, when the last label has been popped, only the payload remains. This can be an IP packet, or any of a number of other kinds of payload packet. The egress router must therefore have routing information for the packet's payload, since it must forward it without the help of label lookup tables. An MPLS transit router has no such requirement. In some special cases, the last label can also be popped off at the penultimate hop (the hop before the egress router). This is called Penultimate Hop Popping (PHP). This may be interesting in cases where the egress router has lots of packets leaving MPLS tunnels, and thus spends inordinate amounts of CPU time on this. By using PHP, transit routers connected directly to this egress router effectively offload it, by popping the last label themselves.

MPLS can make use of existing ATM network infrastructure, as its labelled flows can be mapped to ATM virtual circuit identifiers, and vice-versa.

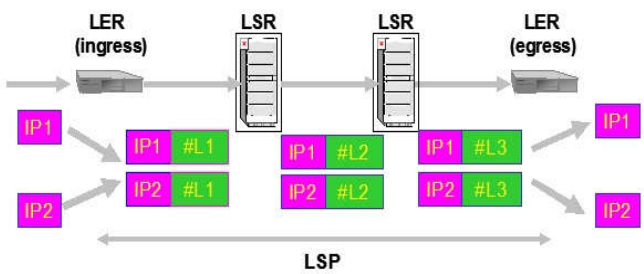
Forwarding Equivalence Class (FEC)

It is certainly possible for each flow to have its own set of labels through the subnet. However, it is more common for routers to group multiple flows that end at a particular router or LAN and use a single for them. The flows that are grouped together under a single label are said to belong to the same FEC (Forwarding Equivalence Class). This class covers not only where the packets are going, but also their service class because all their packets are treated

FEC/label binding mechanism

- Binding is done once at the ingress LER/OXC (in MPlambdaS)
- Label binding is based on destination IP address

Forwarding Equivalence Classes (FEC)



Flow of IP packets

- Over the same path
- Treated in the same manner
- Mapped to the same label

MPLS Signaling Protocols

- Hop-by-hop & Explicit
- Label Distribution Protocol (LDP)
- Constraint-based Routing LDP (CR-LDP)
- Extensions to RSVP

Benefits of using MPLS

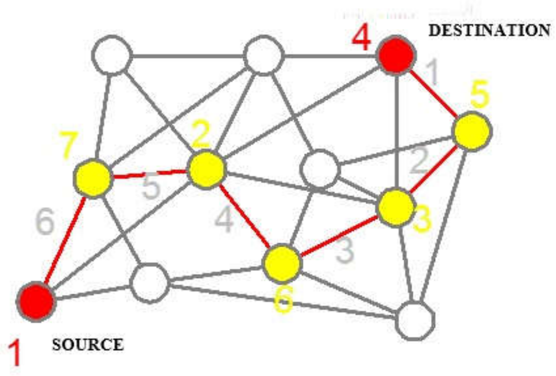
- Forwarding packets between different types of networks is simplified to a great extent, because there is a single forwarding algorithm irrespective of the routing or link layer protocol used.
- Switching based on short labels is an advantage over long headers.

Graceful Numbering of Graphs is used in MPLS for the following reasons

- It is a well-known labeling method in graphs.
- Number the nodes of a graph G with 1, 2... e so that induce edge labels computed by absolute node number differences are all distinct.

MPLS unicast with graceful numbering

The following diagram shows the connection of MPLS with graceful numbering.



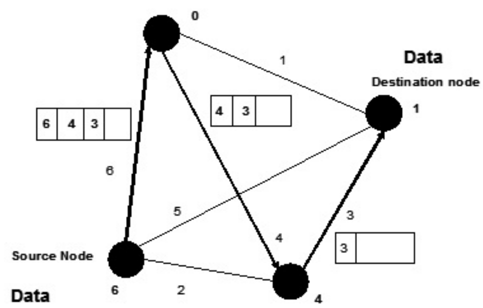
MPLS combined with Graceful Numbering

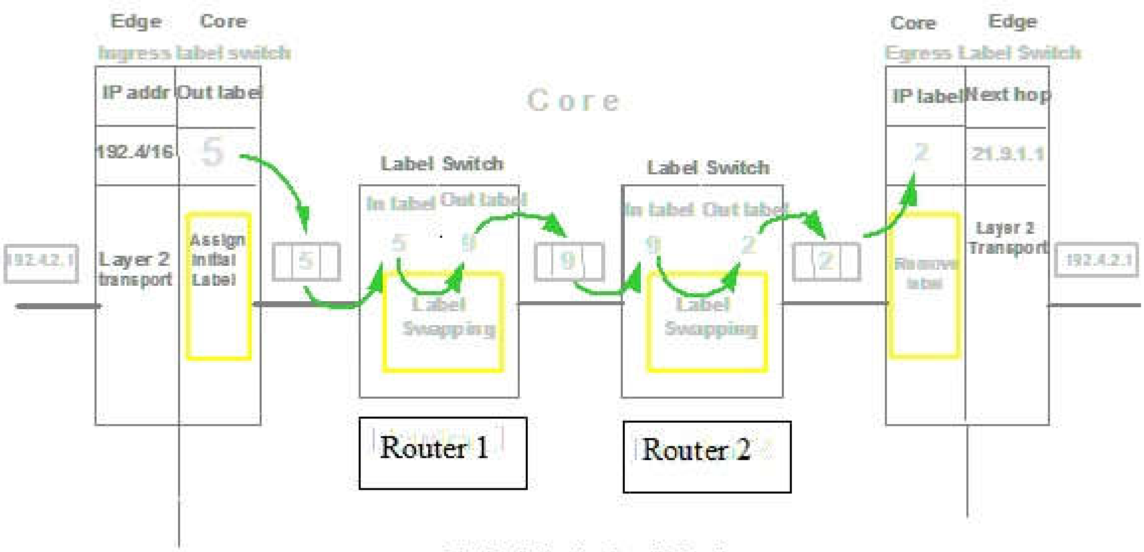
By imposing the graceful labeling concept, we can eliminate the label switching operation. Here the whole LAN Network is considered as the graph and they are labelled graceful. And the labelled stack is maintained by each node in the LAN Network. Labelling is automatically done when the system on the network is booted. MPLS works by prepending packets with an MPLS header, containing one or more 'labels'. This is called a label stack.

Each label stack entry contains four fields:

- a 20-bit label value.
- a 3-bit field for QoS priority
- a 1-bit *bottom of stack* flag. If this is set, it signifies the current label is the last in the stack.
- an 8-bit TTL (time to live) field.

A Packet may carry an entire stack of labels with it. Once it passes through an edge whose corresponding label will be **stripped** at each router. An example for packet transmission in MPLS domain is as follows





Label Switched Path

Lab el	Ex p	S= 0	TT L	Lab el	Ex p	s= 0	TT L	Lab el	Ex p	s= 1	TT L	IP Header	TCP header	Payload

Packet transmission in MPLS domain using graceful labelling

The current label with **S** bit is 1 then it is the last edge to reach the destination node. This feature is mostly used in virtual private network.

Conclusion

Graceful graphs are easy to understand, but have many seemingly simple unresolved questions that no one has been able to answer yet. Graceful graphs lend themselves naturally to computational approaches. As discussed in our paper, Graceful graphs have wider scope for implementation mainly in the field of Networking and with respect to MPLS protocol, Graceful labeling is highly useful in the forwarding of packets with greater efficiency due to following reasons:

1. Label swapping operation is not needed.
2. Provides an efficient technique for labeling a Network.
3. Forwarding table is needed
4. Existence of protocol independent packet forwarding.

REFERENCES

Aldred, R. E. L.; McKay, Brendan D. 1998. "Graceful and harmonious labellings of trees", Bulletin of the Institute of Combinatorics and its Applications, 23: 69–72.

Gallian, Joseph A. 1998. "A dynamic survey of graph labeling", Electronic Journal of Combinatorics, **5**: Dynamic Survey 6, 43 pp. (389 pp. in 18th ed.)

Graceful Graphs and Graceful Labelings: Two Mathematical Programming Formulations and Some Other New Results, Timothy A. Redl, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, January 2003.

Horton, Michael P. 2003. Graceful Trees: Statistics and Algorithms

Rekhter; Y. and R. Aggarwal (January 2007), RFC 4781: Graceful Restart Mechanism for BGP with MPLS.

Rosa, A. 1967. "On certain valuations of the vertices of a graph", Theory of Graphs (Internat. Sympos., Rome, 1966), New York: Gordon and Breach, pp. 349–355.

Virginia Vassilevska, "Coding and Graceful Labeling of trees". SURF 2001.

****