Internet Engineering Task Force
Internet Draft
Expires in October, 2002
draft-ietf-ospf-scalability-01.txt

Gagan L. Choudhury
Vera D. Sapozhnikova
AT&T

Anurag S. Maunder
Sanera Systems

Vishwas Manral
Netplane Systems

April, 2002

Explicit Marking and Prioritized Treatment of Specific IGP Packets for Faster IGP Convergence and Improved Network Scalability and Stability

Status of this Memo

Abstract

In this draft we propose the following mechanisms in order to allow fast IGP convergence and at the same time maintain scalability and stability of a network:

(1) Explicitly mark Hello packets, to differentiate them from other IGP packets, so that efficient implementations can detect and process the Hello packets in a priority fashion.

(2) In the absence of special marking, or in addition to it, use other mechanisms in order not to miss Hello packets. One example is to treat any packet received over a link as a surrogate for a Hello packet for the purpose of keeping the link alive.

(3) The same type of explicit marking and prioritized treatment may be beneficial to other IGP packets as well. Some examples include (a) LSA acknowledgment packet, (b) Database description (DBD) packet from a slave that is used as an acknowledgement, and (c) LSAs carrying intra-area topology change information.

It is possible that some implementations are already using one or more of the above mechanisms in order not to miss the processing of critical packets during periods of congestion. However, we suggest the above mechanisms to be included as part of the standard so that all implementations can benefit from them.

Table of Contents

1. Motivation

The motivation of this draft is to address the following two key objectives of any data network: (a) Fast restoration under failure conditions, and (b) Improved network scalability and stability. Using analytic and simulation models we show that in general the two objectives are in conflict, i.e., improvement in one usually results in the degradation of the other. However, special marking and prioritized processing of certain key messages can allow us to achieve both objectives.

The first item we address is fast restoration. The theoretical limit for link-state routing protocols to re-route is in link propagation time scales, i.e., in tens of milliseconds. However, as pointed

out in [Ref1], in practice it may take from seconds to tens of seconds to detect the link failure and disseminate this information to the network followed by the convergence on the new set of paths. This is an inordinately long period of transient time for mission critical traffic destined to the non-reachable nodes of the network.

One component of the long re-route time is the link failure detection time of between 20 and 30 seconds through typically three missed Hello packets with the typical hello interval of 10 seconds (between 30 and 40 seconds if missed hello threshold is 4). This component would be much shorter in the presence of link level detection, but as pointed out in [Ref1] it does not work in some cases. For example, a device driver may detect the link level failure but fail to notify it to the IGP level. Also, if a router fails behind a switch in a switched environment then even though the switch gets the link level notification it cannot communicate that to other routers. Therefore for faster reliable detection at the IGP level, one has to reduce the hello interval. [Ref1] suggests that this be reduced to below a second, perhaps even to tens of milliseconds. A second component of the long re-route time is delayed SPF (shortest-path-first) computation. The typical delay value is between 1 and 5 seconds but in order to have sub-second rerouting it needs to be reduced significantly.

The second item we address is the ability of a network to withstand the simultaneous or near-simultaneous update of a large number of link-state-advertisement messages, or LSAs. We call this event, an LSA storm. An LSA storm may be initiated due to many reasons. Here are some examples:

(a) one or more link failures due to fiber cuts,

(b) one or more node failures for some reason, e.g., software crash or some type of disaster in an office complex hosting many nodes,

(c) requirement of taking down and later bringing back many nodes during a software/hardware upgrade,

(d) near-synchronization of the once-in-30-minutes refresh instants of some types of LSAs,

(e) refresh of all LSAs in the system during a change in software version.

In addition to the LSAs generated as a direct result of link/node failures, there may be other indirect LSAs as well. One example in ATM/MPLS networks is LSAs generated at other links as a result

of significant change in bandwidth resulting from rerouting of virtual circuits that went down during the link/node failure. The LSA storm tends to drive the node CPU utilization to 100% for a period of time and the duration of this period increases with the size of the storm and the node adjacency, i.e., the number of links connected to it. During this period the Hello packets received at the node would see high delays and if this delay exceeds the Router-Dead Interval (typically 30-40 seconds or three to four hello intervals) then the associated link would be declared down.

In this draft we address only the issue of links being declared down due to the delayed processing of Hello messages, but in general, depending on the implementation, there may be other impacts of a long CPU busy period. For example, in a reliable node architecture with an active and a standby processor, a processor switch-over may result during an extended CPU-busy period which may mean that all the adjacencies would be lost and need to be re-established. A processor switch-over may also result from a memory-exhaust caused by an extended CPU busy period. Both of the above events would cause more database synchronization with neighbors and network-wide LSA flooding which in turn might cause extended CPU-busy periods at other nodes. This may cause unstable behavior in the network for an extended period of time and potentially a meltdown in the extreme case.

Due to world-wide increased traffic demand, data networks are ever increasing in size. As the network size grows, a bigger LSA storm and a higher adjacency at certain nodes would be more likely and so would increase the probability of unstable behavior. One way to address the scalability issue is to divide the network hierarchically into different areas so that flooding of LSAs remains localized within areas. However, this approach increases the network management and design complexity and may result in less optimal routing between areas. Also, unless addresses are aggregated, a large number of summary LSAs may need to be flooded. Thus it is important to allow the network to grow towards as large a size as possible under a single area.

The undesirable impact of large LSA storms is understood in the networking community and it is well known that large scale flooding of control messages (either naturally or due to a bug) has been responsible for several network events in the past causing a meltdown or a near-meltdown. For some recent examples see [Ref2-Ref5]. Recently, proposals have been submitted to reduce flooding overhead in case more than one interface goes to the same neighbor [Ref6,Ref7]. Also, [Ref8-Ref9] considers a wide range of congestion control and failure recovery mechanisms.

Section 2 uses a simulation model to illustrate the onset of instability in the network as the result of a large LSA storm. Section 3 uses a simple, approximate but easy-to-understand analytic model to make the point that reducing hello intervals and more frequent SPF computation would in fact reduce network scalability and stability. Section 4 makes the point that many of the underlying causes of network scalability can be avoided if certain IGP messages are specially marked and provided prioritized treatment. [Ref10] also provides simulation and analytic models to show the onset of instability in large networks due to LSA storms and proposes the prioritization of Hello and other special packets to improve scalability and stability.

2. Simulation Study

We have developed a network-wide event simulation model to study the impact of an LSA storm. It captures the actual congestion seen at various nodes and accounts for propagation delay between nodes, retransmissions in case an LSA is not acknowledged, failure of links for LSAs delayed beyond the Router-dead interval, and link recovery following database synchronization and LSA flooding once the LSA is processed. It approximates a real network implementation and uses processing times that are roughly in the same order of magnitude as measured in the real network (of the order of milliseconds). There are two categories of IGP messages processed at each node in the simulation. Category 1 messages are triggered by a timer and include the Hello refresh, LSA refresh and retransmission packets. Category 2 messages are not triggered by a timer and include received Hello, received LSA and received acknowledgments. Timer-triggered messages are given non-preemptive priority over the other type. As a result, the received Hello packets and the received acknowledgment packets may see long queuing delays under intense CPU overload.

Table 1 below shows sample results of the simulation study when applied to a network with about 300 nodes and 800 links. The node-adjacency varies from node to node and the maximum node-adjacency is 30. The Hello interval is assumed to be 5 seconds, the minimum interval between successive SPF (Shortest-Path-First) calculations is 1 second, and the Router-Dead Interval is 15 seconds, i.e., a link is declared down if no Hello packet is received for three successive hello intervals. During the study, an LSA storm of size X is created at instant of time 100 seconds where storm-size is defined as the number of LSAs generated during a storm. Three cases are considered with X = 300, 600 and 900 respectively. Besides the storm, there are also the normal once-in-thirty-minutes LSA refreshes. At any given point of time we define a quantity "dispersion" that is the number of LSU packets already generated in the network but not received and processed in at least one node (each LSU packet is assumed to carry three LSAs).

Table 1 plots dispersion as a function of time and thereby identifies the impact of LSA storm on network stability.

Table 1: DISPERSION as a FUNCTION of TIME (in sec) for different LSA Storm Sizes

LSA STORM SIZE	100s	106s	110s	115s	140s	170s	230s	330s	370s
300	0	39	3	1	0	1	0	0	0
600	0	133	120	100	12	1	0	0	0
900	0	230	215	196	101	119	224	428	488

Before the LSA storm, the dispersion due to normal LSA refreshes remains small. We expect the dispersion to jump to a high value right after the storm and then come down to the pre-storm level after some period of time (this happens with X=300 and X=600 but not with X=900). In Table 1 with a LSA storm size 300, the "heavy dispersion period" lasted about 11 seconds and no link losses were observed. With a LSA storm of size 600, the "heavy dispersion period" lasted about 40 seconds. Some link losses were observed a little after 15 seconds within the "heavy dispersion period" but eventually all links recovered and the dispersion came down to the pre-storm level. With a LSA storm of size 900, the "heavy dispersion period" lasted throughout the simulation period (6 minutes).

The generic observations are as follows:

(1) If the initial LSA storm size (e.g., X=300) is such that the delays experienced by Hello packets are not big enough to cause any link failures anywhere in the network, the network remains stable and quickly gets back to a period of "low dispersion". These types of LSA storms are observed quite frequently in operational networks, from which the network easily recovers.

(2) If the initial LSA storm size (e.g., X=600) is such that the delays experienced by a few Hello packets in a few nodes cause link failures then some secondary LSA storms are generated. However, the secondary storms do not keep growing indefinitely and the network remains stable and eventually gets back to a

period of "low dispersion". This type of LSA storm was observed in an operational network triggered by a network upgrade, from which the network recovered but with some difficulty.

(3) If the initial LSA storm size (e.g., X=900), is such that the delays experienced by many Hello packets in many nodes cause link failures then a wave of secondary LSA storms are generated. The network enters an unstable state and the secondary storms are sustained indefinitely or for a very long period of time. This type of LSA storm was observed in an operational network triggered by a network failure [Ref2] from which the network recovered only after taking some corrective steps (manual procedures based on reducing adjacencies at heavily congested nodes were used to reduce LSA flooding and stabilize the network).

The results show that there is a LSA storm threshold above which the network shows unstable behavior. It was also observed that if Hello packets (both received and sent) are given higher priority compared to other IGP packets then the LSA storm threshold above which network shows unstable behavior is significantly increased. In this draft we only look at the failure of links due to missed Hellos, but in general there may be many other types of failures once a network enters an unstable state. Examples of failures include memory exhaust and shooting down of the node processor due to the inability of performing certain critical jobs.

3. Analytic Model for Delay experienced by a Hello Packet During an Initial LSA Storm

From the simulation results of the previous section it is clear that it is important to identify the delay experienced by a Hello packet during an initial LSA storm and compare that against the maximum allowed delay so as not to declare the link down. We develop a simple and approximate analytic model for this purpose and use it to study the impact of Hello and SPF intervals on network stability. As explained in Section 2, for every link interface, a node has to send and receive a Hello packet once every hello interval. Sending of a Hello packet is triggered by a timer. We assume that higher priority is given to timer-triggered jobs and therefore no significant delay is experienced in the sending of Hello packets. However, a received Hello packet cannot be easily distinguished from other IGP packets and therefore we assume that it is served in a first-come-first-served fashion. Let's assume:

S = Size of LSA storm, i.e., the number of LSAs in it. Also, it is assumed that each LSA is carried in one LSU packet.
L = Link adjacency of the node under consideration.

t1 = Time to send or receive one IGP packet over an interface (the same time is assumed for Hello, LSA, duplicate LSA and LSA acknowledgment even though in general there may be some differences. However, this would be a good approximation if majority of the time were in the act of receiving or sending and a relatively small part for packet-type-specific work.) In the numerical examples we assume t1 = 1 ms.

t2 = Time to do one SPF calculation. For large networks, this time is usually in hundreds of ms and in the numerical examples we assume t2 = 200 ms.

Hi = Hello interval (the gap between successive Hello messages on the same link).

Si = Minimum interval between successive SPF calculations.

ro = Rate at which non-IGP work comes to the node (e.g., forwarding of data packets). For the numerical examples we assume ro = 0.2.

T = Total work brought in to the node during the LSA storm. For each LSA update generated elsewhere, the node will receive one new LSA packet over one interface, send an acknowledgment packet over that interface, and send copies of the LSA packet over the remaining L-1 interfaces. Also, assuming that the implicit acknowledgment mechanism is in use, the node will subsequently receive either an acknowledgment or a duplicate LSA over the remaining L-1 interfaces. So over each interface one packet is sent and one is received. It can be seen that the same would be true for self-generated LSAs (see Table 1 for an example). So the total work per LSA update is 2*L*t1. Since there are S LSAs in the storm, we get

T = 2*S*L*t1   (1)

In Equation (1) we ignore retransmissions of LSAs in case acknowledgments are not received or processed within 5 seconds. From the simulation study we see that this is a reasonable assumption since usually only a few retransmissions result during the processing of the initial LSA storm (usually retransmissions happen at a higher rate during the secondary storms).

T2 = Time period over which the work comes. Due to differences in propagation times and congestion at other nodes, it is possible for the work arrival time to be spread out over a long interval. However, since we are primarily interested in a few nodes that are bottlenecks or near-bottlenecks, it is reasonable to assume that most of the work comes in one chunk. We verified this to be usually true using simulations. One part of T2 will be of the order of link propagation delay and we assume that there is a second part which is proportional to T. Therefore we get,

T2 = A + B*T (2)

Where A and B are constants. For the numerical examples we assume A = 10 ms and B = 0.1.

D = Maximum delay experienced by a Hello packet during the LSA storm. We assume first-come-first-served service and hence the delay seen by the Hello packet would be the total outstanding work at the node at the arrival instant plus its own processing time. We assume that the outstanding work steadily increases over the interval T2 and so the maximum delay is seen by a Hello packet that comes near the end of this interval. We write down an approximate expression for D and then explain the various terms on the right hand side:

D = T - T2 + max(1,2*T2/Hi)*t1 + max(1,T2/Si)*t2 + ro*T2 (3)

The first term is the total work brought in due to the LSA storm. The second term is the work the node was able to finish since we are assuming that it was continuously busy during the period T2. The third term is the total work due to the sending and receiving of Hello packets during the period T2. Note that it is assumed that at least one Hello packet is processed, i.e., itself. The fourth term is due to SPF processing during the period T2 and we assume that at least one SPF processing is done. The last term is the total non-IGP work coming to the node over the interval T2.

Dmax = Maximum allowed value of D, i.e., if D exceeds this value then the associated link would be declared down. In the numerical examples below we assume

Dmax = 3*Hi (4)

If we assume that the previous Hello packet was minimally delayed then exceeding Dmax really means four missed hellos since the Hello packet under study itself came after a period Hi. In the numerical examples below, both D and Dmax change with choice of system parameters and we are mainly interested in identifying if D exceeds Dmax. For this purpose we define the following ratio variable

Delay Ratio = D/Dmax (5)

and identify if Delay Ratio exceeds 1.

In Tables 2-4 we plot the Delay Ratio as a function of LSA Storm size with node adjacencies 10, 20 and 50 respectively. All parameters except for the ones noted explicitly on the Tables are as stated earlier. Table 2 assumes Hello packets every 10 seconds and SPF calculation every 5 seconds, which are typical default values

today. With a node adjacency of 10, the Delay Ratio is below 1 even with an LSA storm of size 900. However, with a node adjacency of 20, the Delay Ratio exceeds 1 at around a storm of size 800 and with a node adjacency of 50, the Delay Ratio exceeds 1 at around a storm of size 325.

Table 2: Ratio of Hello Packet Delay to Maximum Allowed Hello Packet Delay as a function of LSA Storm Size (LSS) (Hello Every 10 Seconds, SPF Every 5 Seconds, Dmax = 30 seconds)

NODE Adjacency	LSS=100	LSS=300	LSS=500	LSS=700	LSS=900
10	0.0677	0.1904	0.3131	0.4358	0.5584
20	0.1291	0.3744	0.6198	0.8651	1.1104
50	0.3131	0.9264	1.5398	2.1558	2.7718

In a large network it is not unusual to have LSA storms of size several hundreds since the LSA database size may be several thousands. This is particularly true if there are many Autonomous-System-External (ASE) LSAs and there are special LSAs for carrying information about available bandwidth at links as is common in ATM networks and might be used in MPLS-based networks as well. Table 3 decreases the hello interval to 2 seconds and SPF calculation is done once a second. LSA storm thresholds are significantly reduced. Specifically, with a node adjacency of 10, the Delay Ratio exceeds 1 at around a storm of size 310; with a node adjacency of 20, the Delay Ratio exceeds 1 at around a storm of size 160; and with a node adjacency of 50, the Delay Ratio exceeds 1 at around a storm of size only 65.

Table 3: Ratio of Hello Packet Delay to Maximum Allowed Hello Packet Delay as a function of LSA Storm Size (LSS) (Hello Every 2 Seconds, SPF Every 1 Second, Dmax = 6 seconds)

NODE ADJACENCY	LSS=30	LSS=90	LSS=150	LSS=210	LSS=270
10	0.124	0.308	0.492	0.676	0.86
20	0.216	0.584	0.952	1.32	1.691
50	0.492	1.412	2.349	3.289	4.229

Table 4 decreases the hello interval even further to 300 ms and SPF calculation is done once every 500 ms. LSA storm thresholds are really small now. Specifically, with a node adjacency of 10, the Delay Ratio exceeds 1 at around a storm of size 40, with a node adjacency of 20, the Delay Ratio exceeds 1 at around a storm of size 20, and with a node adjacency of 50, the Delay Ratio is already over 1 even with a storm of size 10.

Table 4: Ratio of Hello Packet Delay to Maximum Allowed Hello Packet Delay as a function of LSA Storm Size (LSS) (Hello Every 300 ms, SPF Every 500 ms, Dmax = 900 ms)

NODE ADJACENCY	LSS=10	LSS=30	LSS=50	LSS=70	LSS=90
10	0.419	0.828	1.237	1.646	2.055
20	0.623	1.441	2.259	3.078	3.896
50	1.237	3.282	5.333	7.467	9.602

Based on the simulation observations we understand that if Delay Ratio is less than 1 for all Hello packets then the system is stable and if it exceeds 1 at many nodes then the system tends to enter an unstable region. Therefore, the LSA storm threshold at which the Delay Ratio exceeds 1 may also roughly be considered as the network stability threshold. Tables 2-4 show that the stability threshold rapidly decreases as the hello interval and SPF computation interval decreases. One reason for this is the increased CPU work due to more frequent hello and SPF computations, but the dominant reason is that Dmax itself decreases and so a smaller CPU busy interval is needed to exceed it. Specifically, Dmax is 30 seconds in Table 2, 6 Seconds in Table 3 and only 900 ms in Table 4. It is clear from the above examples that in order to maintain network stability as the hello interval decreases, it is necessary to provide faster prioritized treatment to received Hello packets which can of course be only done if those packets can be distinguished from other IGP packets.

4. Need for Special Marking and Prioritized Treatment of Specific IGP packets

The analytic and simulation models show that a major cause for unstable behavior in networks is received Hello packets at a node getting queued behind other work brought in to the node during an LSA storm and missing the deadline of typically three or four hello intervals. Clearly, if the Hello packet can be specially marked to distinguish it from other IGP packets then they can be given prioritized treatment and they would not miss the deadline even during a large LSA storm. However, the key is that the detection mechanism should be significantly faster than the complete processing of an IGP packet and it should be possible to do detection and separate queueing at the line rate.

Usually a special Diffserv codepoint is used to differentiate all IGP packets from other packets. We propose a separate Diffserv codepoint for Hello packets that allows them to be queued separately from other IGP packets and given prioritized treatment.

We also suggest the use of additional mechanisms in order not to miss Hello packets during periods of congestion and thereby avoid declaring links to be down. One such mechanism is to treat any packet received over a link as an implicit Hello packet for the purpose of keeping the link alive. Under this mechanism a link will be declared down only if no packets are received over the link for a duration of the Router Dead interval. So, during a period of congestion, if Hello packets are queued behind LSAs or some other packets but at least one such packet is received over the link no slower than once every Router Dead interval, the link will stay up.

Besides the Hello packets there may be other IGP packets that could also benefit from special marking and prioritized treatment. We give some examples below but clearly others are possible.

(1) One example is the LSA acknowledgment packet. This packet disables retransmission and if a large queueing delay to this packet expires the retransmission timer (typical default value is 5 seconds) then a needless retransmission will happen causing extra traffic load. A special marking and prioritization of the

LSA acknowledgment packet would eliminate many needless retransmissions. During the database exchange process between neighbours following a link coming up, Database Description packets are exchanged and the successful receipt of such a packet is acknowledged by sending a properly sequenced Database Description packet back to the sender. Since these packets are used as acknowledgments, it makes sense to properly mark and prioritize them as well.

(2) Another example is an LSA carrying a change information. It is preferable to transmit this information faster than other LSAs in the network that are just once-in-30-minutes refreshes.

Among "change" LSAs we can distinguish further and give preferential treatment to only those "change" LSAs that carry intra-area topology change information as opposed to other "change" LSAs that are summary LSAs or Opaque LSAs. We can also distinguish between "change" LSAs carrying "bad" information (node/link failure) versus those carrying "good" information (node/link coming up) and give higher priority to LSAs carrying "bad" information. There may be multiple levels of priority depending on the relative importance of the various IGP packets.

The explicit identification can also be used for preferentially triggering the SPF calculation. We can normally have a longer gap between successive SPF calculations, but revert to a shorter gap after receiving an LSA that carries a area-topology-change information. This will speed up restoration time following a failure but would not unduly increase the SPF processing overhead.

5. Summary

In this draft we point out that if a large LSA storm is generated as a result of some type of failure/recovery of nodes/links or synchronization among refreshes then the Hello packets received at a node may see large queueing delays and miss the deadline of typically three or four hello intervals. This causes the associated link to be declared down, starts a secondary storm and is potentially the beginning of unstable behavior in the network. This is already a concern in today's network but would be a bigger concern if the hello interval and the minimum interval between SPF calculations are substantially reduced (below or perhaps well below a second) in order to allow faster rerouting. To avoid the above, we propose the following:

(1) Explicitly mark Hello packets to differentiate them from other IGP packets so that efficient implementations can detect and act upon these packets in a priority fashion. This may be done by

using a special Diffserv codepoint for Hello packets (separate from that used for other IGP packets).

(2) In the absence of special marking or in addition to it, other mechanisms should be used in order not to miss Hello packets. One example is to treat any packet received over a link as a surrogate for a Hello packet for the purpose of keeping the link alive.

(3) The same type of explicit marking and prioritized treatment would also help other IGP packets and should be considered. Some examples include LSA acknowledgment packets, Database Description packets from the slave during database exchange and LSAs carrying intra-area topology change information. LSAs carrying bad news (node/link failures) may also be given priority over LSAs carrying good news (node/link coming back up).

It is possible that some implementations are already using one or more of the above mechanisms in order not to miss the processing of critical packets during periods of congestion. However, we suggest the above mechanisms to be included as part of the standard so that all implementations can benefit from them.

6. Acknowledgments

The authors would like to acknowledge several people for their helpful comments. In AT&T we recognize Tushar Amin, Jerry Ash, Margaret Chiosi, Elie Francis, Jeff Han, Tom Helstern, Shih-Yue Hou, S. Kandaswamy, Beth Munson, Aswatnarayan Raghuram, Moshe Segal, John Tinacci, Mike Wardlow and Pat Wirth. In Lucent Technologies we recognize Nabil Biter and Roshan Rao.

7. References

[Ref1] C. Alaettinoglu, V. Jacobson and H. Yu, "Towards Milli-second IGP Convergence," Work in Progress.

[Ref2] Pappalardo, D., "AT&T, customers grapple with ATM net outage," Network World, February 26, 2001.

[Ref3] "AT&T announces cause of frame-relay network outage," AT&T Press Release, April 22, 1998.

[Ref4] Cholewka, K., "MCI Outage Has Domino Effect," Inter@ctive Week, August 20, 1999.

[Ref5] Jander, M., "In Qwest Outage, ATM Takes Some Heat," Light Reading, April 6, 2001.

[Ref6] A. Zinin and M. Shand, "Flooding Optimizations in Link-State Routing Protocols," Work in Progress.

[Ref7] J. Moy, "Flooding over Parallel Point-to-Point Links," Work in progress.

[Ref8] J. Ash, G. Choudhury, J. Han, V. Sapozhnikova, M. Sherif, M. Noorchashm, S. Mcallister, A. Maunder, V. Manral, "Proposed Mechanisms for Congestion Control / Failure Recovery in OSPF & ISIS Networks" Work in Progress.

[Ref9] J. Ash, G. Choudhury, V. Sapozhnikova, M. Sherif, A. Maunder, V. Manral, "Congestion Avoidance & Control for OSPF Networks", Work in Progress.

[Ref10] G. Choudhury, A. Maunder and V. Sapozhnikova, "Faster Link-State IGP Convergence and Improved Network Scalability and Stability," Presentation at LCN 2001, Tampa, Florida, November 14-16, 2001.

8 Authors' Addresses

Gagan L. Choudhury
AT&T
Room D5-3C21
200 Laurel Avenue
Middletown, NJ, 07748
USA
Phone: (732)420-3721
email: gchoudhury@att.com

Vera D. Sapozhnikova
AT&T
Room C5-2C29
200 Laurel Avenue
Middletown, NJ, 07748
USA
Phone: (732)420-2653
email: sapozhnikova@att.com

Anurag S. Maunder
Sanera Systems
370 San Aleso Ave.
Second Floor
Sunnyvale, CA 94085
Phone: (408)734-6123
email: amaunder@sanera.net

Vishwas Manral
NetPlane
189, Prashasan Nagar,
Road Number 72
Jubilee Hills, Hyderabad
India
email: Vishwasm@netplane.com