COMMITTEE T1 – TELECOMMUNICATIONS
T1X1.5 **T1X1.5/2001-097**
Boulder, CO., March 26 - 28, 2001

CONTRIBUTION TO T1 STANDARDS PROJECT

TITLE **A rough analysis of the control traffic pattern in an optical environment**

SOURCE **Sudheer Dharanikota, Raj Jain, Jay Shah** **Nayna Networks Inc.**
Yong Xue, Curtis Brownmiller **WorldCom**
2400 N. Glenville Dr.
Richardson, TX. 75082
CONTACT **Raj Jain**

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

PROJECT **Optical Hierarchical Interfaces**

ABSTRACT

In this document we perform a rough analysis of the control traffic in a typical optical network. To being with, we provide only an intra-domain traffic analysis, with OSPF as an IGP (Interior Gateway Protocol), RSVP as a signaling protocol and LMP as the link management protocol.

Key words: IGP, OSPF, RSVP, LMP, Control Traffic

Notice

This Document has been prepared to assist Standards Committee T1X1. It is offered to the committee as a basis for discussion and is not a binding proposal on Nayna Networks Inc. or WorldCom. Information presented in this document may be subject to change after more study. Nayna Networks Inc. and WorldCom specifically reserve the right to add to, amend, or to withdraw the statements contained herein.

* CONTACT: Curtis Brownmiller, Curtis.Brownmiller@wcom.com, 972-729-7171, 972-729-7261

1. Introduction

The protocols that constitute control plane traffic are:

1. Element management protocols
2. Link management protocols
3. Routing protocols and
4. Signaling protocols

Traffic over a network can be grouped into three categories:

- Traffic pattern during the node initialization,
- Traffic pattern during stable conditions, and
- Traffic pattern during the failure conditions.

The analysis is divided into the following groups for an incremental analysis:

- Intra-domain (or intra-area) communication
- Inter-domain (or inter-area) communication and
- Inter-AS communication

1.1 Reference model

A basic reference model required for this analysis is presented in Figure 1. Refer to the appendix for the detailed calculation of the protocol packet sizes. Note: The traffic in this document is measured from a node's perspective (to avoid duplicate counting of protocol data) as outgoing and incoming traffic, which account for east bound and west bound traffic.

1.1.1 Notation

- N – Edge nodes to a given domain
- M – Average number of peers for every node (Like capable elements)
- C – Average number of clients for every node (Customers)
- TE_L – Average number of TE links between the two neighbors
- D_L – Average number of data bearing links in a TE link

- T_r – Keep alive time between routing neighbors
- T_s – Keep alive time between signaling neighbors
- T_l – Keep alive time between link management neighbors

1.1.2 Assumptions

- EML (Element Management Layer) and NMS (Network Management System) traffic is not considered in this version of the document.
- We consider (to begin with) only OSPF as the IGP routing protocol, RSVP as the signaling protocol and LMP, DWDM LMP as the link management protocols.
- To be relevant for the optical domain, we consider GMPLS scenario with link bundling concept and with LMP between the nodes.

- We only consider the intra-domain case to begin with in this document.
- One node comes up or goes down at a given time (long enough for the network to reach stable condition after this event).
- We do not consider the retransmission of the packets, as this spreads the traffic across time, instead of increasing the peak of the traffic, and hence, no modification to the bandwidth requirements.
- It is assumed that all the data can be fit into one MTU in the following discussion. This assumption will be relaxed in the next releases of the document.
- A broadcast medium is assumed between the routers in the network, to identify the maximum possible traffic.
- In this analysis, we do not consider nodal load.
- We assume that the nodal processing is negligible.

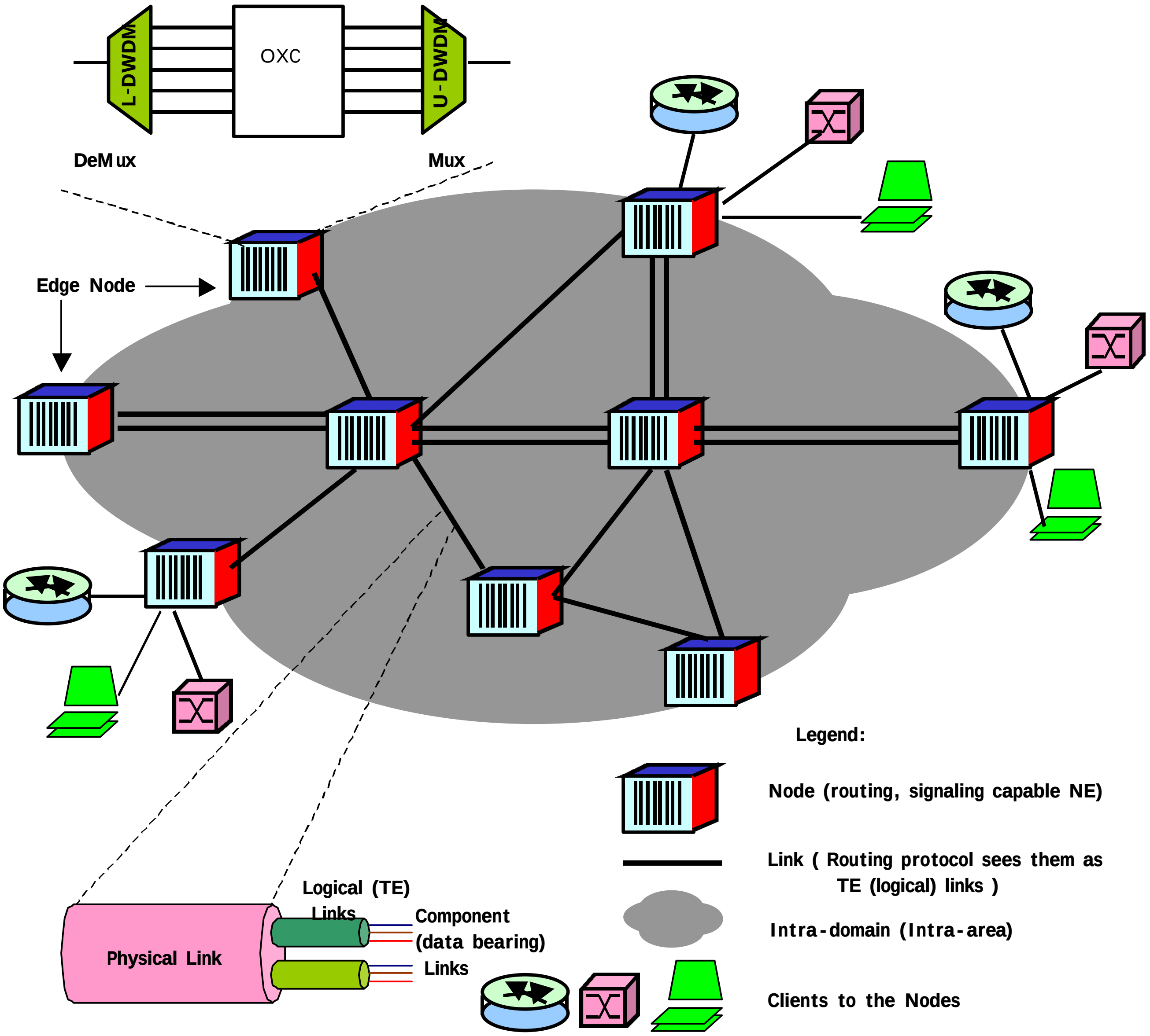
L-DWDM
OXC
U-DWDM
DeMux
Mux
Edge Node
Legend:
Node (routing, signaling capable NE)
Link (Routing protocol sees them as TE (logical) links)
Intra-domain (Intra-area)
Logical (TE) Links
Component (data bearing) Links
Physical Link
Clients to the Nodes

Figure 1 A reference optical network model for the discussion in this document

2. Summary of traffic pattern

The following is the approach used for arriving at the reasonable numbers:

1. Calculate different protocol data packet lengths for the routing, signaling and link management protocols.
2. Divide the traffic into the three conditions as mentioned earlier into initialization time, stable condition, and failure condition cases.
3. Further group the traffic in each condition to approximate the traffic that can go in parallel.
4. Find traffic in bytes for each case with certain realistic assumptions on the variables.
5. Make guess on the spread of the traffic with the help of default timer values for each of the protocols.
6. Find the final numbers for a given topological assumptions, which is presented in the following table.

In the following table, we present the amount of traffic being generated under different topological assumptions. The numbers are derived by using the analysis presented in the appendices as referred in the table.

Assumptions	Traffic during Initialization (Ref. to Appendix 10.2)		Traffic during stable conditions (Ref. to Appendix 10.3)		Traffic during failure conditions (Ref. To Appendix 10.4)			
	Outgoing	Incoming	Outgoing	Incoming	Outgoing		Incoming	
					Node Failure	Link Failure	Node Failure	Link Failure
Small size M = N = L = TE_L = D_L =								
Medium size M = N = L = TE_L = D_L =								
Large size M = N = L = TE_L = D_L =								

3. Future work

Extend this work to:

- Inter-area and inter-AS scenarios,
- Include other routing and signaling protocols,
- Include traffic due to element management protocols and applications, and
- Extrapolate this work to complex topologies and scenarios.

4. Acknowledgements

The authors would like to acknowledge Vikram Rautela, Toni Liu and Soumitra Dan for verifying the numbers for us.

5. References

[OSPF] J. Moy, "OSPF Version 2," RFC1247.

[RSVP] R. Braden et al., "Resource ReserVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205.

[LMP] J. P. Lang, et al., "Link Management Protocol (LMP)," IETF working group document, draft-ietf-mpls-lmp-02.txt, an IETF working group draft.

6. Protocol packet sizes

6.1 OSPF

Abbreviation	Message type	Number of bytes
CMH	Common message header	24
LSH	Link state header (LSH)	20
RLLSA	Router link LSA	LSH + 4 + {(# of links) * (12 + [4 * (# of ToS reported)]) }
NLLSA	Network link LSA	LSH + 4 + (# of attached routers) * 4
SLSA	Summary LSA	LSH + 4 + (# of ToS reported) * 8
ASELSA	AS External LSA	LSH + 4 + (# of ToS reported) * 12 + 12
Hpkt	Hello packet	CMH + 20 + (# of valid neighbors) * 4
DDpkt	Database description packet	CMH + 8 + (# of LSAs) * 20
LSRpkt	Link state request packet	CMH + (# of LSAs) * 16
LSUpkt	Link state update packet	CMH + 4 + [LSA Length 1 + …]
LSACKpkt	Link state acknowledgement packet	CMH + (# of LSAs) * 20

Note: TE stuff (link capability information) need to be added.

6.2 RSVP

Note: All objects are for IPV4. IPV6 will be considered later.

Abbreviation	Message type	Number of bytes
CMH	Common message header	8
GOBJH	General object header	4
SOBJ	Session Object (1)	GOBJH + 8 = 12
HOBJ	Hop object (3)	GOBJH + 8 = 12
IOBJ	Integrity object (4)	GOBJH + 28 = 32
TVOBJ	Time values object (5)	GOBJH + 4 = 8
ERROBJ	Error specification object (6)	GOBJH + 8 = 12
SCPOBJ	Scope object (7)	GOBJH + N * 4
STYOBJ	Style object (8)	GOBJH + 4 = 8
CLFOBJ	Controlled load flow specification object (9)	GOBJH + 32 = 36
GSFOBJ	Guaranteed service flow specification object (9)	GOBJH + 44 = 48
FSPECOBJ	Filter specification object (10)	GOBJH + 8 = 12
STEMPOBJ	Sender template object (11)	GOBJH + 8 = 12
STSPECOBJ	Sender Tspec object (12)	GOBJH + 32 = 36
ADSPECOBJ	ADSPEC object (13)	
POBJ	Sample policy object (14)	GOBJH + 4 + L = 12 + L
CONFOBJ	Confirmation object (15)	GOBJH + 4 = 8
Ppkt	PATH packet	CMH + (4) + (1) + (3) + (5) + [(14)] + (11) + (12) + [(13)] = 140 + L + [(13)]
PERRpkt	PATH ERR packet	CMH + (4) + (1) + (6) + (14) +(11) + (12) + [(13)] = 132 + L + [(13)]
Rpkt	RESV packet	CMH + (4) + (1) + (3) + (5) + [(15)] + [(7)] + [(14)] + (8) + (# of filters) * ((9) + (12)) = 112 + N * 4 + L + n * (60)
RERRpkt	RESV ERR packet	CMH + (4) + (1) + (3) + (6) + [(7)] + [(14)] + (8) + (# of error filters) * ((9) + (12)) = 100 + 4 * N + L + m * 60
RCONFpkt	RESV CONF packet	CMH + (4) + (1) + (6) + (15) + (8) + (# of filters) * ((9) + (12)) = 84 + n * 60
PTEARpkt	PATH TEAR packet	CMH + (4) + (1) + (3) + (11) + (12) + [(13)] = 132 + L + (13)

RTEARpkt	RESV TEAR packet	CMH + (4) + (1) + (3) + [(7)] + (8) + (# of filters) * ((9) + (12)) = 76 + N * 4 + n *(60)
MIDpkt	MESSAGEID packet	
SREFRESHpkt	SREFRESH packet	

6.3 LMP/DWDMLMP

Abbreviation	Message type	Number of bytes
CMH	Common Message Header	12
AB	Authentication block	24
TLVH	LMP TLV Header	4
HCTLV	Hello configuration TLV	TLVH + 4 = 8
CAPTLV	LMP capability TLV	TLVH + 4 = 8
TELTLV	TE Link TLV	TLVH + 12 = 16
DLTLV	Data-link TLV	TLVH + 12 + DLSUBTLV = 16 + DLSUBTLV
DLSUBTLV	Data-link sub TLV	None yet
FCTLV	Failed channel TLV	TLVH + 4 * # of LCIDs
ACTTLV	Active channel TLV	TLVH + 4 + 4 * # of LCIDs
BSpkt	Bootstrap packet	CMH + 24 = 36
CONFpkt	CONFIGURE packet	CMH + 8 + (HCTLV + CAPTLV+ ..) = 36 + ..
CONFACKpkt	CONFIGURE ACK packet	CMH + 16 = 28
CONFNACKpkt	CONFIGURE NACK packet	CMH + 16 + TLVs = 28 + TLVs
Hpkt	HELLO packet	CMH + 8 = 20
LSUMpkt	LINK SUMAMRY packet	CMH + 4 + (# TELTLV) * 16 + (# DLTLV) * 16
LSUMACKpkt	LINK SUMAMRY ACK packet	CMH + 20 = 32
LSUMNACKpkt	LINK SUMMARY NACK packet	CMH + 8 + (# TELTLV) * 16 + (# TELTLV * # DLTLV) * 16
BVpkt	BEGIN VERIFY packet	CMH + 28 = 40
BVACKpkt	BEGIN VERIFY ACK packet	CMH + 16 = 28
BVNACKpkt	BEGIN VERIFY NACK packet	CMH + 12 = 24
TESTpkt	TEST packet	CMH + 8 = 20
TSUCCpkt	TEST SUCCESS packet	CMH + 16 = 28
TFAILpkt	TEST FAILURE packet	CMH + 8 = 20
TACK	TEST STATUS ACK packet	CMH + 8 = 20
EVpkt	END VERIFY packet	CMH + 8 = 20
EVACKpkt	END VERIFY ACK packet	CMH + 8 = 20
CFAILpkt	Channel Fail packet	CMH + 8 + (# failed Chs) * FCTLV

CFACKpkt	Channel fail ACK packet	CMH + 8 = 20
CFNACKpkt	Channel fail NACK packet	CMH + 8 + (# failed Chs) * FCTLV
CACTpkt	Channel active packet	CMH + 8 + (# of active TLVs) * ACTTLV
CACTACKpkt	Channel active ACK packet	CMH + 8 = 20

6.4 Others

7. Initialization time traffic analysis

When an optical node comes up it involves itself in the following activities:

1. Element management protocols (TBD)
    a. Configuration download
2. Link management protocol traffic
    a. Boot strapping
    b. Link association
3. Routing protocols
    a. Neighbor discovery
    b. Topology discovery
    c. Link capability discovery
4. Signaling protocols
    a. Neighbor discovery
    b. Session maintenance

7.1 Traffic due to link management protocols

7.1.1 Boot strapping

The messages exchanged during this phase are:

- Bootstrap packets (BSpkt)
- Configure (CONFpkt), configure acknowledgement (CONFACKpkt), configure negative acknowledgement (CONFNACKpkt)
- Hello packets (Hpkt)

The traffic generated by these packets is due to:

- Bootstrap packets exchanged by the new node with the other existing nodes
- Configuration exchanges with the neighbors
- Initial hello packet generation for keep alive

Amount of traffic going out = M * BSpkt + M * CONFpkt + M * Hpkt
= M * 92 bytes

Amount of traffic coming in = M * BSpkt + M * CONFACKpkt + M * Hpkt
= M * 84 bytes

7.1.2 Link association

The messages exchanged during this phase are:

- Link summary (LSUMpkt), link summary acknowledgement (LSUMACKpkt) or negative acknowledgement (LSUMNACKpkt)
- Begin verification (BVpkt) and end verification packets (EVpkt) and their acknowledgements (BVACKpkt, EVACKpkt)

The traffic generated by these packets is due to:

- Exchanging link validation and verification related messages

Amount of traffic going out = M * LSUMpkt + [M * TE_L * BVPkt]+ [M * TE_L * EVpkt]
= M * 16 * (1 + TE_L * D_L) + [M * TE_L * 60] bytes

Amount of traffic coming in = M * LSUMACKpkt + [M * TE_L * BVACKpkt]
+ [M * TE_L * EVpkt]
= M * 32 + [M * TE_L * 48] bytes

7.2 Traffic due to routing protocols

7.2.1 Neighbor discovery

The messages exchanged during this phase are:

- Hello packets (Hpkt)

The traffic generated by these packets is due to:

- Hello packet by source to all the OSPF capable routers
- Hello responses by the destinations to the new neighbor

Amount of traffic going out = Hpkt = 48 bytes

Amount of traffic coming in = M * Hpkt = M (44 + 4 * M) bytes

7.2.2 Topology discovery

Assumption: Node under reference is assumed to be a slave and its peers are assumed to be masters.

The messages exchanged during this phase are:

- Database description packets (DDpkt), Link state request packets (LSRpkt), link state update packets (LSUpkt) and Link state acknowledgement packets (LSACKpkt)

The traffic generated by these packets is due to:

- Database synchronization packet(s) (Master –> Slave)
- Link state request packet(s) (Slave -> Master)
- Link state update packets (Master -> Slave)

- Link state acknowledgement packets (Slave -> Master)

Amount of traffic going out = M * {LSRpkt (L LSAs) + LSACKpkt (L LSAs)}
$= M * (24 + 16 * L) + M * \{(24 + 20 * \Sigma_{n = 1..5}(L_n))\}$ bytes
Amount of traffic coming in = M * {DDpkt (L LSAs) + LSUpkt (L LSAs)}
$= M * \{(32 + L * 20) + (28 + \Sigma_{n = 1..5}(L_n * N_n))\}$ bytes

Where:
n = LSA type
L_n = Length of LSA type n
N_n = Number of LSAs of type n

The LSA types are:

- Router LSA
- Network link LSA
- Summary LSA
- AS external LSA

Out of these, in an intra-area case, we can consider traffic due to only the router LSAs. Other traffic can be assumed null, especially in an overlay model.

On router LSA:

With M routing peers, TE_L links between each routing peers, and assuming 5 ToS declarations between the routers the length of the router LSA can be determined as $L_{routerLSA} = \{TE_L * (12 + [4 * 5])) = (TE_L * 32)$ per routing peer. And the number of such router LSAs is M.

The above summations will become:

Amount of traffic going out= $M * (24 + 16 * TE_L) + M * (24 + 20 * (TE_L * 32))$ bytes
Amount of traffic coming in= $M * \{(32 + TE_L * 20) + (28 + (TE_L * 32) * M)\}$ bytes

7.2.3 Link capability discovery (TBD)

Note:

Link TLV	2
LINK_TYPE_SUB_TLV	2
LINK_ID_SUB_TLV	4
LINK_OUTD_SUB_TLV	4
LINK_INID_SUB_TLV	4
LINK_PROTECTION_SUB_TLV	4
LINK_DESCRIPTOR_SUB_TLV	8
LINK_SRLG_SUB_TLV	16
LOCAL_IP	4
REMOTE_IP	4
TE_METRIC	4
MAX_BW	4

MIN_BW	4
UNRESERV_BW	4
RESRC_CLASS/COLOR	4

The messages exchanged during this phase are the:

- Link TE capability packets (Opaque LSAs)

7.3 Traffic due to signaling protocols

7.3.1 Neighbor discovery (not applicable to RSVP)

7.3.2 Session maintenance (not applicable to RSVP)

8. Stable condition traffic analysis

When an optical node is in the stable condition, the following are the activities it involves itself with:

1. Element management protocols (TBD)
    a. Changes in the configuration activity
2. Link management protocol
    a. Keep alive messages
    b. Changes in the link association
    c. Exchanging performance monitored information
3. Routing protocols
    a. Keep alive activity
    b. Communicating new link(s) availability
    c. Communicating the changes in the link capability
4. Signaling protocols
    a. Signaling protocol activity for connection management
    b. Refreshing the existing connections (RSVP)

8.1 Traffic due to link management protocols

8.1.1 Keep alive messages

The messages exchanged during this phase are:

- Hello packets (Hpkt)

The traffic generated by these packets is due to:

- Hello packet generation for keep alive

Amount of traffic going out = M * Hpkt = M * 20 bytes

Amount of traffic coming in = M * Hpkt = M * 20 bytes

8.1.2 Changes in the link association

The messages exchanged during this phase are:

- Link summary, link summary acknowledgement or negative acknowledgement
- Link verification process related messages

The traffic generated by these packets is due to:

- Exchanging changes to the link validation and verification related messages

Amount of traffic going out = LSUMpkt + [TE_{LC} *BVPkt]+ [TE_{LC} *EVpkt]
= 16 * (1 + TE_{LC} * D_{LC}) + [TE_{LC} * 60] bytes

Amount of traffic coming in = LSUMACKpkt + [TE_{LC} *BVACKkt]+ [TE_{LC} *EVpkt]
= 32 + [TE_{LC} * 48] bytes

Where:

TE_{LC} – Number of TE links changes
D_{LC} – Average number of data bearing links that are affected

8.1.3 Exchanging performance monitored information (TBD)

8.2 Traffic due to routing protocols

8.2.1 Keep alive activity

The messages exchanged during this phase are:

- Hello packets

The traffic generated by these packets is due to:

- Hello packet by source to all the OSPF capable routers

Amount of traffic going out = Hpkt = (44 + 4 * M) bytes
Amount of traffic coming in = M * Hpkt = M (44 + 4 * M) bytes

8.2.2 Communicating new link(s) availability

The messages exchanged during this phase are:

- Link state update packets

The traffic generated by these packets is due to:

- Link state update packets (Master -> Slave)
- Link state acknowledgements (Slave -> Master)

Amount of traffic coming in = M * {LSUpkt (L LSAs)}
= $M * \{(28 + \Sigma_{n = 1..5}(L_n * Nc_n))\}$ bytes

Where:

n = LSA type
L_n = Length of LSA type n
Nc_n = Number changed of LSAs of type n

Amount of traffic going out = M * {LSACKpkt (L LSAs))
= $M * \{(24 + 20 * \Sigma_{n = 1..5}(L_n))\}$

- TE_{CL} – Average number of TE links change per second
- D_{CL} – Average number of data bearing links change per TE link

As we assumed in the previous condition, we ignore LSAs other than router LSA. With this in consideration the above traffic changes to:

Amount of traffic coming in = $M * (28 + (D_{CL} * 32) * TE_{CL})$ bytes

Amount of traffic going out = $M * (24 + 20 (TE_L * 32))$ bytes

- TE_{CL} – Average number of TE links change per second
- D_{CL} – Average number of data bearing links change per TE link

8.2.3 Communicating the changes in the link capability (TBD)

Need to add opaque LSA stuff here.

8.3 Traffic due to signaling protocols

8.3.1 Signaling protocol activity for connection management

The messages exchanged during this phase are:

- Path (Ppkt), Resv (Rpkt) and Resv Confirmation (RCONFpkt) messages

The traffic generated by these packets is due to:

- Path messages to add new connections
- Resv messages to add new connections and a confirmation to the resv

Amount of traffic coming in = $M * (\Sigma_{1..M} Nm) * Rpkt$
= $M * (\Sigma_{1..M} Nm) * (112 + N * 4 + L + n * 60)$

Nm = Number of connection request for the mth neighbor
L = Length of the policy objects for the path
n = Average number of filters per reservation = 2 (reasonable assumtption)

Amount of traffic going out = $N * (\Sigma_{1..M} Nm) * (Ppkt + RCONFpkt)$

$$= N * (\Sigma_{1\,..\,M}\, Nm) * (140 + L + 84 + n * 60)$$
$$= N * (\Sigma_{1\,..\,M}\, Nm) * (224 + L + n * 60)$$

Nm = Number of connection request for the M^{th} neighbor
L = Average length of the policy objects for the path = 100 bytes (off the air assumption)

If we assume on average there are C clients to an edge node as shown in Figure 1, and each client makes CR number of average connection requests then the above equations become:

$$\text{Amount of traffic coming in} = N * C * CR * (112 + N * 4 + 100 + 2 * 60)$$
$$= N * C * CR * (332 + N * 4) \text{ bytes}$$

$$\text{Amount of traffic going out} = N * C * CR * (224 + 100 + 2 * 60)$$
$$= N * C * CR * 444 \text{ bytes}$$

8.3.2 Refreshing the existing connections (RSVP)

The messages exchanged during this phase are:

- Path and Resv messages

The traffic generated by these packets is due to:

- Path messages to refresh existing connections
- Resv messages to refresh existing connections

$$\text{Amount of traffic coming in} = M * (\Sigma_{1\,..\,M}\, Nm) * Rpkt$$
$$= N * (\Sigma_{1\,..\,M}\, Nm) * (112 + N * 4 + L + n * 60)$$
$$= N * C * CR * (412 + N * 4)$$

Amount of traffic going out = $M * (\Sigma_{1\,..\,M}\, Nm) * Ppkt$
$$= N * (\Sigma_{1\,..\,M}\, Nm) * (140 + L)$$
$$= N * C * CR * 240$$

9. Failure condition traffic analysis

When an optical node or a link is down then the following are the activities that create traffic on the network:

1. Element management protocols (TBD)
2. Link management protocol
    a. Keep alive messages
    b. Fault reporting traffic
    c. Switching over activity
3. Routing protocols
    a. Keep alive messages
    b. Topological synchronization
4. Signaling protocols

a. Propagating the connection failures
b. Rerouting/ reestablishing the connections

Note: The routing hello packets or signaling refresh mechanisms detect node failure. It cannot be detected by the link management protocols.

9.1 Traffic due to link management protocols

9.1.1 Link failure

9.1.1.1 Keep alive messages

The messages exchanged during this phase are:

- Hello packets (Hpkt)

The traffic generated by these packets is due to:

- Hello packet generation for keep alive

Amount of traffic going out = M * Hpkt = M * 20 bytes

Amount of traffic coming in = M * Hpkt = M * 20 bytes

9.1.1.2 Fault reporting traffic

The messages exchanged during this phase are:

- Channel fail and channel fail acknowledgement packets

The traffic generated by these packets is due to:

- Exchanging link failure messages related to the link(s) unavailability

Amount of traffic going out = CFAILpkt = 20 + (TE_{FL} * (4 + 4 * (D_L))) bytes

Where:

TE_{FL} = Number of failed TE links
D_L = Average number of component links per failed TE link

Amount of traffic coming in = CFACKpkt = 20 bytes

9.1.1.3 Switching over activity (TBD)

9.1.2 Node failure

Node failure is not detected by the link management protocols.

9.1.2.1 Keep alive messages

The messages exchanged during this phase are:

- Hello packets (Hpkt)

The traffic generated by these packets is due to:

- Hello packet generation for keep alive

Amount of traffic going out = M * Hpkt = M * 20 bytes

Amount of traffic coming in = M * Hpkt = M * 20 bytes

9.2 Traffic due to routing protocols

9.2.1 Link failure

9.2.1.1 Keep alive activity

The messages exchanged during this phase are:

- Hello packets

The traffic generated by these packets is due to:

- Hello packet by source to all the OSPF capable routers

Amount of traffic going out = Hpkt = (44 + 4 * M) bytes
Amount of traffic coming in = M * Hpkt = M (44 + 4 * M) bytes

9.2.1.2 Topological synchronization

The messages exchanged during this phase are:

- Link state update, acknowledgement packets

The traffic generated by these packets is due to:

- Link state update packets (Detected node -> Other neighbors)
- Link state acknowledgements (Neighbors -> Detected node)

Amount of traffic coming in = M * {LSUpkt (L LSAs)}
$= M * \{(28 + \Sigma_{n = 1..5}(L_n * TE_{FL}))\}$ bytes

Where:
n = LSA type
L_n = Length of LSA type n
TE_{FL} = Number of failed LSAs of type n

Amount of traffic going out = M * {LSACKpkt (L LSAs))
= M * {(24 + 20 * $\Sigma_{n = 1..5}(L_n)$)}

With the router LSA assumption this will become:

Amount of traffic coming in = M * {(28 + (TE_L * 32) * TE_{FL}))} bytes

Amount of traffic going out = M * {(24 + 20 * (TE_{FL} * 32) bytes

9.2.2 Node failure

9.2.2.1 Keep alive activity

The messages exchanged during this phase are:

- Hello packets

The traffic generated by these packets is due to:

- Hello packet by source to all the OSPF capable routers

Amount of traffic going out = Hpkt = (44 + 4 * M) bytes
Amount of traffic coming in = M * Hpkt = M (44 + 4 * M) bytes

9.2.2.2 Topological synchronization

The messages exchanged during this phase are:

- Link state update, acknowledgement packets

The traffic generated by these packets is due to:

- Link state update packets (Detected node -> Other neighbors)
- Link state acknowledgements (Neighbors -> Detected node)

Amount of traffic coming in = M * M * {LSUpkt (L LSAs)}
= M * M * {(28 + $\Sigma_{n = 1..5}(L_n * TE_{FL})$)} bytes

Where:
n = LSA type
L_n = Length of LSA type n
TE_{FL} = Number of failed LSAs of type n

Amount of traffic going out = M * M * {LSACKpkt (L LSAs))
= M * M * {(24 + 20 * $\Sigma_{n = 1..5}(L_n)$)}

9.3 Traffic due to signaling protocols

No extra traffic due to RSVP as it does behave exactly as it is during the stable conditions. So we just add the stable condition traffic here.

Amount of traffic going out = {N * C * CR * (744 + N * 8)} bytes

Amount of traffic coming in = {N* C * CR * 844} bytes

10. Total traffic

10.1 Spread factor calculation

In the following table we capture different times (default values from the MIBs) that helps in spreading the traffic.

Traffic condition	LMP	RSVP	OSPF
Initialization	Hello interval (T_{LH}) = 1 sec	Path/Resv refresh interval (T_{RR}) = 10 sec	Hello interval (T_{OH}) = 10 sec
Stable	Link summary timeout interval (T_{LT}) = 2 sec	- ditto -	Update timeout (T_{OU}) = 5 sec
Fault	- ditto -	- ditto -	- ditto -

Spread factor per phase will be the inverse of the time that is considered above for the protocol.

10.2 During initialization

Total traffic = Traffic due to link management protocols + traffic due to routing protocols + traffic due to signaling protocols

Traffic going out = $\{M * 92\} + \{M * 16 * (1 + TE_L * D_L) + [M * TE_L * 60]\}$
$+ \{48\} + M * (24 + 16 * TE_L) + M * (24 + 20 * (TE_L * 32))$

$= M * (92 + 16 * (61 + TE_L * D_L + TE_L)) + 48$
$+ M * (48 + 16 * TE_L + 20 * TE_L * 32))$

Traffic coming in = $\{M * 84\} + \{M * 32 + [M * TE_L * 48]\}$
$M (44 + 4 * M) + M * \{(32 + TE_L * 20) + (28 + (TE_L * 32) * M)\}$

$= M * (116 + TE_L * 48)$
$+ M * (104 + 4 * M + TE_L * 20 + (TE_L * 32) * M)$

Traffic going out with the spread factor =
$\mathbf{(1/ T_{LH}) * \{\{M * 92\} + \{M * 16 * (1 + TE_L * D_L) + [M * TE_L * 60]\}\}}$
$\mathbf{+ (1/T_{OH}) * \{M * (48 + 16 * TE_L + 20 * TE_L * 32))\}}$

Traffic coming in with the spread factor =
$\mathbf{(1/ T_{LH}) * \{M * (116 + TE_L * 48)\}}$
$\mathbf{+ (1/T_{OH}) * \{M * (104 + 4 * M + TE_L * 20 + (TE_L * 32) * M)\}}$

10.3 During stable condition

Total traffic = Traffic due to link management protocols + traffic due to routing protocols + traffic due to signaling protocols

Traffic going out = {M* 20} + {16 * (1 + TE_{LC} * D_{LC}) + [TE_{LC} * 60]}
+ {(44 + 4 * M)} + {M * (28 + (D_{CL} * 32) * TE_{CL})}
+ {N * C * CR * (332 + N * 4)} + {N * C * CR * (412 + N * 4)}

= {M * 20} + {16 * (1 + TE_{LC} * D_{LC}) + [TE_{LC} * 60]}
+ {(44+ 4 * M)} + {M * (28 + (D_{CL} * 32) * TE_{CL})}
+ {N * C * CR * (744 + N * 8)}

Traffic coming in = {M * 20} + {32 + [TE_{LC} * 48]}
+ {M * (44 + 4 * M)} + {M * (24 + 20 (TE_{L} * 32))}
+ {N * C * CR * 444} + {N * C * CR * 240}

= {M * 20} + {32 + [TE_{LC} * 48]}
+ {M * (44 + 4 * M)} + {M * (24 + 20 (TE_{L} * 32))}
+ {N* C * CR * 844}

Probability of change = **Pc**

Final estimate on the traffic going out =
Pc * [(1/ T_{LT}) * {{M * 20} + {16 * (1 + TE_{LC} * D_{LC}) + [TE_{LC} * 60]}}
+ (1/T_{OU}) * {{(44+ 4 * M)} + {M * (28 + (D_{CL} * 32) * TE_{CL})}}
+ (1/T_{RR}) * {N * C * CR * (744 + N * 8)}]

Final estimate on the traffic coming in =
Pc * [(1/ T_{LT}) * {{M * 20} + {32 + [TE_{LC} * 48]}}
+ (1/T_{OU}) * {{M * (44 + 4 * M)} + {M * (24 + 20 (TE_{L} * 32))}}
+ (1/T_{RR}) * {N * C * CR * (744 + N * 8)}]

10.4 Due to failure conditions

10.4.1 Link failure

Total traffic = Traffic due to link management protocols + traffic due to routing protocols + traffic due to signaling protocols

Traffic going out =
{M* 20} + {20 + (TE_{FL} * (4 + 4 * (D_{L})))}
+ {(44 + 4 * M)} + {M * ((28 + (TE_{L} * 32) * TE_{FL}))}
+ {N * C * CR * (744 + N * 8)}

Traffic coming in = **{M * 20} + 20**
+ {M * (44 + 4 * M)} + {M * (24 + 20 * (TE_{FL} * 32))}
+ {N* C * CR * 844}

Probability of link failures = P_{L}

Final estimate on the traffic going out =
PL * [(1/ T_{LT}) * {{M* 20} + {20 + (TE_{FL} * (4 + 4 * (D_{L})))}}
+ (1/T_{OU}) * {{(44 + 4 * M)} + {M * ((28 + (TE_{L} * 32) * TE_{FL}))}}
+ (1/T_{RR}) * {{N * C * CR * (744 + N * 8)}}]

Final estimate on the traffic coming in =

$\mathbf{PL * [(1/ T_{LT}) * \{\{M * 20\} + 20\}}$
$\mathbf{+ (1/T_{OU}) * \{\{M * (44 + 4 * M)\} + \{M * (24 + 20 * (TE_{FL} * 32))\}\}}$
$\mathbf{+ (1/T_{RR}) * \{\{N* C * CR * 844\}\}}$

10.4.2 Node failure

Total traffic = Traffic due to link management protocols + traffic due to routing protocols + traffic due to signaling protocols

Traffic going out = $\mathbf{\{M * 20\}}$
$\mathbf{+ \{(44 + 4 * M)\} + \{M * M * ((28 + (TE_{L} * 32) * TE_{FL}))\}}$
$\mathbf{+ \{N * C * CR * (744 + N * 8)\}}$

Traffic coming in = $\mathbf{\{M * 20\}}$
$\mathbf{+ \{M * (44 + 4 * M)\} + \{M * M * (24 + 20 * (TE_{FL} * 32))\}}$
$\mathbf{+ \{N* C * CR * 844\}}$

Probability of link failures = P_N

Final estimate on the traffic going out =

$\mathbf{PL * [(1/ T_{LT}) * \{M* 20\}}$
$\mathbf{+ (1/T_{OU}) * \{\{(44 + 4 * M)\} + \{M * ((28 + (TE_{L} * 32) * TE_{FL}))\}\}}$
$\mathbf{+ (1/T_{RR}) * \{\{N * C * CR * (744 + N * 8)\}\}]}$

Final estimate on the traffic coming in =

$\mathbf{PL * [(1/ T_{LT}) * \{\{M * 20\} + 20\}}$
$\mathbf{+ (1/T_{OU}) * \{\{(44 + 4 * M)\} + \{M * M * ((28 + (TE_{L} * 32) * TE_{FL}))\}\}}$
$\mathbf{+ (1/T_{RR}) * \{N* C * CR * 844\}]}$