A Hierarchical Distributed Protocol for MPLS Path Creation

International Symposium on Computer and Communications (ISCC’02), IEEE

報告人：唐崇實

Outline

- Introduction
- Hierarchical Networks
- The Hierarchical Distributed Protocol (HDP)
- Evaluation
- Conclusions

Introduction

Question:

How do increases in size of the physical network affect the service creation performance under different loads?

Answer:

Detailed performance results showed the network edge routers to be the system bottleneck because they centrally deploy service control algorithms.

Solution

- Hierarchical Distributed Protocol (HDP)

Hierarchical Networks (1/2)

- Nodes are organized into different domains or Autonomous Systems (AS)
- Bandwidth Brokers (BB’s)
  - A BB maintains topological and state information about the nodes and links of an AS.
  - BB is a server node separate from physical nodes of the AS.
  - BB’s are cluster-based server farms that can grow in capacity.
- The *BB*’s for the level-*i AS*’s are grouped into virtual level-*(i+1) AS’s*

Hierarchical Networks (2/2)

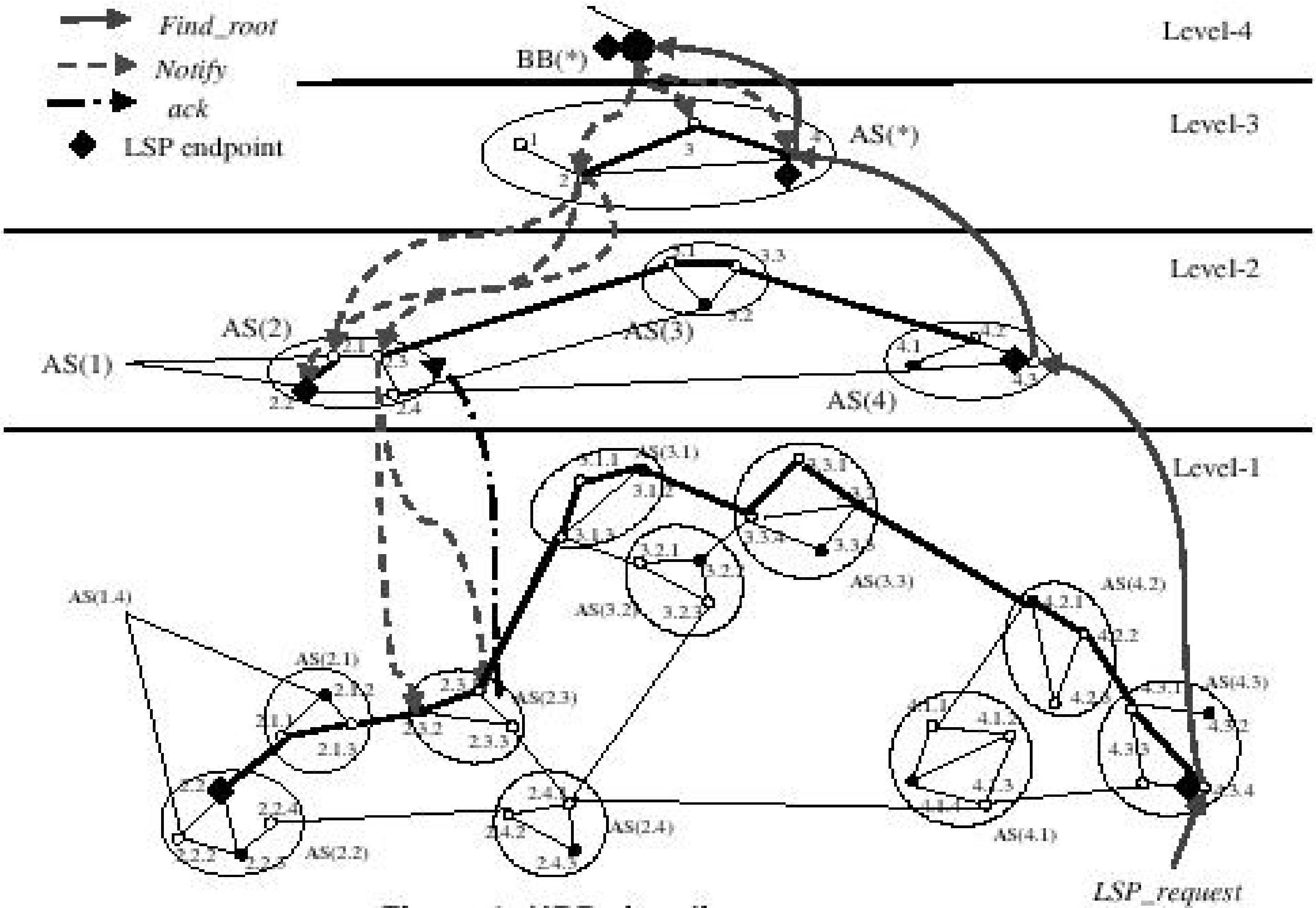


Figure 1: HDP signaling messages.

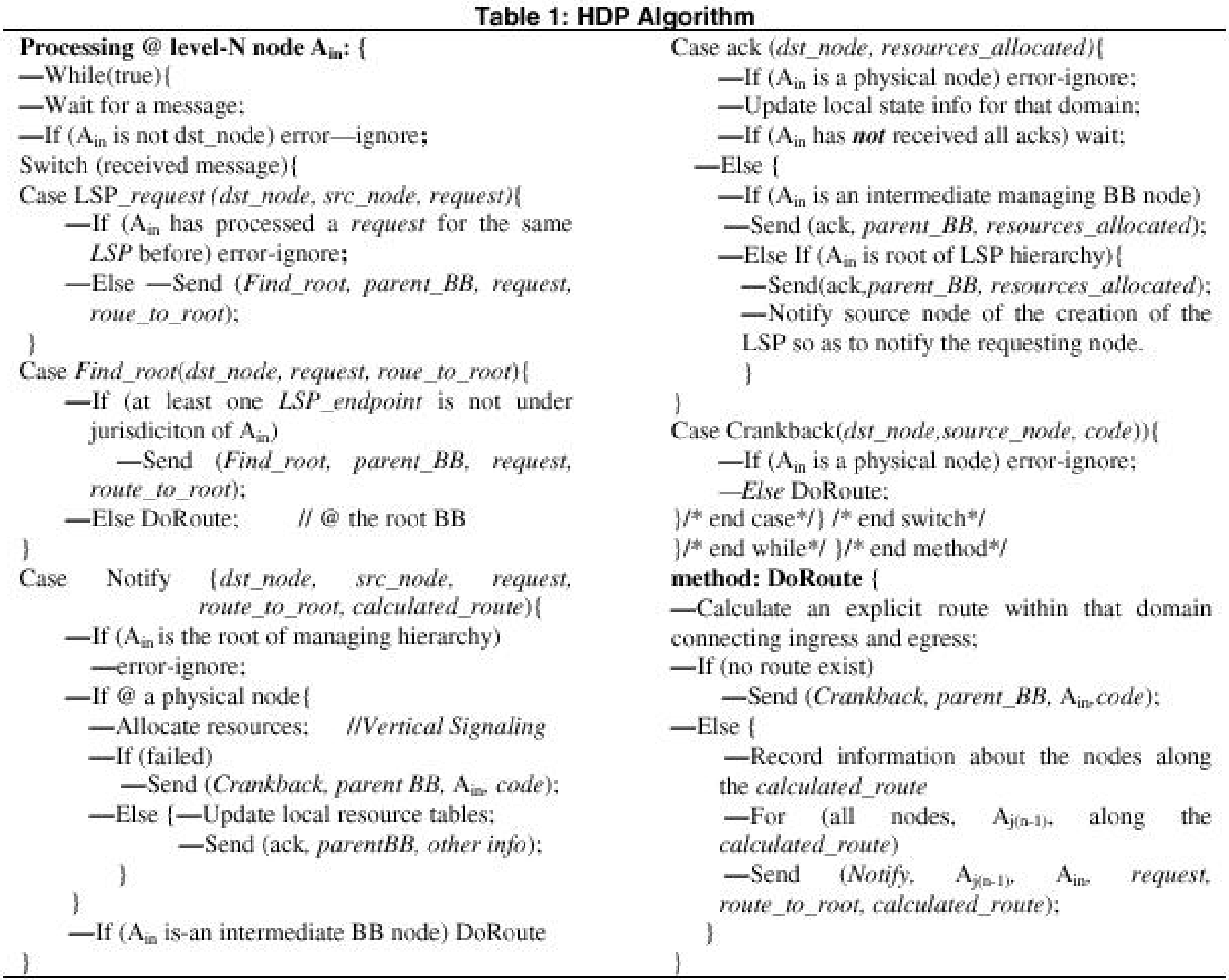
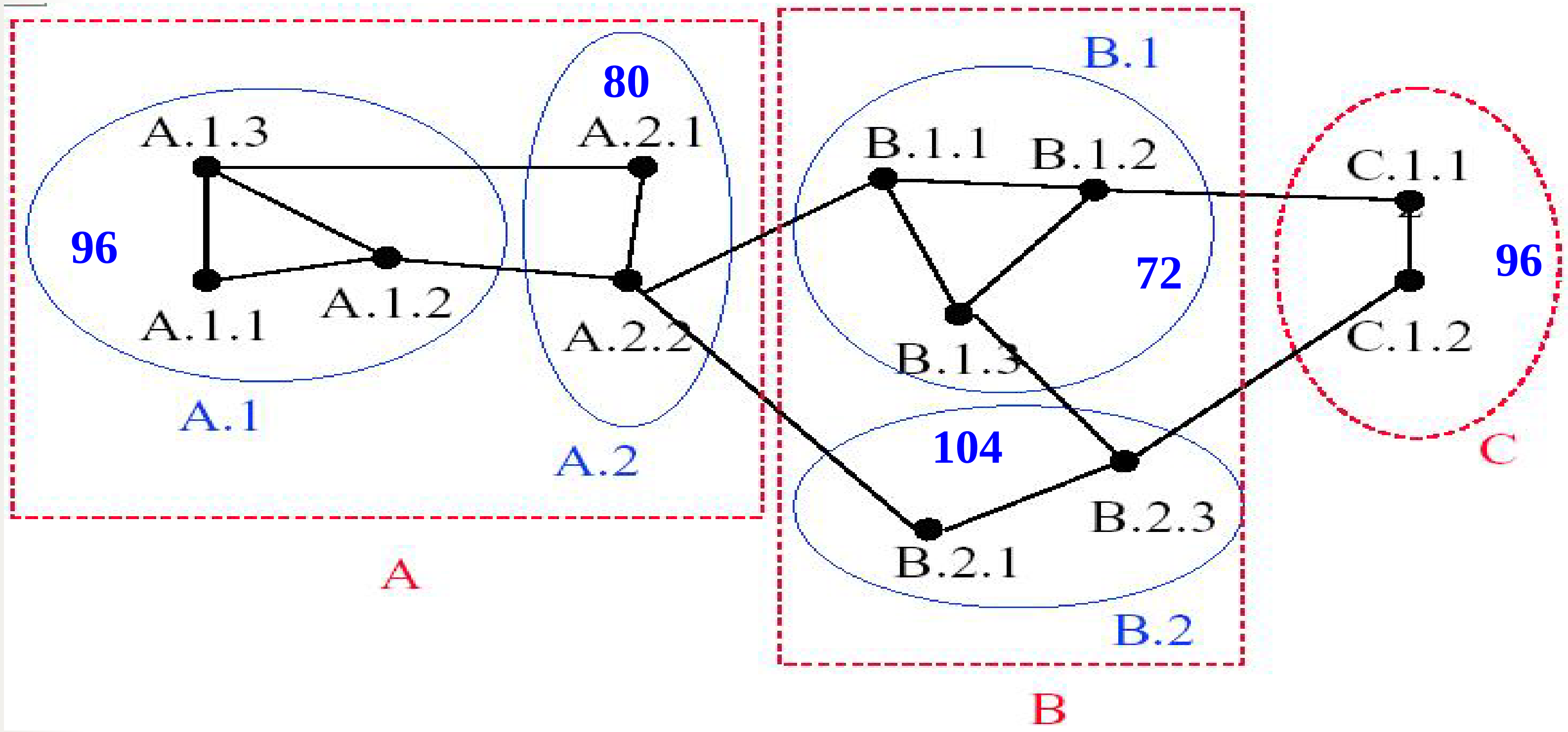


Table 1: HDP Algorithm

```
Processing @ level-N node $A_{in}$: {
—While(true){
—Wait for a message;
—If ($A_{in}$ is not dst_node) error—ignore;
Switch (received message){
Case LSP_request (dst_node, src_node, request){
      —If ($A_{in}$ has processed a request for the same
        LSP before) error-ignore;
      —Else —Send (Find_root, parent_BB, request,
        roue_to_root);
 }
Case Find_root(dst_node, request, roue_to_root){
      —If (at least one LSP_endpoint is not under
        jurisdiciton of $A_{in}$)
           —Send (Find_root, parent_BB, request,
        route_to_root);
      —Else DoRoute;        // @ the root BB
}
Case   Notify   {dst_node,   src_node,   request,
                route_to_root, calculated_route){
      —If ($A_{in}$ is the root of managing hierarchy)
        —error-ignore;
      —If @ a physical node{
        —Allocate resources;     //Vertical Signaling
        —If (failed)
           —Send (Crankback, parent BB, $A_{in}$, code);
        —Else {—Update local resource tables;
                  —Send (ack, parentBB, other info);
             }
       }
      —If ($A_{in}$ is-an intermediate BB node) DoRoute
}
Case ack (dst_node, resources_allocated){
      —If ($A_{in}$ is a physical node) error-ignore;
      —Update local state info for that domain;
      —If ($A_{in}$ has not received all acks) wait;
    —Else {
      —If ($A_{in}$ is an intermediate managing BB node)
       —Send (ack, parent_BB, resources_allocated);
      —Else If ($A_{in}$ is root of LSP hierarchy){
        —Send(ack,parent_BB, resources_allocated);
        —Notify source node of the creation of the
        LSP so as to notify the requesting node.
         }
}
Case Crankback(dst_node,source_node, code)){
      —If ($A_{in}$ is a physical node) error-ignore;
      —Else DoRoute;
}/* end case*/} /* end switch*/
}/* end while*/ }/* end method*/
method: DoRoute {
—Calculate an explicit route within that domain
connecting ingress and egress;
—If (no route exist)
      —Send (Crankback, parent_BB, $A_{in}$,code);
—Else {
      —Record information about the nodes along
      the calculated_route
      —For (all nodes, $A_{j(n-1)}$, along the
      calculated_route)
      —Send (Notify, $A_{j(n-1)}$, $A_{in}$, request,
      route_to_root, calculated_route);
    }
}
```

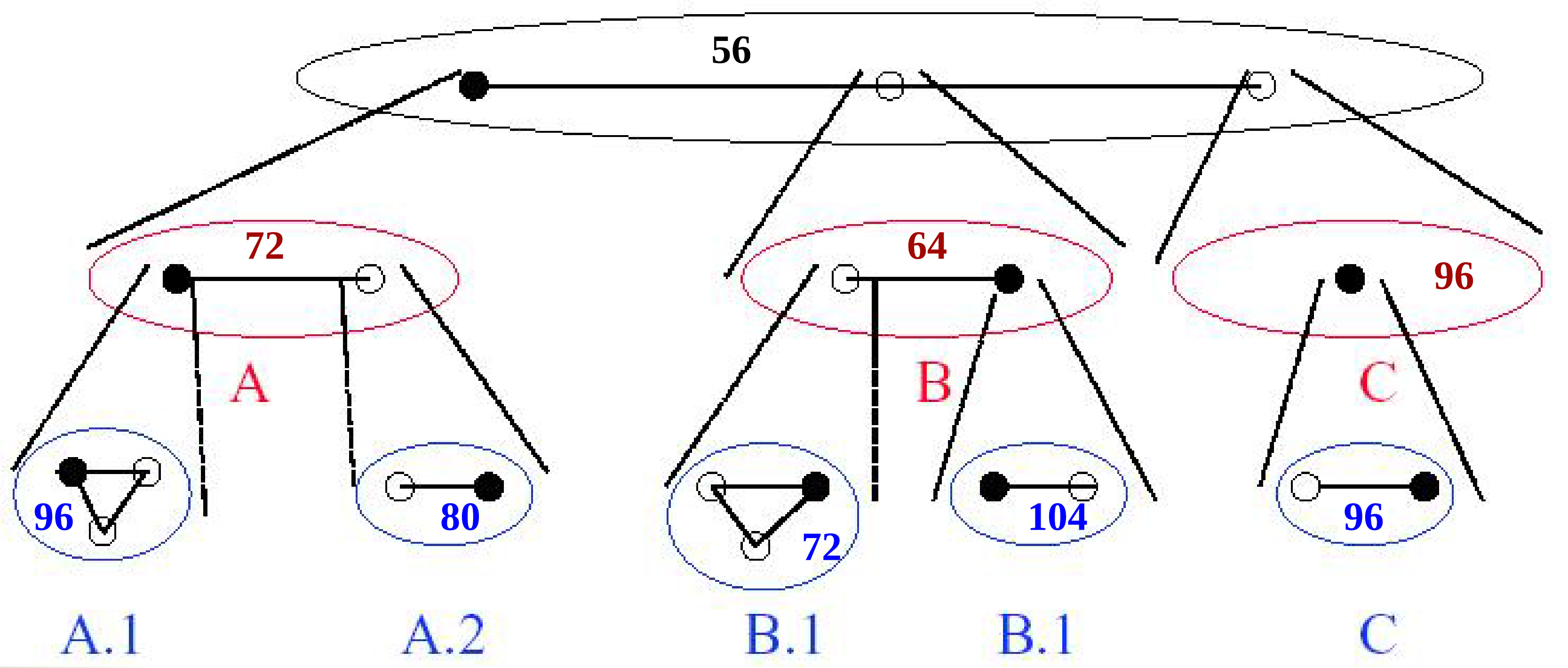
PNNI Hierarchical Routing (1/3)

- Uses 13-byte prefix to support 104 levels of hierarchy
- Nodes at a specific level are grouped into Peer Group (PG)



PNNI Hierarchical Routing (2/3)

- Hierarchical view



●：Peer Group Leader (PGL)

PNNI Hierarchical Routing (3/3)

- Main differences between PNNI and HDP
  - In PNNI, a physical node would do the routing calculations within the PG of its current level.
  - In HDP, BB's, rather than physical nodes, will maintain information about their hierarchy.
  - Route calculations in HDP are done in parallel as opposed to the in-series route calculation of PNNI.

Evaluation (1/4)

- Assume a hierarchy of
  - (***L***+1) uniform levels (including root BB)
  - ***m*** (network fan-out factor): average number of nodes in a physical/logical AS
  - ***d*** (path fan-out factor): average number of nodes in an AS that the MPLS path would traverse
  - ***E***: number of edges in a N-node domain is estimated

$$E = \frac{m^{L}}{2(R+1)}\left(1 - \frac{1}{m^{LR}+1}\right)$$ take R= -0.8

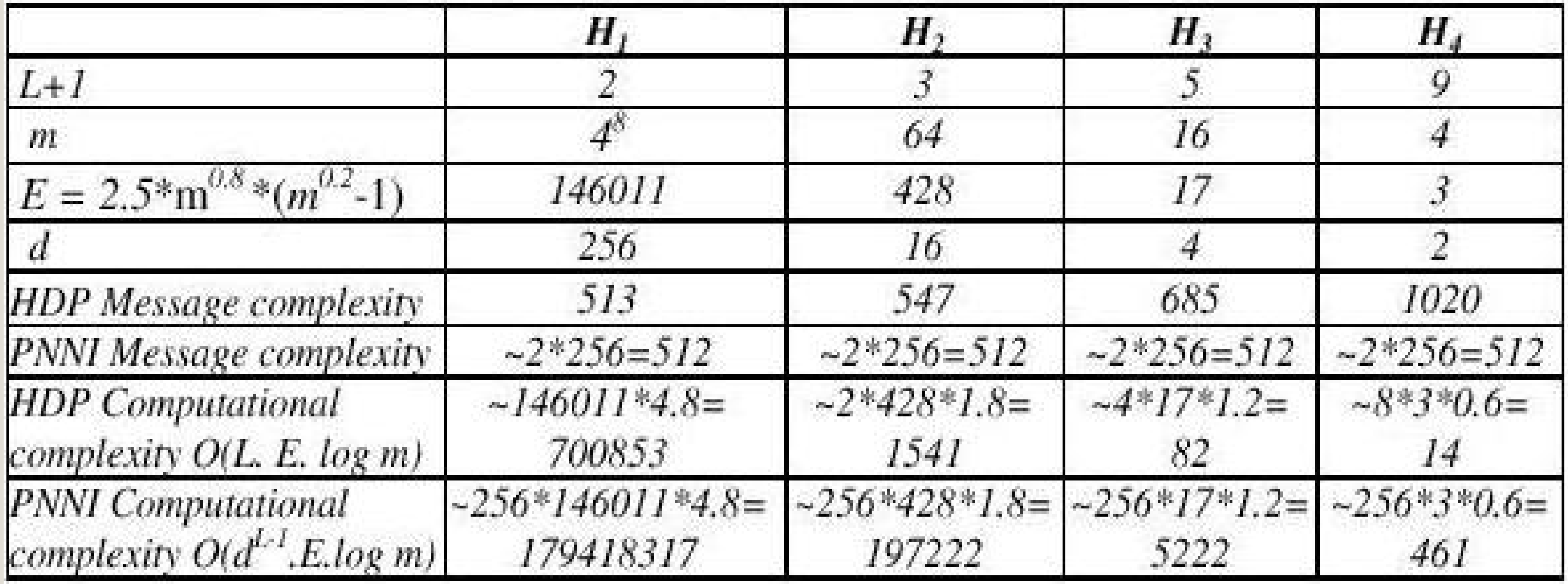
Evaluation (2/4)

Routing Algorithm	Message Complexity	Setup Time Complexity
HDP	$O\left(\sum_{i=1}^{L} d^{i}\right)$	$O(L \cdot E \cdot \log m)$
PNNI	$O(2 \cdot d^{L})$	$O(d^{L-1} \cdot E \cdot \log m)$
Flat Routing	$O(E + 2 \cdot d^{L})$	$O(2 \cdot d^{L})$

- HDP has a smaller routing computation time than PNNI at the expense of an increased number of messages
- Flat routing has a lighter computational load than HDP and PNNI, but comes at a higher message complexity.

Evaluation (3/4)

- H_1: all nodes arranged in a single physical system
- H_2: resembles the current architecture of Internet
- H_3 and H_4: one more level and two more levels than H2



	H_1	H_2	H_3	H_4
$L+1$	2	3	5	9
m	4^8	64	16	4
$E = 2.5*m^{0.8}*(m^{0.2}-1)$	146011	428	17	3
d	256	16	4	2
HDP Message complexity	513	547	685	1020
PNNI Message complexity	~2*256=512	~2*256=512	~2*256=512	~2*256=512
HDP Computational complexity $O(L.\ E.\ \log m)$	~146011*4.8= 700853	~2*428*1.8= 1541	~4*17*1.2= 82	~8*3*0.6= 14
PNNI Computational complexity $O(d^{L-1}.E.\log m)$	~256*146011*4.8= 179418317	~256*428*1.8= 197222	~256*17*1.2= 5222	~256*3*0.6= 461

Evaluation (4/4)

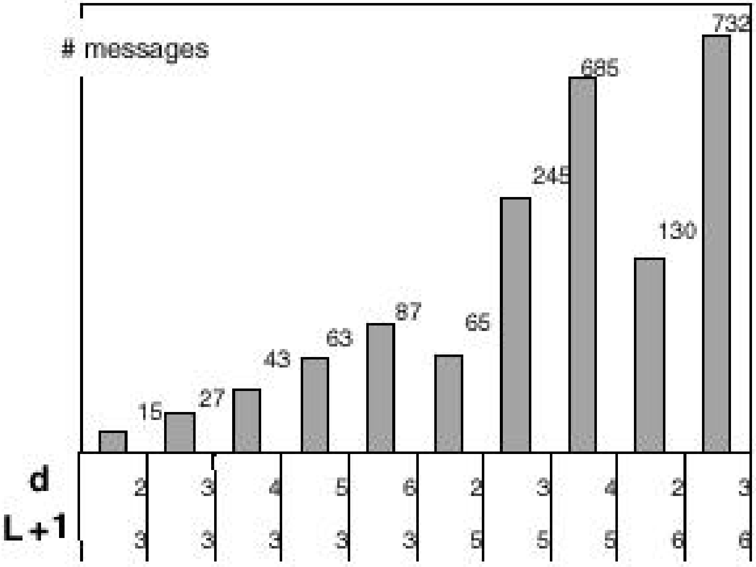


Figure 2: Number of setup messages

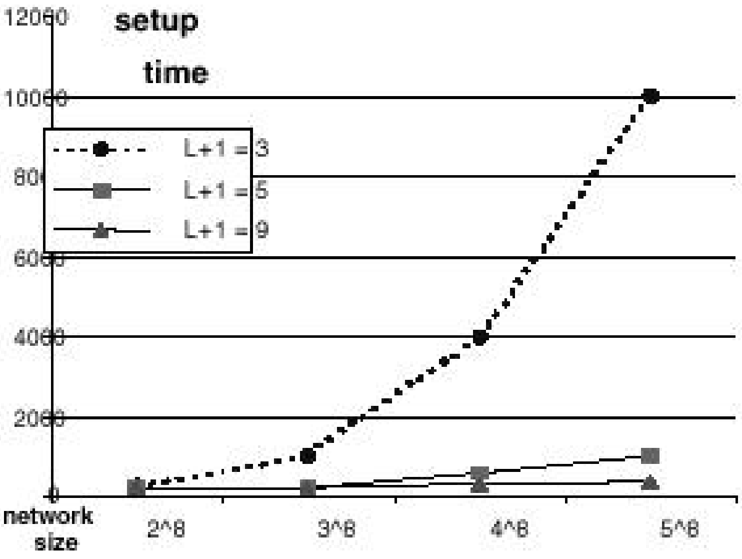
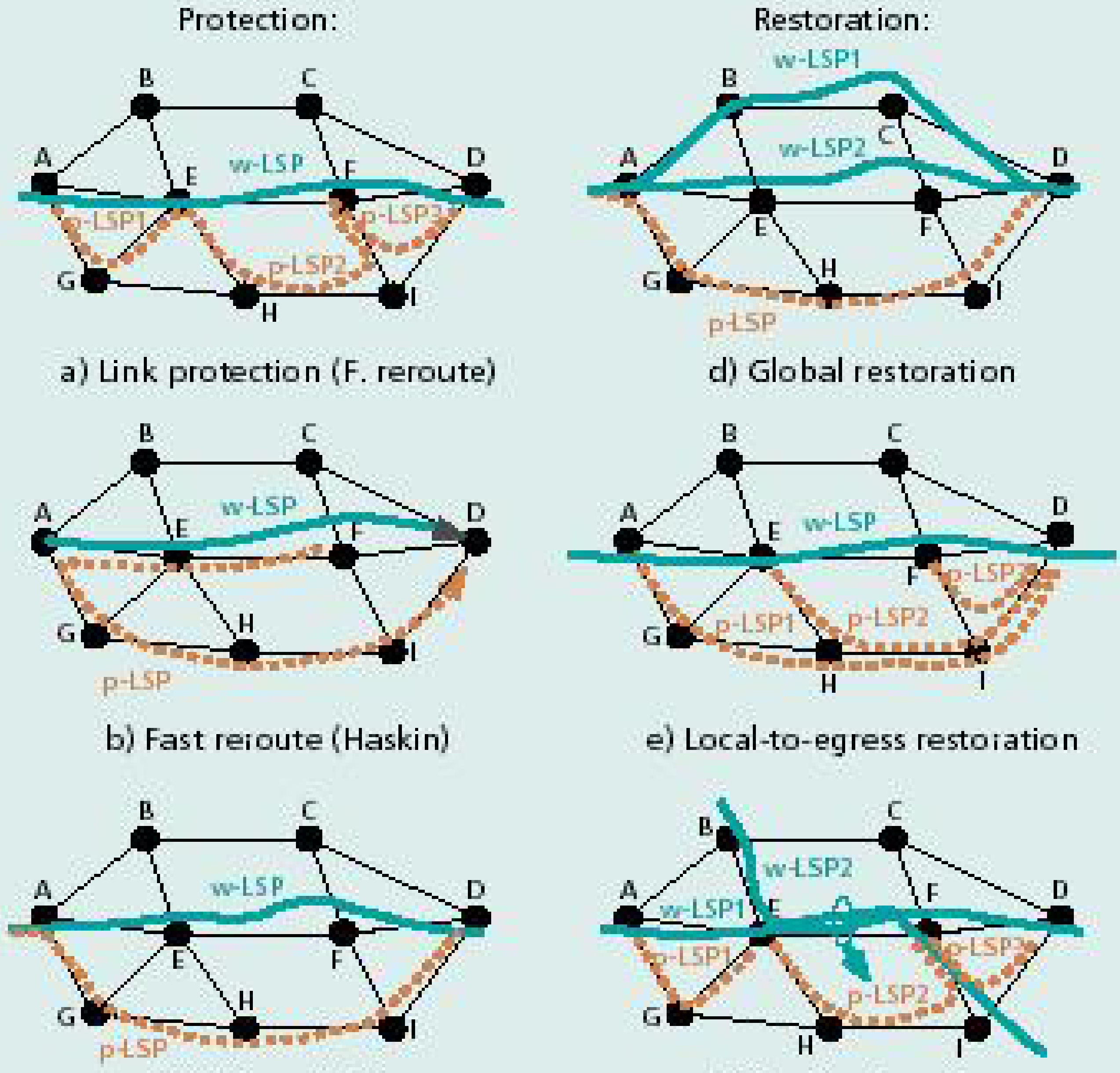


Figure 3: Setup time for different hierarchies

Conclusions

- A novel HDP for the creation of MPLS path is proposed.
- HDP reduces the setup time at the expense of an increased number of signaling messages.
- Discussion
    - Although BB's are separate from the physical nodes, it still needs to provide a "physical path" for signaling messages.
    - It is a question that if the hierarchy of more than two levels is really necessary.
    - Is is worthy to reduce the setup time at the expense of an increased number of signaling messages?
    - Other applications?



a) Link protection (F. reroute)

b) Fast reroute (Haskin)

c) Path protection

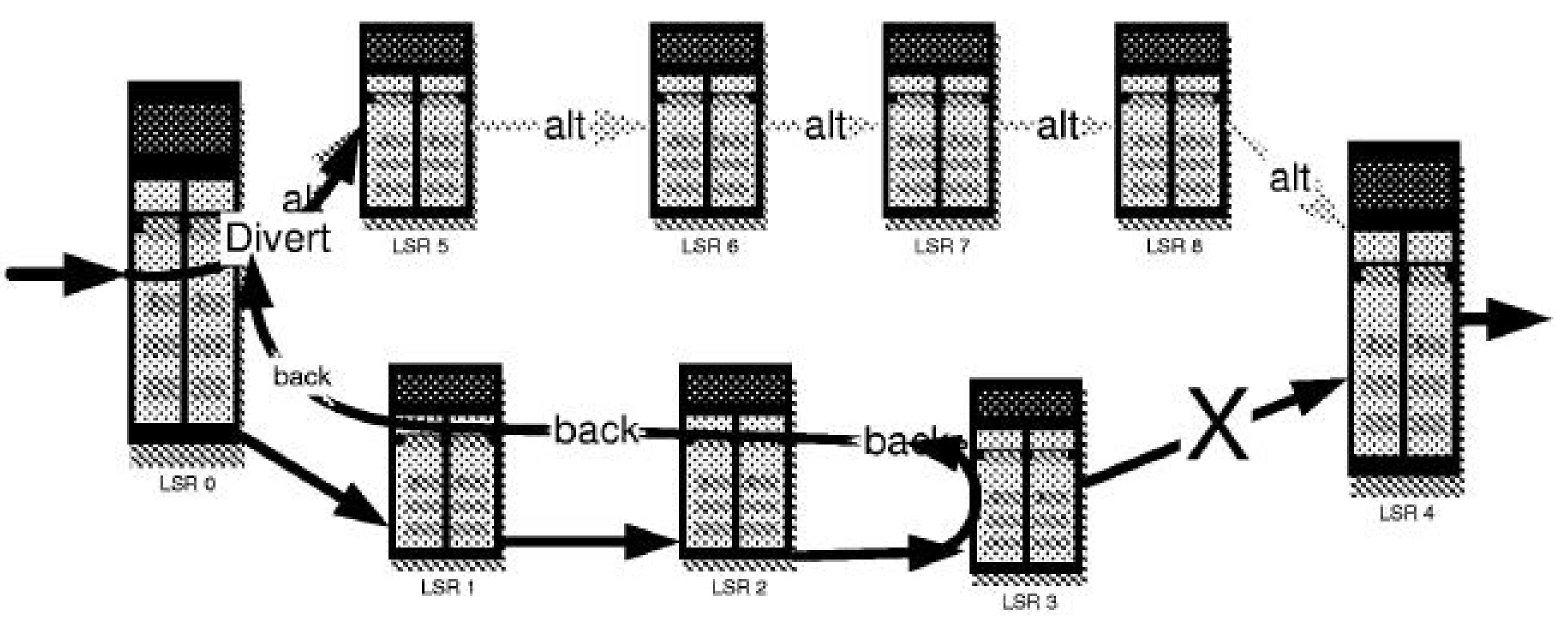
d) Global restoration

e) Local-to-egress restoration

f) Local restoration

Haskin Approach

- Important drawbacks
  - Long delay to send back the packets to ingress node
  - Data packet disordering



Another Improvement

- Fast rerouting mechanism for a protected LSP
  - When a fault is detected, packets are sent back via the backward LSP as in Haskin’s
  - Upstream nodes detect the packet on backward LSP then start storing incoming packets
  - The last packet forwarded before initiating storing is tagged
  - Preserve the ordering of packets and reduce delay
  - Needs large storage in each node