



---

Annual ADFSL Conference on Digital Forensics, Security and Law

2014  
Proceedings

---


May 29th, 1:40 PM

## Work in Progress: An Architecture for Network Path Reconstruction via Backtraced OSPF LSDB Synchronization

Raymond A. Hansen

*Dept. of Computer & Information Technology, Purdue University*

Follow this and additional works at: <https://commons.erau.edu/adfsl>

 Part of the [Aviation Safety and Security Commons](#), [Computer Law Commons](#), [Defense and Security Studies Commons](#), [Forensic Science and Technology Commons](#), [Information Security Commons](#), [National Security Law Commons](#), [OS and Networks Commons](#), [Other Computer Sciences Commons](#), and the [Social Control, Law, Crime, and Deviance Commons](#)

---

### Scholarly Commons Citation

Hansen, Raymond A., "Work in Progress: An Architecture for Network Path Reconstruction via Backtraced OSPF LSDB Synchronization" (2014). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 5. <https://commons.erau.edu/adfsl/2014/thursday/5>

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).

**EMBRY-RIDDLE**  
Aeronautical University™  
SCHOLARLY COMMONS

(c)ADFSL



# **WORK IN PROGRESS: AN ARCHITECTURE FOR NETWORK PATH RECONSTRUCTION VIA BACKTRACED OSPF LSDB SYNCHRONIZATION**

Raymond A. Hansen  
Dept. of Computer & Information Technology  
Purdue University

## **ABSTRACT**

There has been extensive work in crime scene reconstruction of physical locations, and much is known in terms of digital forensics of computing devices. However, the network has remained a nebulous combination of entities that are largely ignored during an investigation due to the transient nature of the data that flows through the networks. This paper introduces an architecture for network path reconstruction using the network layer reachability information shared via OSPF Link State Advertisements and the routines and functions of OSPF::rt\_sched() as applied to the construction of identical Link State Databases for all routers within an Area.

## **1. INTRODUCTION**

The traditional approaches in digital forensics deal with reconstructing events within one, or a small set of digital devices, such as computers, cellular phones, etc. These devices complicate the evidence gathering process due to not being designed or constructed with forensically-sound retrieval of data being a requirement. This difficulty is further increased when these devices are attached to a network and send or receive data with another device or devices [13, 1]. This transmitted data flows over networks in a transient manner, making forensic analysis of the flow difficult unless evidence is being gathered in real-time [6]. Frequently, there is a push to recover the system as quickly as possible and place it back into production. In doing so, forensic information may be lost, and additional investigation of the system is likely eliminated. As such, there is a balancing act between depth and time dedicated to the investigation and the speedy return to production.

A survey of discipline-centric publications and media highly suggest a continued increase in security incidents. As the number of incidents increase, the number of investigations will also increase proportionately. If current trends hold, relatively few perpetrators of cyber-incidents will be brought to justice. Yet, there is also a need to hold those perpetrators responsible for the actions and crimes [2], which has occurred with only marginal success.

As the bad guys recognize and smile at the slim likelihood of being held accountable for their online misdeeds, those who aren't guilty worry, with justification, that they could be wrongly accused, and those who are victims are largely without recourse.

It is clear that additional efforts are to be made in multiple areas of the judicial chain: investigators; processes, policies, and procedures; tools; legislators; and lawyers and judiciary.

While much effort in digital forensics has been given towards frameworks, policies, and end-user devices, the network infrastructure has not seen the same research efforts. Much of the research that has been accomplished in the network infrastructure arena has been centered around traceback of IP addresses [3, 5, 11, 12]. However, these approaches ignore the fact that networks are dynamic and changing, even over short periods of time, such as the flow of a nefarious payload. As with physical crime scene reconstruction, there is a need for a digital forensic investigator to be able to reconstruct and accurate cyber crime scene. The purpose of this research is to devise and develop an architecture

that maximizes the ability to accurately collect network path information while minimizing the evidence recovery and incident response costs [13, 1]. This research will focus on the Open Shortest Path First (OSPF) routing protocol, which is the most common open interior gateway routing protocol (IGP) for enterprise networks.

## **2. THE OSPF ROUTING PROTOCOL**

Network paths are constructed through the use of routing protocols, which share network layer reachability information. As routers share this information, a topology is agreed upon by all routers within the network and a valid next-hop router is chosen based on the destination IP address in each packet. The Open Shortest Path First (OSPF) routing protocol constructs its view of a network through exchanges of Database Descriptions (DD) and typed Link State Advertisements (LSA) [10]. Each OSPF router within an area processes the DD and LSA messages it receives and parses them into its link state database (LSDB). A router then builds its own network topology tree by parsing the LSDB with it- self as the root [8]. The resulting tree gives the router the necessary information to populate the forwarding information base (a.k.a. routing table) for routing decisions. In order to ensure up-to-date information across all routers within an OSPF area, synchronization of the LSDBs across all routers is requisite. This synchronization process results in each router within a specified area having an identical LSDB [9]. It is this identical LSDB that is to be examined in more detail.

### **2.1 LSDB Construction**

The following sections briefly describe the processes used to construct a routing table for routers within an OSPF area.

#### **2.1.1 Peer Adjacency**

Routers in an OSPF network are able to automatically learn of other routers through the use of Hello messages, which indicate the capability to communicate via OSPF and also provides the router's IP address for future communication. This two-way process is required to initiate the remainder of the process to establish a state where both routers are fully aware of the capabilities and reachability information of the other router, and is referred to as adjacency. Routers are not able to share Link State Updates until they have formed an adjacency; meaning this is a critical stage for routers to establish communication with other OSPF routes.

#### **2.1.2 Flooding**

Flooding is a critical, yet incomplete, component of the synchronization process between OSPF routers. Figure 1 shows a portion of this process, including the flow of data to/from other functions.

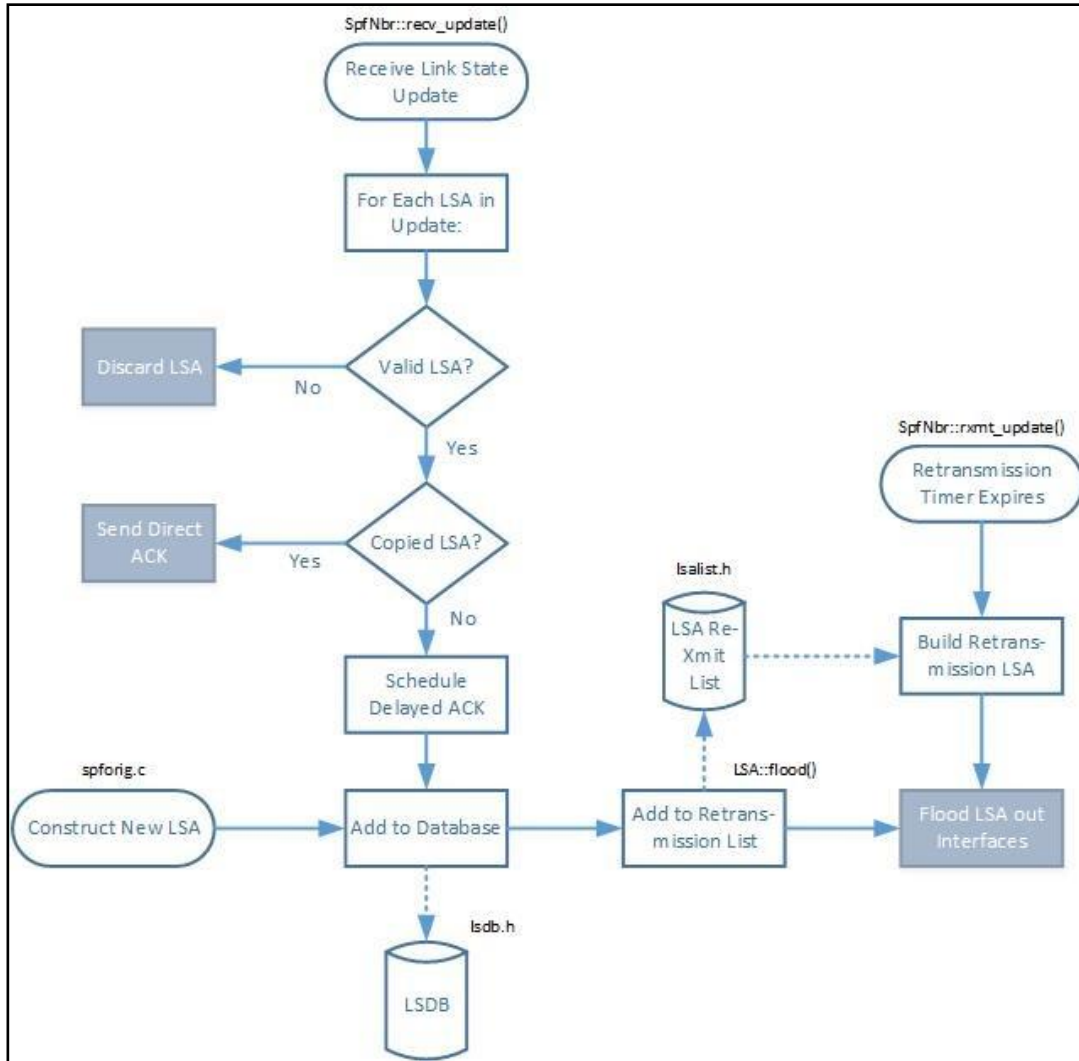


Figure 1 LSA Flooding Data Flow in Single Router

Utilizing this flow of data, each router within an OSPF area will share link state information of the network layer reachability information. Each router that receives these updates will utilize the parsing and processing structures to populate the LSDB.

### 2.2 Populating the Routing Table

An OSPF router will take the information learned via the adjacency construction, Database Descriptions, and Link State Advertisements to construct a shortest path first tree. In practice, it uses Dijkstra’s Algorithm to build a tree as a weighted, directed graph with itself as the root (single source vertex) of the tree. The shortest path to each destination (edge) is determined via the minimum of all possible summations of vertices [4]. Each of these shortest paths are then parsed from the tree and added to the routing table with the associated metric (cost) to reach that destination network.

### 2.3 Maintaining the Neighbor Adjacency

As long as no topological changes occur within the collection of networks, then no LSAs need to be sent. However, instead of long periods of silence occurring within the network where the assumption of no LSAs means no changes, Hello messages are sent by each interface (and subinterface) that is enabled within the OSPF process on a router. This periodic sending/receiving of Hello messages maintains the neighbor relationship to indicate that the peer is still active and operational, and no

changes have occurred. As long as this continues to occur within the allotted time frame, then the adjacency remains and the routers need not prune the local instance of the LSDB, prune their routing table, and subsequently send LSAs indicating the loss of a neighbor and its associated links and networks.

### **3. ARCHITECTURE FOR RT RECONSTRUCTION**

Based on each router within an area maintaining an identical LSDB, this study is examining the capabilities required to reconstruct the routing table of all routers within that area in order to reconstruct the physical network path that a packet or flow of packets would take from ingress to a destination device. Additionally, the reverse network path can be identified. The express purpose of such a tool is to provide attribution of a flow to specific devices as part of the investigation process in order to define the entire crime scene. Tan suggests four sources of incident data: victim machine, attacker machine, logged data (including intermediary systems), and physical security [13].

#### **3.1 Information Acquisition Mechanism**

This acquisition mechanism can, and likely will need, multiple incarnations. As there are multiple architectures to which OSPF can be implemented, each effecting the information shared between routers within an area, this acquisition of OSPF routing information will need to account for the architectural differences. The acquisition of OSPF NLRI in a broadcast network (i.e., Ethernet) or non-broadcast multiple access network can be simplified by mirroring the input to the Designated Router or Backup Designated Router. This could be accomplished with a packet capture tools, such as WireShark/TShark, TCPDump, or a variety of other tools.

In other network architectures, such as point-to-point links, this acquisition will require more effort. A network tap, or splice, could be used to eavesdrop on all traffic and subsequently filter out the non-routing protocol traffic. While not ideal, this could provide the desired information. However, as Ethernet continues to gain market share as a WAN and MAN interconnection, this may become a moot point. In the intervening time, a combination of NetFlow and SNMP may be sufficient.

#### **3.2 Information Storage Mechanism**

The storage component of this architecture should have the following criteria:

1. Sufficient storage capacity for archiving months of topology construction and maintenance information. A tool for backtracking intrusions via their process calls and dependencies required over 1 GB/day of storage to track all events on a single device [7]. This is a significant amount of storage required for a single host, and is untenable in an enterprise network which may host 100s of routers. Initial validation of storage requirements for this architecture suggest 50 MB/day is sufficient for a stable topology. An unstable network has not yet been tested where flapping events or significant AUDs may be occurring.
2. The storage system should not be an active network device. As an active network device, the storage system could become a target for attack which could undermine the reconstruction of its own attack and any others that occur within the effected timeframe. As such, it should be a passively- connected network device. It should only listen to the OSPF information that propagates through the network, and not participate in any other network operations.
3. The storage system should provide an interface to archive topology information to an offline state. This archival could be used for historical references and baselining in the event of an attack, or perceived attack, to identify typical trends and potential variability within the updating and maintenance processes.

### 3.3 Reconstruction Mechanism

This process is the critical component of this research. While a WireShark packet capture will provide a graphical interface to view individual packets and flows, it does not provide a mechanism for actual reconstruction of any events. NetFlow and SNMP traps and messages are incomplete in the information they provide about the state of the routing table as well as when and where topological information propagates. Based on a WireShark capture, however, replaying captured Hello messages into the 'SpfNbr::recv\_hello()' process and LSAs into the 'SpfNbr::recv\_update()' process, a reconstruction of the LSDB may be possible.

According to the OSPF Version 2 specification, a full Database Description exchange should happen between all OSPF peers every 30 minutes. Based on this exchange of information, the complete LSDB is purged, and reconstructed using only information from the most recent DD messages. The reconstruction mechanism, therefore, would only have to parse backwards to the 30-minute interval where an attack was initiated to begin reconstructing the full network path.

Validation of the state of the LSDB can be accomplished via MD5 hashing, which can be matched to a list of hashes that can be retrieved via SNMP from any router in the OSPF area [9]. This is one mechanism to verify database synchronization within OSPF as well. Once the LSDB has been constructed and validated to the proper timeframe, Dijkstra's Algorithm can be executed against the LSDB, and the resulting routing table entries can be identified. By setting the source router to any other router within the area when running Dijkstra's Algorithm against the LSDB, that router's routing table can be constructed. Repeating this for every router within the area would result in the ability to identify the path that any packet or flow would take through that OSPF area at a specified instant of time. This information can then also provide additional investigation opportunities to verify the integrity of the routers involved in the attack flow.

### 3.4 The Importance of Time

Timing is of critical importance in a synchronized routed network. Reconverging upon topological changes is a key component of the dynamic routing protocol's ability to quickly provide an up-to-date path to a destination network. Synchronized clocks between network devices allows sufficient time-stamping to be applied to update packets as to indicate an event occurred at an exact time, as opposed to some general time. Additionally, OSPF maintains internal clocks and counters to specify the initiation of other functions and process to maintain adjacencies and the LSDB [10]. As per the standard, Hello messages are to be sent every 10 seconds. Figure 2 shows a 24 hour sample of Hello messages from a single subinterface on a Cisco Catalyst WS-6504-E in a small-scale production network with very little user data flowing during the testing window.

While the standard calls for a 10 second interval, it is apparent that there is variability within a real-world implementation. The exactness of timing within this environment is quite important beyond just the network topology maintenance. Suppose an attacker wishes to modify an existing known path through a network as a means of "covering their tracks". If that perpetrator were able to inject spoofed topology information rapidly, conduct their attack, and reverse those topology changes rapidly enough, a simple analysis of the network may not show this alteration. By understanding the timing constraints of the OSPF protocol, both in theory and in practice, a better grasp of the timing requirements for this architecture can be understood. Given that the run time for a single OSPF router is  $O(E \log V)$ , the longest time to reconverge the network is of the same order  $O(E \log V)$  of all routers in the area.

Additional testing of timing of Database Description events and specific LSA information needs to be performed and evaluated in order to understand the potential time window for an unacknowledged attack that would provide details towards the failure rate of this architecture.

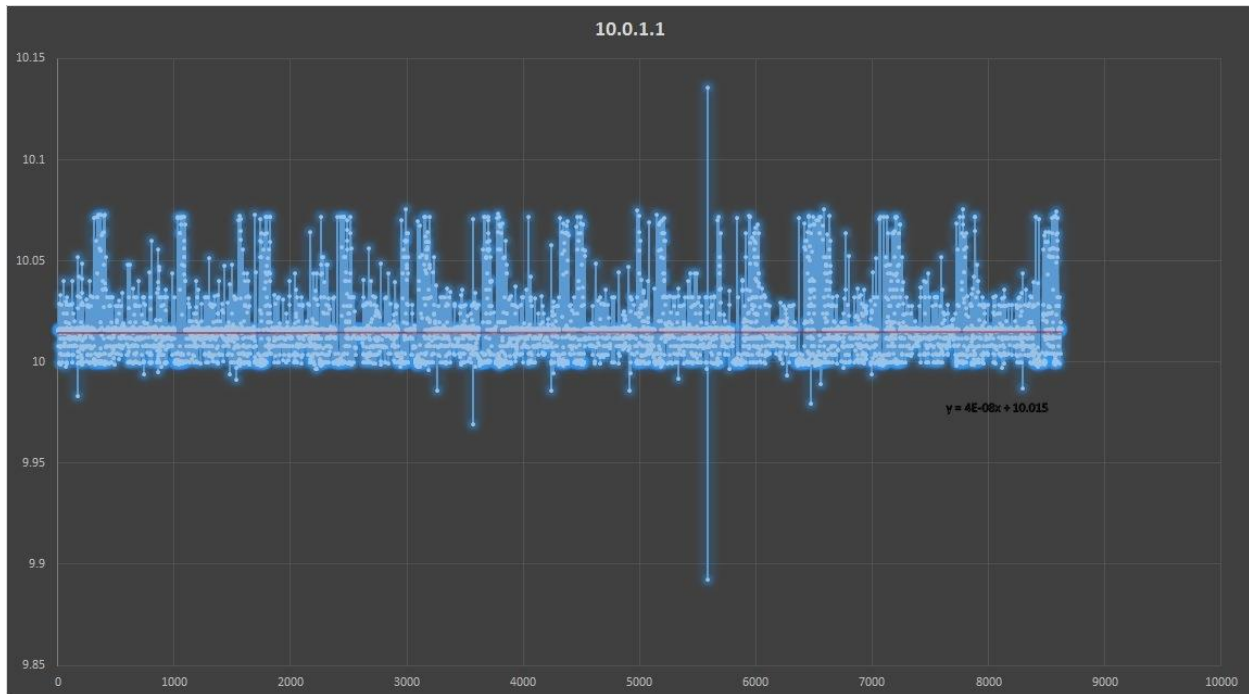


Figure 2 Hello Message Interval Timing in Tested Network

#### 4. CHALLENGES MOVING FORWARD

A short list of the known challenges are listed below. As additional challenges present themselves, this list will be appended. As those challenges are resolved, the resolutions will be worked into the body of this and subsequent papers.

1. OSPF Message Timing in the Real-World • OSPF Adjacency Establishment
  - Ongoing OSPF Hello Messages
  - OSPF Adjacency Establishment
  - Propagation Time of LSA Flooding in different topologies
2. Exceptions to RFC2328 and subsequent IETF documents
  - It appears that some implementations of OSPF implement functionality contrary to standard design (i.e., Cisco - LSDB refresh @ 30min)
3. Multicast OSPF
4. IPv6 and OSPFv3 operations and intricacies

#### REFERENCES

- Endicott-Popovsky, B., Frincke, D.A., & Taylor, C.A. (2007). A theoretical framework for organizational network readiness. *Journal of Computers*, 2(3).
- Endicott-Popovsky, B., & Horowitz, D.J., Unintended consequences: Digital evidence in our legal system. *IEEE Security and Privacy*, 10(2), 80-83.
- Dean, D., Franklin, M., & Stubblefield, A. An Algebraic Approach to IP Traceback.
- Dijkstra, E. (1959). *A note on two problems in connexion with graphs*.
- Doepfner, T., Klein, P., & Koyfman, A. (2000). Using Router Stamping to Identify the Source of IP Packets. 7<sup>th</sup> ACM Conference on Computer and Communications Security, Athens, Greece, Nov. 2000.

- Garfinkle, S. (2002). *Network forensics: Tapping the Internet*. O'Reilly Network. Retrieved on January 25, 2014 from <http://www.oreillynet.com/lpt/a/1733>
- King, S.T., & Chen, P.M. (2003). *Backtracking Intrusions*. 2003 SOSP, ACM. Bolton Landing, New York, NY, October, 19-22.
- Moy, J. (2001). *OSPF: Anatomy of an Internet Routing Protocol*. Upper Saddle River, NJ: Addison-Wesley Publishers, 20.
- Moy, J. (2001). *OSPF: Complete Implementation*. Upper Saddle River, NJ: Addison-Wesley Publishers.
- Moy, J. OSPF Version 2, Internet Engineering Task Force, 1998. Retrieved on February 20, 2014 from <http://tools.ietf.org/html/rfc2328>
- Savage, S., Wetherall, D., Karlin, A., & Anderson, T. (2000). *Textit* Practical network support for IP traceback, 2000 ACM SIGCOMM Conference.
- Song, D., & Perrig, A. (2000). *Advanced and authenticated marking schemes for IP traceback*, technical report UCB/CSD-00-1107, University of California, Berkeley, CA.
- Tan, J. (2001). *Forensic readiness*, Second Annual CanSecWest Conference.



