A novel approach for a Distributed Denial of Service Detection Engine

Christos Siaterlis
email:csiat@netmode.ntua.gr

Basil Maglaris
email:maglaris@netmode.ntua.gr

Panagiotis Roris
email:prori@netmode.ntua.gr

Network Management and Optimal Design (NETMODE) Lab
Department of Electrical and Computer Engineering
National Technical University of Athens

2002-2003

Abstract

In our present work we present some of the most popular data fusion algorithms that have inspired us to build an innovative Distributed Denial of Service (DDoS) Detection Engine. Our approach is based on the mathematical ground of Dempster-Shafer's Theory of Evidence (D-S). Using a set of simple heuristics to feed our D-S inference engine we attempt to detect flooding attacks in a set of experiments, that were conducted in real network topologies (in the National Technical University of Athens campus) using well known DDoS tools, like Stacheldraht.
The use of D-S model to express beliefs in some hypotheses, the ability to add the notion of uncertainty in the system and the quantitative measurement of the belief and plausibility of our detection results are some of the main advantages that this theory adds to an Intrusion detection framework and especially in comparison to a Bayesian estimator approach. Finally, we discuss several implementation and deployment issues in the context of security management and DDoS mitigating techniques.

1 Introduction

Although Distributed Denial of Service attacks have been in the focus of the internet research community during the last years, they still remain an open problem. The recent DDoS attack against the "AlJazeera" news network [16] or against the 13 root name servers [11] are only some of the attacks that have reached the mass media and highlight their usage in electronic warfare. As the Internet is going to evolve and become an inseparable part of our everyday life, regardless if it's in terms of education, information acquisition, communication, e-commerce or recreational activities, the ability of a single individual to deny our access to network resources is perilous.
Several DDoS prevention techniques (like Ingress [8] and RPF filtering [5]) have

been proposed in the literature and implemented by router vendors but they were not able to mitigate the problem. Most of the state of the art detection algorithms assume that the detection infrastructure is located near a saturated link in the vicinity of the victim, where the detection is "easy". In these cases local detection and response is ineffective as the available bandwidth has already been consumed in the upstream path. To couple with this problem "IP traceback" [18] and "IP Pushback" [12] aim to move the countermeasures near the sources of the attack. They assume though some sort of large scale cooperation. Another possible scenario, that has received much less attention, is occurring when the server under attack belongs to a customer hosted in a well connected ISP that performs DDoS detection on a link with low utilization. In this case the attack might stay undetected by the ISP. This case has great practical importance in the security management of ISP's, as its preferable to perform DDoS detection at few points of the over provisioned backbone and not necessarily on small, congested customer links. Additionally it is economically questionable to expect customers to pay for a dedicated DDoS detection service. In our current research effort we try to detect attempted DDoS attacks on high bandwidth links that can sustain the flooded packets without severe congestion. Throughout this article we will refer with the term DDoS attack to packet flooding attacks and not to logical DoS attacks that exploit certain OS or application vulnerabilities regardless if the attackers are trully distributed in the network topology.

Based on an exploration of the field of multi-sensor data fusion, we will present the use of Dempster-Shafer's "Theory of Evidence" as a framework for developing a DDoS detection engine. Our system's architecture consists of a set of distributed, autonomous but collaborating sensors which share their beliefs of the network's true state, ie whether its under an attack or not. We view the network as a system with stochastic behavior without assuming any underlying functional model. The attempt to infer the unknown system state is based on knowledge reported by sensors, that may have acquired their evidences based on totally different criteria. Possible sources of information could be signature-based IDS, DDoS detection programs, SNMP-based network monitoring systems, active measurements or network accounting systems like CISCO's Netflow [4]. Our detection principle tries to combine the reports of various network sensors and differs from many of the existing detection techniques that are focused on a single metric.

This paper is structured as follows: we will begin with a brief introduction of data fusion systems and commonly used algorithms (section 2). A more detailed presentation of the "Theory of Evidence" and its mathematical foundations will follow and particularly in contrast to the traditional data-fusion approach of "Bayesian Inference"(section 3). In section 4, we analyze the architecture of our detection engine prototype. Before we conclude, we present the preliminary results of our experiments in detecting DDoS attacks in real operational network in the NTUA campus network(section 5.1) and discuss the main advantages of our approach.

2 Introduction to Data Fusion

The process of collecting information from multiple and possibly heterogeneous sources and combine it in order to get a more descriptive, intuitive and meaningful result is done continuously by the human brain. The same process is needed for several systems in many different realms of science. The most common examples where such systems have been developed and widely used, are military systems for threat assessment and weather forecast systems. Generally, data fusion is a process performed on multisource data towards detection, association, correlation, estimation and combination of several data streams into one with a higher level of abstraction and greater meaningfulness.
The relevance of 'data fusion' with the main problem that current state of the art distributed intrusion detection systems face is obvious and has already been mentioned in [2]. Our innovation consists in the use of a typical data fusion algorithm to develop a DDoS detection engine that can combine the knowledge gathered by independent sensors and many different detection approaches in a powerful way and under a clear mathematical framework.

2.1 Data Fusion Architectures

There are many different architectures of data fusion systems and as these systems consist of several individual functional blocks we have many different combinations. Instead of providing a lengthy description of the different architectures we will just logically group and summarize the main functionalities of the different processing stages occurring in most data fusion systems (Figure 1)

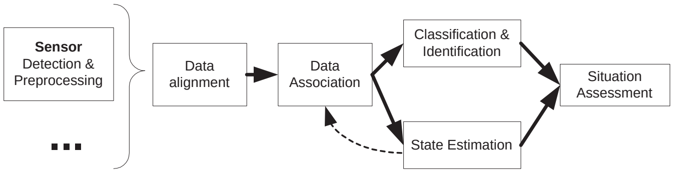

Figure 1: Typical data fusion system architecture

- Data Collection : Various sensors monitor, detect and report the environmental state
- Data Alignment & Association: Multisensor data may exhibit differences in time, space or measurement units that have to be aligned
- State Estimation: Based on a model of the system behavior and the knowledge acquired by the sensors a data fusion algorithm estimates the state of the system.
- Attribute classification & Identification: In this phase we identify the different targets and events that are being monitored.
- Situation Assessment: It's the highest level of information fusion where based on the states of the various targets and their identities (inputs from

the previous 2 processing phases) we determine the overall status of the system (”monitored world”).

From an algorithmic point of view many of these processing steps use the same mathematical methods. These ’Data fusion algorithms’ come from many different scientific areas and even a brief introduction to them would be very lengthy. Instead, we will just list in the next section a very brief description of some widely used methods by presenting the taxonomy that was proposed by Hall [10]. Our presentation and comments are especially targeted to the field of Intrusion Detection Systems.

2.2 Data Fusion Algorithms

2.2.1 Physical models

To begin, we can distinguish physical models which are based on the accurate modeling of the observed target and matching the measured data to the model. One of the most important representatives of these methods is the Kallman filter. It will provide the solution (state estimate) that minimizes the mean square error between the true state of the system and the estimate of state. It requires the knowledge of the state transition matrix and that the measurements are ”corrupted” by white zero mean noise with known covariance matrix. As the network behavior hasn’t been successfully modeled yet, the usability of such methods is doubtful.

2.2.2 Parametric Classification

In the second category, parametric classification , the algorithms make a direct mapping of parametric data to the classification space (for example the state of the system). We will briefly comment on some of them:

- Bayesian Inference Method (will be discussed in section 3.1)
- Dempster-Shafer Theory of Evidence (will be discussed in section 3.2)
- Adaptive Neural Nets are a very interesting and generic method that doesn’t assume a model for the observed system, but bases its output in the successful training of its nodes (neurons) using training data. The training could be supervised (by giving the correct classification on each sample data set) or unsupervised. The different kinds of neural networks differ in the number of nodes and layers as well as the processing function that is performed in each node (step, sigmoid). These methods have been used in the context of IDS’s but require training data that will be representative of the normal traffic data, which are very hard to gather or generate.
- Voting Methods: Probably the simplest and most intuitive method is voting. Each sensor’s data serves as a vote in a democracy where the fused declaration is the declaration of the majority. This method is extremely useful when a priori statistics are not known. Alternatives to the simple

voting is a weighted voting system or the use of intermediate decisions on a decision tree.

2.2.3 Cognitive algorithms

Members of the third category of cognitive based algorithms, try to simulate the human brain inference process. Some of them are:

- Expert systems have been successfully used in many applications. These systems consist of a knowledge base that represents the knowledge of some "field expert" usually in a production rule form. This knowledge can be facts, algorithms, heuristics etc. Expert systems have an inference algorithm and a separate control module which is the rule interpreter. One important advantage of these systems is that most of the times the system doesn't do an exhaustive search of the knowledge base, it presents only one possible inference and can also show the logical production rules used to show the path that has been followed. Most of the times the underlying theory is "First Order Logic", that has the drawback that cannot model the whole spectrum between belief and disbelief in a statement but uses a plain true or false approach instead.

- Fuzzy set theory is the fundamental theory that supports fuzzy logic, which is in turn used as an alternative to logical reasoning. In fuzzy logic, a statement is not just true or false but is rather a proposition with an associated value between 0 ,that represents a completely false proposition, and 1 - completely true (this is the membership value to the truthfulness set). The field of fuzzy logic is well defined and includes: combination rules and syllogisms. The "Theory of Evidence", that our research was based on, has in fact many common elements with Fuzzy logic.

We reviewed these candidate algorithms based on their applicability in the area of Denial of Service Attacks detection and we concluded that a promising method that needed further investigation was "Dempster-Shafer's Theory of Evidence". The main reasons that leaded us towards the D-S approach were that we don't have a good model for the normal network state and that methods that need training data, like neural networks, are excluded because representative data of a normal state (in terms of traffic or other attributes) is hard to obtain and time consuming to construct. Additionally there is a clear need to utilize information from multiple heterogeneous sources with different sensitivity, reliability and false alarm rates; for example anomaly detection heuristics that go beyond signature based methods. Expert knowledge, acquired by network administrators, should be able to be incorporated into the system but our detection decisions should not totally rely on it or require the development of complex sets of rules that will describe network behavior. We could argue that these algorithms could be very useful in terms of the individual sensors detection functionality but we prefer a more flexible modeling approach for data fusion.

3 The mathematical foundations of a D-S Detection engine

Our brief presentation of the "Theory of Evidence" will serve only as an introduction to the basic mathematical notations and concepts and will attempt to set the background for our application: the development of a DDoS detection engine. To complement our presentation and highlight the descriptive and modeling power of the theory, we will first present the Bayesian method for estimation that is a traditional modeling approach and has been used for DoS detection in [15]. To ease the reader we will note here, that in our application field, the observed system is the network and the measurements of the deployed sensors serve as evidence.

3.1 Bayesian inference

Let the possible states of a system be $\theta_1, \theta_2, ..., \theta_N \in \Theta$ and that these states are mutually exclusive and complete (exhaustive). The Probability $P(\theta_1)$ is an expression of the belief that the system is in state θ_1 in absence of any other knowledge. Once we obtain more knowledge in form of an evidence E then the appropriate expression to associate with the proposition θ_1 is the conditional probability $P(\theta_1|E)$ also called *"posterior probability"*. Based on the definition of conditional probabilities we have:

$$P(\theta_1|E) = \frac{P(\theta_1, E)}{P(E)} \tag{1}$$

Bayes theorem dictates :

$$P(\theta_1|E) = \frac{P(E|\theta_1)P(\theta_1)}{\sum_{i=1}^{N} P(E|\theta_i)P(\theta_i)} \tag{2}$$

If we have multiple evidences $E_1, ..., E_M$ and assume statistical independence between them, then we can combine them similarly. By using this formula we can combine evidence to infer the state of the observed system. We have to note that this method needs the knowledge of the "a priori" probability distribution of the states: $P(\theta_1), P(\theta_2), ..., P(\theta_N)$. In addition it does not provide any information about the quality of the result of our calculations, in terms of our trust in our evidence or the existence of conflicting evidence.

3.2 Theory of Evidence

Dempster-Shafer's Theory of Evidence can be considered an extension of Bayesian inference. There are many different ways to interpret the basic mathematical formulations of the theory that was introduced by Shafer in 1976 [19]. It can be viewed either from a probabilistic or an axiomatic point of view and all these approaches are concisely surveyed in [14]. Besides the different theoretical approaches and interpretations, all of them boil down to the same mathematical formulas regardless of the application. Theory of Evidence has been analyzed in the fields of statistical inference, diagnostics, risk analysis and decision analysis. Our approach and notations resemble mostly the field of "Diagnostics" [20].
Let us have a set of possible states of a system $\theta_1, \theta_2, ..., \theta_N \in \Theta$,which are

mutually exclusive and complete (exhaustive), which means that the system is certainly in one and only one of these states. The set Θ is often called *the frame of discernment.* We will call hypotheses H_i subsets of Θ, in other words elements of the powerset 2^{Θ}.

Our goal is to infer the true system state without having an explicit model of the system, just based on some observations $E_1, ..., E_M$. These evidences can be considered as hints (with some uncertainty) towards some system states. Based on one evidence E_j we assign a probability that it supports a certain hypothesis H_j. A *basic probability assignment (bpa)* is a mass function m which assigns beliefs in a hypothesis or as Shafer stated "the measure of belief that is committed exactly to H" [19].

$$m : 2^{\Theta} \rightarrow [0, 1] \tag{3}$$

This membership function m has to satisfy the following conditions:

$$m(\emptyset) = 0 \; and \; m(H) \geq 0, \forall H \subseteq \Theta$$

$$\sum_{H \subseteq \Theta} m(H) = 1 \tag{4}$$

Any hypothesis H such that $m(H) > 0$ is called a *focal set* and the set of all focal sets is the *core.*

At this point we have to underline the flexibility and advantages of this theory in contrast to the Bayesian approach, where we can only assign probabilities of single elements of Θ and not on elements of the powerset of the possible states. This theory gives us the opportunity to model uncertainty and the fact that some observations can distinguish between some system states, while they might not be able to provide any hints about others. For example, we might know that our evidence E_1 points to hypothesis $H_1 = \{\theta_1, \theta_2\}$ with a high probability but on the same time it provides no information (complete ignorance) whether the system is in θ_1 or θ_2.

Furthermore it is crucial that the "Theory of Evidence" calculates the probability that the evidence supports a hypothesis rather than calculating the probability of the hypothesis itself (like the traditional probabilistic approach).

We define a belief function Bel, describing the belief in a hypothesis H, as:

$$Bel(H) = \sum_{B \subseteq H} m(B) \tag{5}$$

This definition says intuitively that a portion of belief committed to a hypothesis B must also be committed to any other hypothesis that it implies, ie to any $H \supseteq B$.H A Belief function has the following properties:

$$Bel(\emptyset) = 0 \; and \; Bel(\Theta) = 1$$

The Plausibility of H is defined as

$$Pl(H) = \sum_{B \cap H \neq \emptyset} m(B) \tag{6}$$

and can be correlated to the doubt in the hypothesis H:

$$Pl(H) = 1 - Doubt(H) = 1 - Bel(H^c) \tag{7}$$

where H^c is the complement of H.
Intuitively, this relation means that the less doubt we have in a hypothesis H the more plausible it is. Generally we can characterize Bel(H) as a quantitative measure of all our supportive evidence and Pl(H) as a measure of how compatible our evidence is with H in terms of doubt. The true belief in H lies in the interval [Bel(H),Pl(H)]. Our degree of ignorance is represented by the difference Bel(H)-Pl(H).
The second important element of Dempster-Shafer theory is that it provides a rule to combine independent evidences E_1, E_2 into a single more informative hint $m_{12} = m_1 \oplus m_2$.

$$m_{12}(H) = \frac{\sum_{B \cap C = H} m_1(B) m_2(C)}{\sum_{B \cap C \neq \emptyset} m_1(B) m_2(C)} \tag{8}$$

Based on this formula we can combine our observations to infer the system state based on the values of belief and plausibility functions. In the same way we can incorporate new evidence and update our beliefs as we acquire new knowledge through observations.

Theory of Evidence makes the distinction between uncertainty and ignorance, so its a very useful way to reason with uncertainty based on incomplete and possibly contradictory information extracted from a stochastic environment. It does not need "a priori" knowledge or probability distributions on the possible system states like the Bayesian approach and as such it is mostly useful when we don't have a model of our system. In comparison with other inference processes, like first order logic which assumes complete and consistent knowledge and exhibits monotonicity [1] or probability theory which requires knowledge in terms of probability distributions and exhibits non-monotonicity [2] , Theory of Evidence has a definite advantage in a vague and unknown environment. The main disadvantage of Dempster-Shafer's theory is the assumption that the evidence are statistically independent from each other, since sources of information are often linked with some sort of dependence.

4 A D-S Detection engine prototype

As we have already mentioned, we used Dempster-Shafer Theory of Evidence to build a prototype for a novel DDoS detection engine that might aid network administrators to monitor their network more efficiently and with small set up cost. Network engineers know empirically, that there are often signs of flooding attacks but these signs are not always accurate or definite indications. They are mere hints and there is a clear need to intergrate them into a single higher level indication. In the same time these hints mostly stem from network monitoring or custom measurement systems and are very simple in nature. The reason behind this fact, is that "Measuring the Internet" is still a hard research

[1] if a fact is believed it cannot be refuted, so our knowledge always increases

[2] $P(A|E_1E_2)$ not determined by $P(A|E_1)$

problem and the tools have to cope with constantly increasing wire speeds and limited processing and storage resources. In our research, we have implemented a system that fuses the knowledge collected from the reports of various sensors, in order to infer the state of the monitored network. The system's architecture is depicted in figure 2.

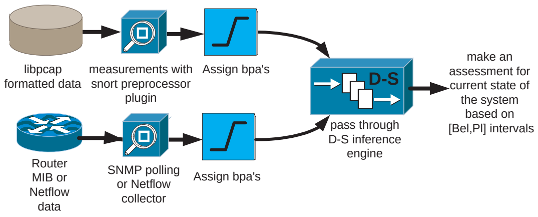
libpcap formatted data
measurements with snort preprocessor plugin
Assign bpa's
D-S
make an assessment for current state of the system based on [Bel,Pl] intervals
pass through D-S inference engine
Router MIB or Netflow data
SNMP polling or Netflow collector
Assign bpa's

Figure 2: System architecture

As in any data fusion system, our DDoS detection system's performance depends from the selection of its sensors. The most obvious source of knowledge acquisition is passive network monitoring, that can be performed in different ways ranging from passive sniffing with optical couplers to switch based port mirroring. Other sensor types might make active measurements like ping probes. Additional information can also be gathered from the Management Information Bases (MIB) that routers maintain or Netflow accounting systems that provide flow level information about network traffic. Generally speaking, some of the constraints in the selection of our sensors was that they had to be simple, efficient and easy to set up. The sensors that we have implemented so far can be classified in two different types:

- A preprocessor plugin for Snort (the popular open source IDS [3]) that produces traffic statistics based on captured packet data (libpcap format).The statistics kept were chosen to be simple so that its efficient and feasible to run at high wire-speeds. We collect data of the incoming and outgoing TCP,TCP SYN, TCP FIN, UDP,ICMP packet rates and their corresponding share of the link utilization.

- A SNMP data collector and analyzer that stores the acquired data in round robin databases (using the RRDtool [17]). Some examples of variables that we measure are bytes/sec, packets/sec, active flow number (based on Netflow [4]) and flow learn failures.

All sensors have their own 'intelligence' based on expert knowledge. In other words they have build-in functionality, so that after the right configuration and fine-tuning they are able to express beliefs about the network state. The main detection principles that were used in the configuration of the sensors are:

- Symmetry of TCP flows. Due to the nature of the TCP protocol we expect a loose symmetry on the incoming versus outgoing packet rates. This symmetry has already been used as a DDoS detection principle in D-WARD [13] and MULTOPS [9].

- ICMP and UDP attacks are mainly bandwidth consumption attacks and as these traffic types generally utilize small amounts of bandwidth, sudden changes in the transferred ICMP or UDP bytes/sec are good indications of attacks.

- The effect of spoofing in the number of active flows seen by a router. A flow is defined as a unique set of the following 5 characteristics <protocol, src IP, src port, dst IP dst port> and thus in the presence of a spoofed attack the number of active flows should rise suddenly. Besides this effect, it has been proven that the number of learning failures of a flow accounting algorithm was able to identify spoofed flooding attempts. The reason is that although the number of flows exhibits a high fluctuation in the face of normal traffic the flows are created and removed from the routers cache in a reasonable time interval. When a flooding attack occurs the amount of 'transports that are not completed' (for example with TCP FIN or RST) is high, so the entries are not removed gracefully but are filling up the cache and causing flow learning failures. More information about this mechanism is provided in [4].

Based on these principles we build algorithms that generate 'basic probability assignments' (bpa's) that match measured values to beliefs about the true system state.

In our first simplified implementation we define the following network states that are based on a flooding attack categorization of the DDoS tools that are currently in use: Θ ={NORMAL, TCP SYN ATTACK, UDP BWDTH ATTACK,ICMP BWDTH ATTACK}. SYN attacks are targeted towards specific services mainly aiming at OS resource consumption and the rest of the attacks base their success on the sheer volume of the generated traffic, thus bandwidth consumption. We have to note here, that this set of network states (Frame of Discernment in 'Theory of Evidence' terminology) must be the same throughout the system, from the sensors to the fusion node.

Lets illustrate the sensors functionality (transforming measurements to bpa's) with an example. Assume that a sensor measures a 'suspiciously high' value of the following metric $x = \frac{incoming\ UDP\ bytes/sec}{outgoing\ UDP\ bytes/sec}$. The sensor must state then its increased belief in the UDP-attack state. To be more specific, a sensor defines a m-value for 3 possible sets:

- It assigns a value that expresses its support for a set of states H that the sensor can recognize or is sensitive to, ie $m(H) \in [0, 1]$
- It assigns a value to the set $\neg H$, to express the refuting evidence of the hypothesis H, ie $m(\neg H) \in [0, 1]$.
- It assigns a value to the set Θ to express the ignorance of the sensor and the possibility that it might be erroneous (false report).$m(\Theta) \in [0, 1]$.

It follows from the equation (4) that $m(H) + m(\neg H) + m(\Theta) = 1$. A general guideline to help us define the individual m-values based on a measured value x is shown in figure 3. The intuition behind this 'rule of thumb' is that although going over and under certain thresholds can lead us towards a quite certain decision, in the interval between these low and high threshold values our beliefs should be treated with an increased uncertainty.

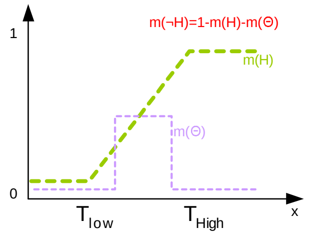

Figure 3: A generic guideline or 'rule of thumb' to define bpa's

The sensors periodically measure, calculate the corresponding bpa's and tranfer the collected knowledge to the fusion node based on a communication protocol that has as its main information a bpa or an m-function definition in the form:

$$< Timestamp >: m < sensorid > (< hypothesisset >) =< value >$$

This information can be easily expressed in XML and carried over an extension of the standard IDS communication protocol IDMEF [6].
We will include here a short example. Assume that one of our sensors measured at consecutive sampling periods:

Timestamp	incoming UDP bytes/sec	outgoing UDP bytes/sec
1053520284	180051	1327200
1053520285	3574611	1299368

If we assume that our heuristic is the incoming UDP bytes/sec vs outgoing UDP bytes/sec ratio and a network engineer has configured the sensor based on its network behavior (this metric is used here just as an example but nevertheless in our expirements it was shown to be stable in time), these measurements will be transformed into the following bpa :

Timestamp	BPA
1053520284	m1(UDP)=0.000000 m1(NORMAL)=1.000000
1053520285	m1(UDP)=0.792353 m1(NORMAL)=0.207647

The knowledge that is being collected by the various sensors will be then transfered in this form to the DS inference engine. The periodic sensor's reports update the current knowledge-base (belief pool) of the fusion node that runs with the sampling period of the fastest sensor (time allignment). The fusion node that implements Dempster's rule of combination was programmed in C and calculates the belief intervals for each member of the Frame of Discernment. The belief intervals that are visually represented with automatically generated graphs, quantify the validity of our results. The interpretation of the results is left to the human operator and from all the possible hypothesis sets, special attention must be paid in the sets: $NORMAL$,$\neg NORMAL$, and the individual attack states. 'Theory of Evidence' suggests the following interpretation of uncertainty intervals:

[Bel(H),Pl(H)]	Interpretation
$[0, 1]$	Total ignorance
$[x, x]$ where $x \in [0, 1]$	A definite probability of x
$[x, y]$ where $x \leq y \in [0, 1]$	Probability of x lies between x and y (uncertainty)
$[0, 0]$	Hypothesis is false
$[1, 1]$	Hypothesis is true

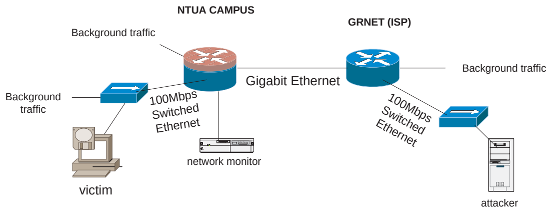

Figure 4: The topology of the experiment setup

A sample output (text format) of our D-S fusion engine if we combine the reports of our sensors, in the context of the same example, follows:

time	TCP-SYN	UDP	ICMP	NORMAL
1053520284	[0.000,0.000]	[0.000,0.000]	[0.000,0.000]	[1.000,1.000]
1053520285	[0.000,0.000]	[0.792,0.792]	[0.000,0.000]	[0.208,0.208]

5 Prototype Evaluation

5.1 Initial Expirement Results

To test and evaluate our D-S detection engine prototype we have performed a series of experiments in the university campus of the National Technical University of Athens (NTUA). In the scenario that we investigated the network that was monitored is single hosted and with an upstream link where traffic is aggregated but stays in low utilization levels. The experiments were conducted over several days during business hours and with background traffic generated from the more than 4000 hosts of the campus. Our university keeps a sustained rate of 200Mbps with its ISP with peaks higher than 300Mbps.
In our experiment scenario the victim is hosted inside the campus with a 10Mbps link whereas the attacker/s were outside the campus coming directly from our ISP (GRNET). The attacker was connected to a Fast Ethernet interface (100Mbps) to simulate the aggregation of traffic from several attacking hosts - zombies. Our attacker was running the well known DDoS tool 'Stacheldraht' [7] and was able to perform a series of flooding attacks with spoofed IP's [3] like SYN-floods, UDP and ICMP attacks. The network topology of our experiment setup is shown in figure 4.
The information sources that our sensors were build upon was a packet sniffer on the Gigabit Ethernet uplink of the university backbone and the MIB entries from the backbone router that were polled by our SNMP sensor. The router is a CISCO 6500 with Netflow enabled, so that we have access to flow level statistics.

In our initial experiments we used both sensor types that we described earlier and were able to detect all 3 types of attacks successfully. In our setup, measuring the false positive or false negative alarms is hard and would be highly variable with the sensor's threshold adjustment. Nevertheless our experience

[3] spoofing was performed by selecting source addresses from the attackers real subnet in order to bypass any e-gress or RPF filtering

indicates that we are able to maintain a low false positive alarm rate with reasonable effort from the part of the network administrator.

We will present here some representative expirement results that highlight that even if one sensor fails to detect an outgoing attack, combined knowledge gathered from other sensors that may work indicates clearly the increased belief on an attack state. In this expirement a UDP attack flooded the victim with a 34Mbps packet stream. As we see in figure 6 the active flows metric failed to identify the attack because the spoofing mechanism was choosing source and destination ports from a limited range. Nevertheless the incoming/outgoing UDP bytes/sec metric successfully identified an anomaly (figure 5).

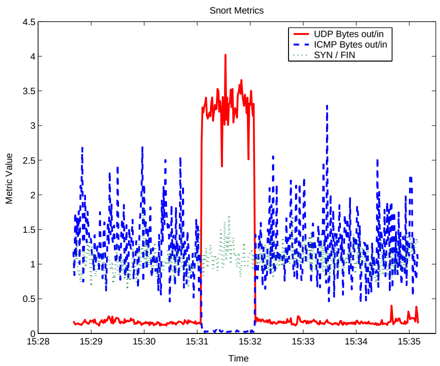

Figure 5: Real output from Snort-plugin that shows that in/out UDP and ICMP bytes/sec are good heuristics for UDP attacks.

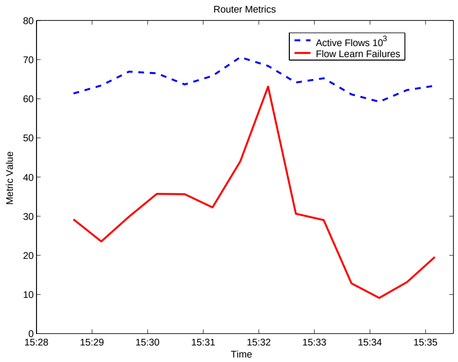

Figure 6: Real output from SNMP collector that shows that the 'Flow learn failure' heuristics partially detected the attack but the 'Number of active flows' metric failed.

These sensor measurements were then translated and expressed as bpa's that are shown in figures 7 and 8.

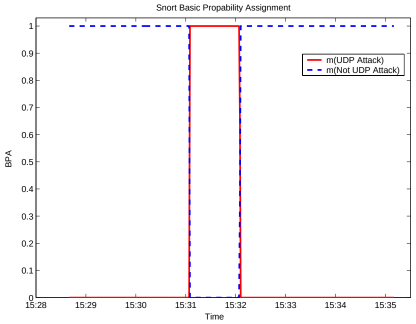

Figure 7: The basic probability assignment that corresponds to figure 5

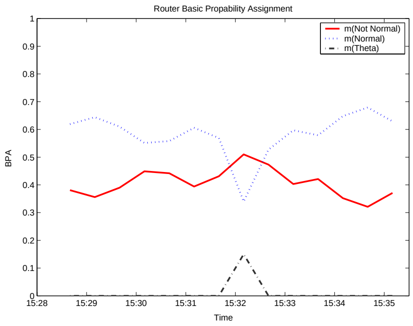

Figure 8: The basic probability assignment that corresponds to figure 6

The fusion node that combined the reported beliefs generated the higher level network state representation that we can see in figure 9. With this picture as input, the human operator could easily identify a potential UDP attack and start an in dept exploration that will begin with an evaluation of the invidual sensor reports.

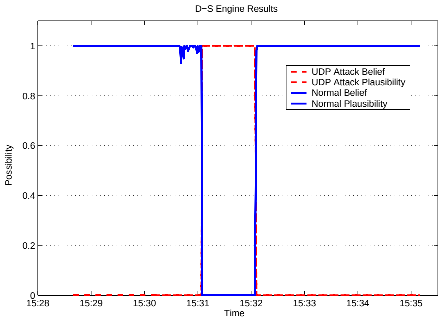

Figure 9: The output of the fusion node that combined the beliefs of figures 8 and 7

5.2 Discussion

The evaluation of the proposed algorithms and architecture can be the topic of a lengthy discussion. Implementing and incorporating these ideas into the security management infrastructure of an operational network may be a task of significant difficulty, but at the same time it may offer several advantages like those summarized below.

- We don't need to assume anything about the probability of the system being on a certain state, ie how often attacks occur. We just express beliefs that a monitored event supports a state.
- We can use the representation of ignorance to incorporate the false alarm rate or the predicted accuracy of a sensor to lower the false alarm rate of the fused reports.
- We can utilize many data sources, based on different detection algorithms. This way we can leverage on promising detection algorithms that have already been proposed.
- Based on the generic representation of knowledge in terms of basic probability assignments we can incorporate knowledge from traditional network monitoring infrastructure like service disruption alerts.
- The mathematical notation of membership function definition can be used to found the basis of a communication protocol for IDS collaboration as it allows fusion of data from diverse sensors.
- We can activate detection algorithms on demand, to refine our beliefs in the light of new evidence.

One common drawback of knowledge-based systems is that they can be as good as the sources from which they acquire their knowledge. Utilizing expert knowledge of network administrators might not be enough. One of the strengths of our approach is that we are able to incorporate any successful detection algorithm that has been proposed in the literature by simply adding a layer of abstraction in terms of basic bpa's, such a candidate algorithm is MULTOPS [9]. The most important fact is, that reports from other scientific fields like, traffic incident detection [1] or equipment condition monitoring [20] indicate that using Dempster-Shafer to combine results of different detection algorithms increases the detection rate. One other point that can be considered as a weakness of the proposed modeling framework is its inability to detect multiple simultaneous attacks, as we assume a mutually exclusive set of system states [4].

6 Conclusion

This paper proposes the use of Dempster-Shafer's Theory of Evidence as the underlying data fusion model for creating a DDoS detection engine. The modeling strength of the mathematical notation as well as the ability to take into account knowledge gathered from totally heterogeneous information sources are only some of the advantages. To demonstrate our idea we have developed a prototype that consists of a Snort preprocessor-plugin and a SNMP data collector that provide the necessary input that through heuristics feed the D-S inference engine. This simple but powerful data fusion paradigm can potentially include many of the proposed DDoS detection algorithms with their own strengths and weaknesses and could provide new solutions to the DDoS mitigation problem.

7 Acknowledgements

We would like to thank the personnel of the Network Operation Center of NTUA that provided us access to the necessary network infrastructure and supported our research efforts with guidelines, comments and helpful ideas.

References

[1] S.C. Byun and D.B. Choi and B.H. Ahn. Traffic incident detection using evidential reasoning based data fusion. In *Proceeding of the 6th World Congress on Intelligent Transport Systems*, Toronto, Canada, 1999.

[2] Tim Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, April 2000.

[3] Brian Caswell and Marty Roesch. Snort: The open source network intrusion detection system. http://www.snort.org.

[4] CISCO Netflow. `http://www.cisco.com/go/netflow`

http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.htm.

[5] CISCO. Unicast reverse path forwarding. `http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newf%t/121t/121t2/rpf_plus.htm`.

[4] Nevertheless we can expand the set Θ to resolve this problem

[6] David A. Curry and Hervé Debar. Intrusion detection message exchange format data model and extensible markup language (XML) document type definition. Internet Draft draft-ietf-idwg-requirements-10.txt, November 2002. Work-in-progress.

[7] David Dittrich. The stacheldraht distributed denial of service attack tool. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.

[8] Ferguson and Senie. RFC2827 network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing, May 2000.

[9] Thomer M. Gil and Massimiliano Poletto. MULTOPS: A data-structure for bandwidth attack detection. In USENIX, editor, *Proceedings of the Tenth USENIX Security Symposium, August 13–17, 2001, Washington, DC, USA*, Berkeley, CA, USA, 2001. USENIX.

[10] D. Hall. *Mathematical Techniques in Multisensor Data Fusion.* Artech House, Norwood, Massachussets, 1992.

[11] Internetnews.com. Massive ddos attack hit dns root servers. http://www.internetnews.com/dev-news/article.php/1486981.

[12] John Ioannidis and Steven M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. In *Proceedings of Network and Distributed System Security Symposium, Catamaran Resort Hotel San Diego, C alifornia 6-8 February 2002*, 1775 Wiehle Ave., Suite 102, Reston, VA 20190, February 2002. The Internet Society.

[13] G. Prier J. Mirkovic and P. Reiher. Attacking ddos at the source. In *Proceedings of ICNP 2002*, pages 312–321, Paris, France, November 2002.

[14] J. Kohlas and P.A. Monney. Theory of evidence - a survey of its mathematical foundations, applications and computational anaylsis. *ZOR- Mathematical Methods of Operations Research*, 39:35–68, 1994.

[15] Mohiuddin, Hershkop, Bhan, and Stolfo. Defending against a large scale dos attack. *Proceedings of the 2002 IEEE*, 2002.

[16] Nwfusion. Al-jazeera hobbled by ddos attack.
http://www.nwfusion.com/news/2003/0326aljahobbl.html.

[17] Tobi Oetiker. About rrdtool. http://people.ee.ethz.ch/ oetiker/webtools/rrdtool

[18] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, August 2000. An early version of the paper appeared as techreport UW-CSE-00-02-01 available at: `http://www.cs.washington.edu/homes/savage/traceback.html`.

[19] Glenn Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, 1976.

[20] K. Tomsovic and B. Baer. Fuzzy information approaches to equipment condition monitoring and diagnosis, 1998.