

---

# Distributed Inference to Detect a Network Attack

---

**John Mark Agosta**  
Intel Research  
Santa Clara, CA 95054  
john.m.agosta@intel.com

**Abraham Bachrach**  
EECS  
University of California, Berkeley  
abachrach@gmail.com

**Denver Dash**  
Intel Research  
Santa Clara, CA 95054  
denver.h.dash@intel.com

**Branislav Kveton**  
Intelligent Systems Program  
University of Pittsburgh  
bkveton@cs.pitt.edu

**Alex Newman**  
Computer Science Department  
Rensselaer Polytechnic Institute  
newmaa2@cs.rpi.edu

**Eve Schooler**  
Intel Research  
Santa Clara, CA 05054  
eve.m.schooler@intel.com

## 1 Distributed Network Attack Detection

We consider the problem of identifying a network attack that provides evidence of its existence through the presence of weak, distributed information. One example of weak distributed information, discussed by Jung *et al.* [2004] and Weaver *et al.* [2004], occurs during a scanning worm’s reconnaissance phase, where several hosts in a network may be able to pick up a signal which alone does not look alarming, but in concert with other detections throughout the network might be reason for alarm. Another example occurs in the early stages of a “stealth” epidemic infection of a network where several (infected) nodes throughout the network might start to behave in a slightly abnormal fashion. Although, a few anomalies are not worthy of attention, several taken together might be cause for concern. The challenge in both of these examples is that evidence of attack is weak since it is distributed, possibly over both space and time. Our goal is to design a detector that can aggregate weak, general evidence of malicious behavior rather than identifying specific known attack signatures and so has a better chance of succeeding over “day-zero” attacks.

The general architecture of our system consists of many host-based, weak local detectors. Given that a host is under attack, the local detector will fire with a probability equal to its *true positive rate*; given that the host is not under attack, it will fire with a probability equal to its *false positive rate*. A “weak” local detector is one that is capable of detecting a wide-range of different attacks, but does so at a relatively low true positive rate and/or a relatively high false positive rate. A weak detector by itself is not useful for detection because high false positive rates would lead to an overly disruptive system.

In addition to possessing local detectors, each host is equipped with an *aggregate detector* that takes as input the states of local detections throughout the network and must decide whether or not the network as a whole is under some attack. Each local detector participates in the corroboration algorithm, sending messages summarizing their local “belief states” to other hosts. The aggregate detectors at the recipient hosts, taking advantage of the combination of their local evidence with received beliefs, must decide whether or not the network is under attack. If so, we assume each node, and the system in general, has a mechanism to rapidly take action against the detected threat. In addition, the aggregate detectors must possess a mechanism to “expire” old information about the network; at the time of a network attack the aggregate detectors typically will contain mixed evidence from the time before and after the attack started, and the aggregate detectors must handle this case gracefully.

In this paper, we explore aggregate models that can be described as dynamic Bayesian networks (DBNs) [Dean and Kanazawa, 1989]. These models contain stages  $k = 1 \dots N$  where each stage models the dependency between global network state and observations generated by host local detectors. Transitions between stages model the time evolution of the state. The connectivity of the DBN network refers to the dependence between the firing of local detectors. In other words, if the probability of host  $A$  firing given that an attack has taken place is independent of whether or not node  $B$  has fired, then there should not be a directed path between  $A$  and  $B$ . Thus, the DBN structure depends strongly on the behavior of the attack (many worms select targets at random whereas others use a hit-list), and is

highly independent of the underlying topology of the computer network.

We explore several DBN models of varying complexity. The simplest are *change-point* models, where the model assumes that prior to the attack all nodes fire at their false-positive rate, and after the attack all host-detectors fire at their true positive rate. This is clearly a gross assumption for the case of a worm attack where either a worm is scanning or infecting selective hosts in the network. We have also begun to explore models that consider the dynamics of the fraction of affected nodes growing in the network as more and more local hosts become infected. The models we have developed vary in their demands on computation, network bandwidth and latency, and their accuracy. All the models that we have tested appear to out-perform the Cumulative Sum (CuSum) classifier, a widely used on-line statistical detection method.

We consider at least two criteria used by the aggregate detectors to draw a conclusion of attack or no attack. One strategy is to treat the problem within a standard decision-theoretical framework known as a *partially-observable Markov decision process (POMDP)* [Kaelbling *et al.*, 1998]. Under this framework, at each time step, the aggregate intrusion detector can take three actions  $A = \{a_{\text{continue}}, a_{\text{alarm}}, a_{\text{clear}}\}$ , where the latter two are terminating. At each step in time, an estimated cost of taking each action is made based on the primitive cost of a false positive  $C_{\text{fp}}$ , the cost of a false negative  $C_{\text{fn}}$ , and the cost of an additional sample  $C_c$ . Roughly speaking, whatever action has lowest expected cost at each step will be the action performed. The second strategy we use amounts to treating the aggregate detector as a classifier using the likelihood ratio of an attack versus non-attack as a trigger. In this framework, standard cross-validation and ROC analyses from machine learning can be applied to select a threshold that is optimal under some cost function, averaged over all time steps.

Since each host possesses an aggregate detector with a possibly different set of evidence, our system is essentially an ensemble of aggregate classifiers. We thus require a policy for reconciling the decisions of each aggregate detector. A simple ensemble policy, and the one we have implemented, is to accept the first alarm generated by any sequential detector as the global alarm. We are exploring the possibility of using the diversity of aggregate detectors to build more sophisticated ensembles such as boosting [Schapire, 2001], which have been shown to generalize better in many empirical settings.

In order to make this system scalable, particularly in a large enterprise network with  $O(100K)$  machines, careful attention must be paid to the messaging protocol that is used to share beliefs between hosts. We are exploring several promising messaging protocols that trade off speed of belief dissemination with network bandwidth consumption: beaconing, epidemic gossip algorithms, multicast, biased channels, and hierarchical messaging. Unlike many other distributed group algorithms, the focus here is not on reaching consensus among all nodes, but rather on finding the minimal number of corroborators necessary to reach a conclusion. Thus when anomalous events occur sparsely in the population, the protocol must endeavor to share those interesting events widely, possibly more quickly than other events. In addition, in the context of network intrusion there are several interesting design considerations: (1) realistically, upper bounds exist on the amount of network bandwidth that such a protocol can or should consume, and (2) the speed of the corroboration algorithm must be as fast or faster than that of the worm, if it is to contain an attack in a timely fashion, yet it still faces the challenge to operate within the bandwidth constraints. At the same time, the messaging must be robust not only to the vagaries of normal networks (delay, packet loss, dynamics of node connectivity), but also to security issues (the gaming of algorithm thresholds externally, the possibility of rogue peers within the algorithm).

## 2 Comparisons

A comparable distributed sequential detector can be constructed from a conventional change-point detector. Change-point detection in its most general form identifies a discontinuity in a noisy signal. Real-time change-point detection appears in Statistical Process Control (SPC), where the approach typically is to compare a running statistic of the sequence to a threshold. Of these methods Cumulative sum (Cusum) and Exponentially weighted moving average (EWMA) methods are the best known [Hawkins and Qiu, 2003]. The Cusum detector maintains a running sum of detections that restarts each time the sum crosses zero. In any single instance this signal tends to remain small until the change-point occurs, at which point it trends toward infinity.

We compare our DBN models against the Cusum method on synthetically generated data. Our data assumed a discrete change-point event in which all nodes fire at their false positive rate (0.15) prior to the event and all nodes fire at their true positive rate (approximately 0.53) after the event.

Evaluating the detection performance of an aggregate detector involves a tradeoff between time to

detection and the false positive rate. The more data collected by the aggregate detector, the more reliable the detections will be, at the cost of the time required to collect the data. We capture this tradeoff by generating the AMOC curve [Fawcett and Provost, 1999] of the detectors, which shows time-to-detection versus false positive rate as the sensitivity of the classifier is swept out from 0 to 1. Figure 1 shows the comparative time-to-detection versus false positive rate for an ensemble of Cusum detectors, for a DBN ensemble and for a POMDP ensemble.

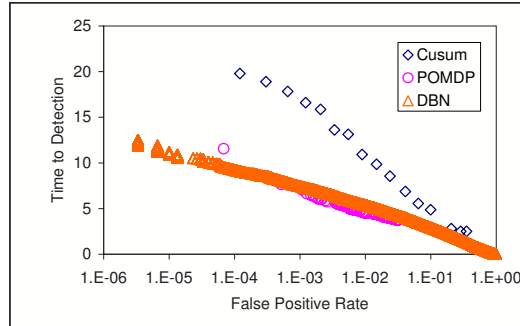


Figure 1: For the lowest false positive rates, the DBN models detect roughly twice as fast as the Cusum model. All assume  $p_{tp} = 0.53$ ,  $p_{fp} = 0.15$ . The POMDP detector curve is parameterized by  $C_{fp}/C_{fn}$ .  $C_c$  was chosen to give best performance.

The results that we show are limited in that they assume a change-point event. For some network attacks such as an aggressive scanning attack, this assumption may be approximately correct. However for others, such as an epidemic infection of the network, this will be a gross oversimplification. We have experimented with altering our DBN models to include dynamic state to better model epidemic growth of attacks. A model of the growth of the threat includes the fraction of the network under attack as an additional unobserved state variable, and a stage transition model for the growth of this fraction given the absence or presence of attack. Preliminary results for these models are very promising, but an extensive evaluation is ongoing. We are also actively making detailed comparisons of performance versus resource usage and robustness to be able to identify under which conditions one model might be preferred to another. In addition, simulations are underway that use real background data collected in a large enterprise network in combination with a range of known and hypothetical worm dynamics.

### 3 Conclusions

We have proposed a class of inference models that can be broadly characterized as varieties of DBNs that meet requirements for detection of network states. By combining the outputs of host detectors we can create fast detectors with limited bandwidth demands and low spurious detection rates. The inspiration for this work grew out of concerns for detecting network worms, specifically scanning-worms, and for detecting epidemic growth in a large population [Cooper *et al.*, 2004]. These models generalize to a wider class of distributed network detection problems. The next modeling concerns that will drive this research program arise from the design of network protocols fitted to these algorithms and reaction and containment strategies triggered by detection. Introduction of spatial aspects, such the source of an attack, the location of infected nodes and the connectivity of the network also raise new concerns.

### References

- Gregory Cooper, Denver Dash, John Levander, Weng-Keen Wong, William Hogan, and Michael Wagner. Bayesian biosurveillance of disease outbreaks. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 94–103, Arlington, Virginia, 2004. AUAI Press.
- Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.
- Tom Fawcett and Foster Provost. Activity monitoring: noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 53–62, New York, NY, USA, 1999. ACM Press.
- Douglas Hawkins and Peihua Qiu. The changepoint model for statistical process control. *Journal of Quality Technology*, 35(4):355–366, 2003.

- Jaeyeon Jung, Vern Paxson, Arthur Berger, and Hari Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 211–225, 2004.
- Leslie Kaelbling, Michael Littman, and Anthony Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
- Robert Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2001.
- Nicholas Weaver, Stuart Staniford, and Vern Paxson. Very fast containment of scanning worms. In *Proceedings of the 13th USENIX Security Symposium*, pages 29–44, 2004.