Policies for Sharing Distributed Probabilistic Beliefs

Christopher Leckie† **Ramamohanarao Kotagiri**‡

ARC Special Research Centre for Ultra-Broadband Information Networks
†Department of Electrical and Electronic Engineering
‡Department of Computer Science and Software Engineering
The University of Melbourne, Victoria 3010, Australia
Email: c.leckie@ee.mu.oz.au, rao@cs.mu.oz.au

Abstract

In this paper, we present several general policies for deciding when to share probabilistic beliefs between agents for distributed monitoring. In order to evaluate these policies, we have formulated an application in network intrusion detection as a multi-agent monitoring problem. We have evaluated our policies based on packet trace data from a real network. Based on this evaluation, we have demonstrated that our policies can reduce both the delay and communication overhead required to detect network intrusions. Although we have focused on network intrusion detection as an application, we contend that our policies can generally be applied to domains that use a probabilistic model for evaluating hypotheses, and have a method for combining beliefs from multiple agents.

Keywords: multi-agent systems, network intrusion detection

1 Introduction

The problem of monitoring distributed sensors arises in a variety of applications, such as aircraft tracking using radar, fire tracking using infra-red detectors, or even tracking network congestion using packet trace probes. In each case, the aim of monitoring is to use sensor measurements to evaluate a hypothesis, such as the path of a vehicle or the source of a network fault. Typically, the sensor measurements are downloaded to a centralised site, where they are analysed using a Bayesian framework to evaluate the hypothesis.

Centralised monitoring of distributed sensors can become impractical when the number of sensors is large, or the frequency of measurement is high. The large volume of measurements means that high bandwidth connections are needed to download the measurements to a central site. For example, if data network traffic must be analysed at the packet level, then the bandwidth required to download the packet measurements can approach the total bandwidth of the monitored network.

An alternative is to use a multi-agent approach, where more intelligence is located near the sensors. Each agent monitors a subset of sensors, and evaluates the hypothesis based on the local measurements of its sensors. If the agent believes that a hypothesis is sufficiently likely based on these local measurements, then it can exchange information with other agents in order to combine their beliefs about the hypothesis. The agents can then reach a global consensus about the likelihood of the hypothesis.

A major advantage of the multi-agent approach to monitoring is that agents exchange only high-level information about hypotheses that are considered sufficiently likely based on local measurements. This reduces the volume of data that is exchanged between agents, thus reducing the bandwidth and cost of the monitoring network. It also eliminates the centralised analysis site as a potential bottleneck.

A key research issue for this type of multi-agent approach is deciding when agents should share their beliefs about a hypothesis. An agent's belief in a hypothesis can change as a result of new local measurements, or new information received about other agents' beliefs. When a new local measurement causes an agent to update its belief, the agent must decide whether this change in belief is significant enough to warrant updating the beliefs of other agents. If the agent exchanges information with other agents each time it receives a new measurement, then it wastes bandwidth communicating insignificant changes in belief. Agents can save bandwidth by waiting until they have observed a large change in the likelihood of a hypothesis due to several new local measurements. However, if an agent waits too long before exchanging information, then the global confirmation of a hypothesis may be delayed due to a lack of shared evidence.

In summary, agents need a policy on when to share information about a hypothesis. This policy should specify what is a significant change in an agent's belief, such that the change in belief warrants communication with other agents. The choice of what constitutes a significant change in belief is a trade-off between minimising the communication between agents, and minimising the delay in confirming a hypothesis.

In this paper, we present several general policies to guide when agents should share their probabilistic beliefs in a hypothesis. We propose that these policies apply to any multi-agent monitoring problem that can be formulated as follows. Given a set of measurements D, our aim is to test whether to accept a hypothesis H_1 in preference to a null hypothesis H_0, i.e., we believe H_1 if $P(H_1|D) > P(H_0|D)$. Suppose that the global set of measurements is comprised of local measurements from k distributed agents, i.e., $D = D^1 \cup \ldots \cup D^k$. Rather than exchanging raw measurements, the agents exchange local summary information $f(D^k)$, such that $P(H|D) \simeq P(H|f(D^1), \ldots, f(D^k))$. This summary information can then be combined to evaluate H_1 and H_0. However, agents require a policy to decide when to share their local information $f(D^k)$ with other agents.

In order to evalaute these policies, we have formulated an application in network intrusion detection as a multi-agent monitoring problem. We have evaluated our policies using this application, based on packet trace data from a real network. This evaluation demonstrates how alternative policies make a trade-off between bandwidth versus delay.

The next section introduces network intrusion detection as our application domain. We develop a distributed model for combining beliefs in the context of intrusion detection. We then describe several general policies for when to share beliefs between agents, and illustrate how

these policies can be used in our application domain. We conclude with an empirical evaluation of how these policies reduce communication overhead and delay when detecting network intrusions.

2 Network Intrusion Detection

The primary role of network intrusion detection systems is to alert system administrators to suspicious network activity based on an analysis of packet traces. The challenge for network intrusion detection systems is to analyse the distribution of packets that are being sent to a network. Based on these distributions, we can detect unusual patterns of accesses that are likely to constitute an attack. In this context, our sensors are the probes used to collect packet traces, our measurements are the packet traces, and our hypothesis is whether a given source of packets is an attacker.

Our focus is one of the most common forms of network intrusion, known as network scans (Northcutt 2000). Network scans are widely used by attackers as a way of mapping the structure and configuration of a target network. For example, a *host scan* involves sending packets to a range of network addresses in order to find out which addresses represent an active host. This activity is often a prelude to a more damaging attack. Consequently, there is a need for effective scan detection techniques to identify possible sources of attacks and to prevent attackers from mapping target networks. For a detailed discussion of different types of network scans see (Northcutt 2000).

Let us begin by defining the terminology that we use to describe this problem. We refer to the network that we are defending as the *target network*. An *access* is an attempt by a source to initiate contact with a service on another host.

Simple forms of scans appear as a sequential pattern of accesses in a short period of time, e.g., accesses to a sequence of IP addresses or port numbers. This type of access pattern can be easily detected. However, more sophisticated attackers are starting to hide their intentions using a range of measures, such as randomising the order of their accesses, scanning over a longer period of time, or initiating the scans from a number of different sources.

In a previous paper (Leckie and Kotagiri 2002) we proposed a probabilistic approach to scan detection that takes into consideration both the number of hosts accessed by a source, as well as how unusual these accesses are. This approach is designed for use in a centralised intrusion detection system. However, a centralised intrusion detection system can suffer from scalability problems when applied to very large networks. An important issue is how to extend this approach to distributed intrusion detection, where several autonomous detection systems share evidence that they have collected from their own local subnetwork. This raises the question of when to share information from different sub-networks about the same suspicious source.

We have extended our earlier centralised scan detection model to a multi-agent intrusion detection framework. Before we describe our distributed model, let us begin by outlining the key features of our previous scan detection technique.

2.1 Centralised Scan Detection

Every time a source makes a new access to our target network, we need to update the access distributions for that source, and then decide whether these accesses constitute a host scan. To this end, we developed a likelihood measure that can be applied to the destination distribution of a source.

Let $\mathcal{D} = \{d_1, d_2, d_3, \dots\}$ denote the set of destinations in our target network, such that each destination has been accessed by an external source. In addition, let $\mathcal{D}_i$ denote the subset of destinations that have been accessed by the source s_i. For each source s_i, our aim is to quantify a measure of how suspicious are the accesses from that source. Two main factors are modelled in this measure of suspicion: (1) how unusual is each individual destination that has been accessed by s_i, and (2) how many destinations have been accessed by s_i. Let us consider each of these factors in more detail.

The first factor captures how unusual are the destinations accessed by a source s_i. It is considered unusual for a source to access a destination d_j that has been accessed by few other sources. Let $n_s(d_j)$ represent the number of sources that have accessed destination d_j in the access trace. The prior probability of a destination d_j being accessed by a normal source is defined as

$$P_n(d_j) = \frac{n_s(d_j)}{N_s},$$

$$\text{where} \quad N_s = \sum_{all\ d_{j'} \in \mathcal{D}} n_s(d_{j'})$$

is calculated over all destinations that have been accessed in the target network. In contrast, attackers tend to access destinations at random. Consequently, a uniform prior distribution can be used for the probability P_a of a destination being accessed by an attacking source, i.e.,

$$P_a(d_j) = \frac{1}{|\mathcal{D}|}.$$

The second factor in the measure of suspicion is how many destinations have been accessed by a source. Normal sources are typically interested in a small number of destinations. In contrast, a source that accesses a large proportion of the hosts in the target network is likely to be an attacker who is trying to map the structure of the network.

In (Leckie and Kotagiri 2002) we showed that the probability distribution of a given set of accesses, assuming a normal source and an attack source, is as follows:

$$P_n(\mathcal{D}_i) = \frac{1}{K(e-1)\,|\mathcal{D}_i|!} \prod_{d_j \in \mathcal{D}_i} P_n(d_j),$$

$$P_a(\mathcal{D}_i) = \frac{1}{K\,|\mathcal{D}|} \prod_{d_j \in \mathcal{D}_i} P_a(d_j),$$

$$\text{where} \quad K = \binom{|\mathcal{D}|}{n} \frac{1}{|\mathcal{D}|^n}$$

is a normalisation constant. A source is considered to be an attacker if $P_a(\mathcal{D}_i) > P_n(\mathcal{D}_i)$.

The main differences between $P_n(\mathcal{D}_i)$ and $P_a(\mathcal{D}_i)$ are the distributions used to estimate the probability of each d_j in $\mathcal{D}_i$, and the factorial term $|\mathcal{D}_i|!$. The factorial term in $P_n(\mathcal{D}_i)$ means that if a source accesses a large number of destinations, then it is less likely to be considered as normal than as an attack source. This corresponds to the intuition that an attacking source is trying to sample a large proportion of the destinations on the target network.

2.2 Distributed Scan Detection

In (Leckie and Kotagiri 2002) we proposed a centralised architecture for scan detection, where a single intrusion detection system analyses the packet trace from the entire target network. Let us now consider the issues that arise

when we extend this approach to a distributed, multi-agent architecture.

Consider a target network containing 10 hosts, i.e., $\mathcal{D} = \{d_1, d_2, \ldots, d_{10}\}$. In addition, consider a source s_i that accesses 4 of these hosts, $\mathcal{D}_i = \{d_4, d_5, d_6, d_7\}$, such that this set of accesses is sufficient to constitute a scan attack, i.e., $P_a(\mathcal{D}_i) > P_n(\mathcal{D}_i)$. A centralised intrusion detection system would monitor all the measured accesses to the target network, and use these measurements to test the hypothesis that s_i is an attacker. Based on these measurements, we have sufficient evidence to believe that the hypothesis is true.

Now consider a multi-agent scenario where the target network is divided into two subnetworks, each with its own intrusion detection system. For convenience, we shall call these the left and right subnetworks $\mathcal{D}^L = \{d_1, \ldots, d_5\}$ and $\mathcal{D}^R = \{d_6, \ldots, d_{10}\}$. Each agent only sees the accesses that are made to its subnetwork. Consequently, the accesses by s_i are split between the two agents as follows: $\mathcal{D}_i^L = \{d_4, d_5\}$ and $\mathcal{D}_i^R = \{d_6, d_7\}$. Each agent may believe that such a small number of accesses is not suspicious, i.e., $P_a(\mathcal{D}_i^L) < P_n(\mathcal{D}_i^L)$ and $P_a(\mathcal{D}_i^R) < P_n(\mathcal{D}_i^R)$. Thus, each agent acting in isolation has insufficient evidence to consider the source to be suspicious.

In order to detect this scan, the two agents need to cooperate by sharing their beliefs about potentially suspicious sources. This raises two challenges. First, we need a probabilistic framework for combining different agents' beliefs about a source. In particular, we must be able to combine beliefs without having to share the raw accesses upon which those beliefs are based. Otherwise the communication overhead would be prohibitive. Second, we need a policy for deciding when to share beliefs about a source. We have extended our centralised model in (Leckie and Kotagiri 2002) to address these two challenges, as described in the following sections.

3 Combining Beliefs

Our aim is to calculate the likelihood of a source s_i being normal or an attacker by combining beliefs from each intrusion detection agent. It is important that our model for combining beliefs should use summary information about the source rather than raw measurements about each access made by the source, in order to minimise the communication overhead. Without loss of generality, we consider a network that is divided into two subnetworks, each with its own intrusion detection agent. It is a trivial matter to apply our model to larger numbers of agents. Let $\mathcal{D}^L$ and $\mathcal{D}^R$ denote the set of hosts in the left and right subnetworks, respectively. Note that these sets are disjoint, since the same IP address cannot appear on multiple subnetworks. Similarly, $\mathcal{D}_i^L$ and $\mathcal{D}_i^R$ denote the set of hosts that have been accessed by source s_i in the left and right subnetworks, respectively.

Our first step is to extend the model of Leckie and Kotagiri to calculate the combined probability of the source being an attacker $P_a(\mathcal{D}_i^L \cup \mathcal{D}_i^R)$. Given that $|\mathcal{D}| = \left|\mathcal{D}^L\right| + \left|\mathcal{D}^R\right|$ and $|\mathcal{D}_i| = \left|\mathcal{D}_i^L\right| + \left|\mathcal{D}_i^R\right|$, it is a simple matter to substitute these expressions into the expression for $P_a(\mathcal{D}_i)$ from the previous section. Thus, we can show that $P_a(\mathcal{D}_i^L \cup \mathcal{D}_i^R)$ is a function only of summary statistics from each agent, i.e.,

$$P_a(\mathcal{D}_i^L \cup \mathcal{D}_i^R) = f_a(\left|\mathcal{D}^L\right|, \left|\mathcal{D}^R\right|, \left|\mathcal{D}_i^L\right|, \left|\mathcal{D}_i^R\right|).$$

Our second step is to express the model for the combined probability of the source being normal $P_n(\mathcal{D}_i^L \cup \mathcal{D}_i^R)$ in terms of summary statistics from each agent. The main challenge is rewriting the expression

$$\prod_{d_j \in \mathcal{D}_i} P_n(d_j)$$

from $P_n(\mathcal{D}_i)$ in terms of information from the left and right subnetworks. Each agent can calculate the prior probability of a destination d_j in its subnetwork being accessed by a normal source. For example, the agent for the left subnetwork can calculate

$$P_n^L(d_j) = \frac{n_s(d_j)}{N_s^L},$$

$$\text{where} \quad N_s^L = \sum_{all\ d_{j'} \in \mathcal{D}^L} n_s(d_{j'}).$$

Similarly, the agent for the right subnetwork can calculate $P_n^R(d_j)$ using N_s^R. We can then make the substitution:

$$\prod_{d_j \in \mathcal{D}_i} P_n(d_j) = \prod_{d_j \in \mathcal{D}_i^L} \frac{N_s^L P_n^L(d_j)}{N_s^L + N_s^R} \prod_{d_j \in \mathcal{D}_i^R} \frac{N_s^R P_n^R(d_j)}{N_s^L + N_s^R}.$$

We can now rewrite the expression for $P_n(\mathcal{D}_i)$ from the previous section in terms of summary statistics from each agent. Thus, we can show that $P_n(\mathcal{D}_i^L \cup \mathcal{D}_i^R)$ is a function only of summary statistics from each agent, i.e.,

$$P_n(\mathcal{D}_i^L \cup \mathcal{D}_i^R) = f_n(\left|\mathcal{D}^L\right|, \left|\mathcal{D}^R\right|, \left|\mathcal{D}_i^L\right|, \left|\mathcal{D}_i^R\right|, N_s^L, N_s^R, \prod_{d_j \in \mathcal{D}_i^L} P_n^L(d_j), \prod_{d_j \in \mathcal{D}_i^R} P_n^R(d_j)).$$

This means that we can calculate the likelihood of a source being normal or an attacker based on summary statistics from each subnetwork, without having to communicate raw measurement data between agents. If an agent wants to update another agent with its beliefs about a source, it only needs to broadcast the 4-tuple:

$$\langle \left|\mathcal{D}^L\right|, \left|\mathcal{D}_i^L\right|, N_s^L, \prod_{d_j \in \mathcal{D}_i^L} P_n^L(d_j) \rangle.$$

Note that the final term in the 4-tuple combines information about a potentially large number of accesses into a single statistic. This dramatically reduces the amount of data that needs to be communicated between agents. When an agent wants to calculate the likelihood of a source being normal or an attacker, it uses the most recent broadcast (if any) from other agents about that source.

We now have a method for combining beliefs about a source from different agents. The next challenge is to formulate a policy for *when* to broadcast this information, and hence combine beliefs between agents.

4 Policies for When to Combine Beliefs

The reason for sharing beliefs between agents is to provide a more accurate evaluation of a hypothesis based on a global set of measurements than if we rely on local measurements alone. In the intrusion detection application, each access helps us determine whether a source is likely to be an attacker. A naive policy for deciding when to share beliefs would be to broadcast our beliefs to other agents every time we observe an access by a source. This liberal policy effectively replicates all measurements at all agents. Given that most sources are normal and thus of little interest to other agents, this approach would be an enormous waste of bandwidth and resources. Even for

potentially suspicious sources, individual accesses do little to change our beliefs.

An alternative policy that is equally naive would be to broadcast our beliefs only when we have sufficient local measurements to confirm our hypothesis. Each source is independently monitored by each agent until one agent confirms that the source is an attacker, and then notifies the other agents. This conservative policy for sharing beliefs minimises communication overheads. However, it also maximises the delay in confirming a hypothesis because beliefs are not shared about unconfirmed hypotheses.

Our aim is to find a policy that lies somewhere between the extremes of these two naive approaches. Agents should share information when there is a significant change in the belief that is likely to help confirm a hypothesis. Although we describe our policies in terms of the intrusion detection application, these policies are general statements that can be applied in a wide range of applications for combining beliefs. Our main assumptions are that we use a probabilistic model for evaluating hypotheses, and we have a method for combining beliefs from multiple agents.

The first policy that we have considered is to broadcast after an agent has received M new measurements relating to a hypothesis, where M can be a constant or a random variable. This policy is simple to implement, but does not consider how the measurements affect our change in belief. It is also extremely difficult to choose a suitable value of M *a priori*. For this reason, we choose M from a uniform distribution $Uniform(1, M_{max})$. This enables the agent to try different broadcast periods, without being stuck on a value of M that is too large or too small. We use this policy as a benchmark to explore the trade-off between communication overhead and confirmation delay by varying M_{max}.

Policy 1 *Broadcast belief in a hypotheses each time M new measurements have been seen for the hypothesis, where $M := Uniform(1, M_{max})$ is reset after each broadcast.*

A more informed approach is to take into consideration the current degree of belief in the hypothesis. The optimum time to broadcast is when the combined measurements from all agents are just sufficient to confirm the hypothesis. Obviously, this is not known beforehand. A way around this problem is for each agent to assume that the other agents have seen a similar set of measurements. An agent can then estimate the combined belief in the hypothesis using the assumed measurements of the other agents. If this estimation of the combined belief would confirm the hypothesis, then the agent should broadcast its belief.

If we consider the two agent intrusion detection scenario as an example, this means that the agent for the left subnetwork would assume $\mathcal{D}_i^R = \mathcal{D}_i^L$, and vice versa. In effect, each measurement is counted twice in order to make a global estimate of P_a and P_n. We denote these estimates as $P_a(2\mathcal{D}_i^L)$ and $P_n(2\mathcal{D}_i^L)$. In general, if there are k agents, then each measurement would be counted k times, which we denote as $P_a(k\mathcal{D}_i^L)$. By assuming $\mathcal{D}_i^R = \mathcal{D}_i^L$, the agent for the left subnetwork can estimate whether the source s_i would have been considered suspicious if the agent for the right subnetwork had seen a similar set of accesses. If $P_a(2\mathcal{D}_i^L) > P_n(2\mathcal{D}_i^L)$, then the agent for the left subnetwork would broadcast its beliefs to the other agent. In practice, the other agent may have seen far fewer measurements to support the hypothesis, in which case further broadcasts may be required.

Policy 2 *Broadcast when $P_a(k\mathcal{D}_i^L) > P_n(k\mathcal{D}_i^L)$, then rebroadcast after every M measurements, where $M := Uniform(1, M_{max})$.*

A variation of Policy 2 is to use a different strategy for choosing the rebroadcast period M. Rather than choosing M randomly, we can base it on the number of measurements seen before the first broadcast was made. In this case, broadcasts are periodic, based on an agent's estimate of when the other agents have seen sufficient evidence.

Policy 3 *Broadcast when $P_a(k\mathcal{D}_i^L) > P_n(k\mathcal{D}_i^L)$. Set $M := \left|\mathcal{D}_i^L\right|$. Rebroadcast after every M new measurements.*

Policies 2 and 3 assume that each agent collects similar measurements about the hypothesis. However, in practice one agent is likely to have more evidence than the others. For example, an attacker's accesses might not be uniformly distributed between subnetworks. We may be assuming that the other agent has more evidence than it actually has. Consequently, we may require multiple broadcasts, and may wait too long between broadcasts. For this reason, we propose two policies to reduce the time between broadcasts.

Policy 4 *Broadcast when $P_a(k\mathcal{D}_i^L) > P_n(k\mathcal{D}_i^L)$. Set $M := \left|\mathcal{D}_i^L\right|/2$. Rebroadcast after every M new measurements.*

Policy 5 *Broadcast when $P_a((k+1)\mathcal{D}_i^L) > P_n((k+1)\mathcal{D}_i^L)$. Set $M := \left|\mathcal{D}_i^L\right|/2$. Rebroadcast after every M new measurements.*

Our expectation is that a more aggressive approach to broadcasting will reduce the delay in confirming the hypothesis, without substantially increasing the number of broadcasts.

5 Evaluation

We have evaluated our our policies for sharing beliefs and our distributed model for scan detection by simulating scans based on traffic measurements from a real-life network. We have based our simulation on a public domain packet trace file from the Measurements and Operations Analysis Team (MOAT), which is part of the National Laboratory for Applied Network Research (NLANR). Their website (http://moat.nlanr.net/Traces) contains regular dumps of packet trace files from various connection points to the vBNS backbone network. Each packet trace file contains timestamped packet headers for the IP packets that traversed the link over a short collection period. For the purposes of our evaluation, we have selected a packet trace file from the vBNS link to Colorado State University for 23 November 2000. The selected MOAT trace file (COS-975003490-1.tsh.enc.gz) contained over 384,000 packets. In this file, we found that 974 sources generated 6237 accesses to 317 destination hosts in the target network.

We used this packet trace to identify the set of destinations $\mathcal{D}$ in the target network, and the prior probability of normal sources accessing these destinations, i.e., $P_n(d_j)$ for all $d_j \in \mathcal{D}$. We then simulated scans to the 30 most popular destinations. The most popular destinations were chosen because these are the least suspicious destinations to access, thus making the scan harder to detect.

We tested our model in a two agent scenario, where the 30 destinations were randomly assigned to two subnetworks, each with its own agent. Each time an agent observes an access to a destination on its subnetwork, it re-evaluates its belief in whether that source is an attacker, e.g., $P_a(\mathcal{D}_i^L) > P_n(\mathcal{D}_i^L)$. If any evidence has been broadcast from the other agent, then it is included in this evaluation. The agent also applies the given policy to determine if it should share its beliefs with the other agent.

Once an agent has confirmed that the source is an attacker, we record the total number of accesses that were made by the source before it was detected, as well as the number of broadcasts received by the agent before it reached its conclusion. We also determined the number

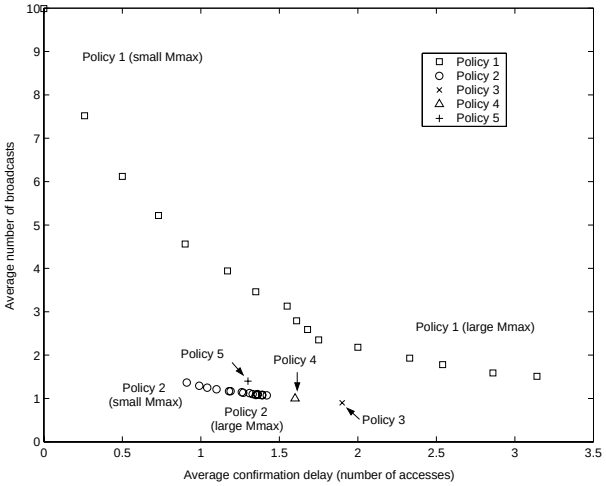

Figure 1: Evaluation of the policies for two agents

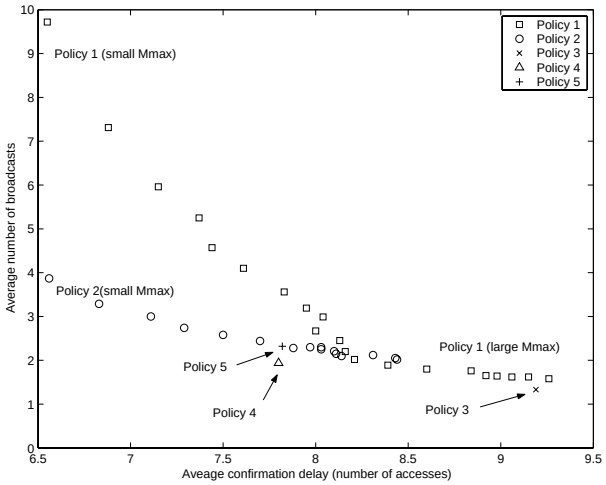

Figure 2: Evaluation of the policies for three agents

of accesses that would have been required by a centralised agent in order to confirm that the source was an attacker. The difference between the number of accesses needed by the multi-agent system and the number needed by the centralised system represents the *confirmation delay* in using a distributed approach.

We used this procedure to evaluate each policy in terms of the number of broadcasts needed and the confirmation delay. For each policy, we averaged the results over 1000 trials, where each trial represents a simulated scan with a random assignment of destinations to subnetworks. In addition, for Policies 1 and 2 we repeated each batch of 1000 trials using 16 settings of M_{max} from 1 to 16. The results for all 5 policies are shown in Figure 1.

The results for Policies 1 and 2 form a curve, with small values of M_{max} on the left, and large values on the right. Clearly, as M_{max} increases, the frequency of broadcasts decreases, and the confirmation delay increases. Any sensible policy should fall below the envelope formed by Policy 1.

In contrast, Policy 2 is much more consistent as M_{max} varies, using between 1 to 1.4 broadcasts on average, and an average confirmation delay of 0.9 to 1.4. This significant improvement over Policy 1 demonstrates the advantage of an informed estimate for when to make the first broadcast.

Policy 3 required on average 1.9 accesses more than the centralised approach, using 0.9 broadcasts. It used the smallest average number of broadcasts, but had a significantly larger average confirmation delay compared to Policy 2.

Policy 4 has an average confirmation delay of 1.6 accesses, which is a 16% improvement over Policy 3. To achieve this improvement, it required 1.0 broadcasts on average, i.e., 11% more than Policy 3. In this case, the decrease in confirmation delay outweighs the increase in communication overhead. Policies 2, 3 and 4 demonstrate that a more aggressive rebroadcast policy can decrease confirmation delay without dramatically increasing communication.

Policy 5 has an average confirmation delay of 1.3 accesses, which is a further 19% improvement over Policy 4. It achieves this improvement at the expense of 1.4 broadcasts on average, which is 40% more than Policy 4. The reason for the increase in broadcasts is that Policy 5 broadcasts earlier, which gives a greater opportunity for a second broadcast. Note that Policy 5 has the closest performance to Policy 2, without the uncertainty of how to choose M_{max}.

We have repeated these simulations in a three agent scenario, where the destinations are randomly assigned to three subnetworks. The results are shown in Figure 2. In this case, the relative performance of the policies exhibited the same trend as in the two agent case. However, the average confirmation delay for all policies is inherently higher in the three agent case, which is to be expected since each agent sees less evidence and is more reliant on broadcasts. In particular, the results for Policy 2 demonstrates the importance of rebroadcasting. This is because we are less likely to get the timing right for the first broadcast when each agent sees fewer accesses. This is also reflected in the results for Policies 4 and 5, where the difference in the timing of the first broadcast has little impact on the relative performance of these two policies. In contrast, Policy 3 fares much worse because it is more conservative in terms of broadcasting.

These results demonstrate that it is possible to formulate a general policy for when to share beliefs, without requiring any prior knowledge of the underlying domain. Furthermore, we have found policies that do not rely on arbitrarily chosen broadcast periods. Instead, agents use their own experience to estimate the likely beliefs of other agents.

6 Related Work

The topic of distributed sensor interpretation has been the subject of extensive investigation by Carver and Lesser, using a model called Functionally Accurate, Cooperative (FA/C) distributed problem solving. They highlight the issue of how to generate high quality global solutions from several local solutions without excessive communication between agents. In (Carver et al. 1996), they discuss the case of *nearly monotonic* problem domains, where good local solutions to a problem are likely to constitute good global solutions. Our multi-agent model of scan detection is an example of such a domain. However, if agents wait too long in order to guarantee a high degree of belief in their local solution, they risk delaying a global confirmation of the hypothesis. This leads to the key issue of this paper, namely, when should agents share their local beliefs in order to form a global confirmation of the hypothesis. We propose that our policies can be used in a variety of applications of this form.

Luo and Tsitsiklis (1996) have also investigated decentralised approaches to detection and hypothesis testing. They investigated how much information needs to be communicated by agents in order to compute a decision function at a central location. Their key results pertain to linear functions of Gaussian random variables. In addition, they assume continuously differentiable functions. In contrast, our application involves nonlinear functions of discrete random variables. Although they are interested in minimising communication bandwidth, they focus on centralised hypothesis testing, whereas we consider distributed hypothesis testing.

Pennock and Wellman (1997) have studied the problem of belief aggregation in multi-agent systems using a

market-based approach. They provide a general model for aggregating beliefs, but they do not consider when to communicate beliefs to other agents in resource-bounded environments.

Heinzelman et al. (1999) have studied how to monitor distributed sensors under communication resource constraints, such as bandwidth, computation and energy. They propose a method of message propagation that includes resource constraints between agents. However, they do not consider the problem of cooperative problem solving.

Xiang and Geng (1999) present a method for distributed monitoring and diagnosis using multiply sectioned Bayesian networks in the context of digital circuit diagnosis. Each monitored component is modelled as a Bayesian subnetwork, and they apply techniques for propagating evidence between subnetworks. They highlight ongoing research issues for their model in modelling the cost of communication, and handling dynamic environments where the system state evolves over time. It would be interesting to investigate whether our policies could be applied to their model.

The problem of distributed scan detection has also been studied by Ohta et al. (2000). They analysed traffic from an existing data network in order to demonstrate the advantages of aggregating evidence compared to isolated detection. They propose the idea of looking for similar traffic patterns in different subnetworks. However, they provide no discussion of how to implement a distributed approach to coordinating this correlation task.

7 Conclusion and Further Work

We have presented several general policies for deciding when to share probabilistic beliefs between distributed agents. In order to evaluate these policies, we have formulated an application in network intrusion detection as a multi-agent monitoring problem. We have evaluated our policies using simulations of network attacks based on packet trace data from a real network. Based on this evaluation, we have demonstrated that our policies can reduce both the delay and communication overhead required to detect network intrusions, in comparison to a default policy that relies on arbitrarily chosen broadcast periods.

Although we have focused on network intrusion detection as an application, we contend that our policies can generally be applied to domains that use a probabilistic model for evaluating hypotheses, and have a method for combining beliefs from multiple agents. An interesting issue for further research is to apply these policies in other distributed monitoring applications, such as vehicle tracking (Carver et al. 1996). Another directions for future work is to develop policies for broadcasting locally to neighbouring agents rather than to all agents.

8 Acknowledgements

We thank the MOAT group for making available their packet trace data, and we acknowledge the NSF NLANR/MOAT Cooperative Agreement (No. ANI-9807479), and the National Laboratory for Applied Network Research. This work was supported by the Australian Research Council.

References

Carver, N., Lesser, N. & Whitehair, R. (1996), Nearly Monotonic Problems: A Key to Effective FA/C Distributed Sensor Interpretation? *in* 'AAAI-96', pp. 88–95.

Heinzelman, W., Kulik, J. & Balakrishnan, H. (1999), Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, *in* 'Proc. Fifth ACM/IEEE Int. Conf. on Mobile Computing and Networking', pp. 174–185.

Leckie, C. & Kotagiri, R. (2002), A Probabilistic Approach to Detecting Network Scans, *in* 'Proc. IEEE Network Operations and Management Symposium', pp. 359–372.

Luo, Z.-Q. & Tsitsiklis, J.N. (1994), Data Fusion with Minimal Communication, *in* 'IEEE Transactions on Information Theory', 40(5), pp. 1551-1563.

Northcutt, S. & Novak. J. (2000), *Network Intrusion Detection: An Analyst's Handbook*, New Riders Publishing.

Ohta, K., Mansfield, G., Takei, Y., Kato, N. & Nemoto, Y. (2000), Detection, Defense, and Tracking of Internet-Wide Illegal Access in a Distributed Manner, *in* 'Proc. 10th Annual Internet Society Conference'.

Pennock, D. & Wellman, M. (1997), Representing Aggregate Belief through the Competitive Equilibrium of a Securities Market, *in* 'Proc. UAI-97', pp. 392-400.

Xiang, Y. & Geng, H. (1999), Distributed Monitoring and Diagnosis with Multiply Sectioned Bayesian Networks, *in* 'Proc. AAAI Spring Symposium on AI in Equipment Service Maintenance and Support', pp. 18-25.