

EVALUATION OF SECURE PEER-TO-PEER OVERLAY ROUTING FOR SURVIVABLE SCADA SYSTEMS

Jeffrey J. Farris
David M. Nicol

Department of Electrical and Computer Engineering
University of Illinois, Urbana-Champaign
Urbana, Illinois 61801, U.S.A.

ABSTRACT

Supervisory Control And Data Acquisition (SCADA) systems gather and analyze data for real-time control. SCADA systems are used extensively, in applications such as electrical power distribution, telecommunications, and energy refining. SCADA systems are obvious targets for cyber-attacks that would seek to disrupt the physical complexities governed by a SCADA system. This paper uses a discrete-event simulation to begin to investigate the characteristics of one potential means of hardening SCADA systems against a cyber-attack. When it appears that real-time message delivery constraints are not being met (due, for example, to a denial of service attack), a peer-to-peer overlay network is used to route message floods in an effort to ensure delivery. The SCADA system, and peer-to-peer nodes all use strong hardware-based authentication techniques to prevent injection of false data or commands, and to harden the routing overlay. Our simulations help to quantify the anticipated tradeoffs of message survivability and latency minimization.

1 INTRODUCTION

SCADA systems are pervasive for monitoring and control of large scale critical infrastructure applications such as electrical power generation and distribution, telecommunications, energy refining and transportation systems. Pressures of modernization, integration, cost, and security will force SCADA systems to migrate from closed proprietary systems and networks toward commercial off the shelf products and hardware, standard network protocols, and shared communications infrastructure. The shared communications infrastructure thus becomes an obvious target for disrupting a SCADA network. An attacker may engineer a denial of service attack that constricts or prevents the real-time delivery of SCADA messages, resulting in a loss of monitoring information or control of portions of the SCADA system. Physical attack on routing infrastructure may ac-

complish this; the attack vector we consider here is a bandwidth consumption (Mirkovic 2004). We explore through network simulation the use of a hardened peer-to-peer overlay network as a means of improving delivery of SCADA messages. As such network will have considerable size, our experiments consider a network topology (SCADA+peer-to-peer) that has more than one thousand active devices.

The issue of *trust* among SCADA system nodes, and overlay network nodes is critical. Without mechanisms to engender trust a system may be attacked by infiltration, injection of bad data, or interruption of message flow when the overlay network is in use. We assume the use of techniques such as those proposed in (Nicol, Smith, and Hawblitzel 2001), which describes how peer-to-peer networking among nodes equipped with specialized authentication hardware can enhance survivability of critical infrastructure to attacks on the networking infrastructure. The inherent decentralized nature of peer-to-peer networks, along with the ability to scale to large number of hosts and adapt to rapid changes in membership, provide good base attributes for a survivable system. These same attributes also present challenges to a centralized and tightly-controlled security model traditionally found in client-server architectures. Current peer-to-peer systems generally ignore the question of trust and security, but this is a fundamental requirement for critical infrastructure. In (Nicol, Smith, and Hawblitzel 2001) is proposed the use of secure co-processors and outbound authentication algorithms (Smith 2002) to allow members of the peer-to-peer network to authenticate the identity, software, and hardware of remote peers. This specialized authentication is combined with hardened communication such as IPSec to achieve the level of security necessary for critical infrastructure.

This paper focuses on the peer-to-peer aspects for enhanced survivability. We describe our model, and present simulation results from experiments that explore the trade offs of combining a SCADA system with

a survivable peer-to-peer overlay. Our evaluation looks into properties of a dynamic peer-to-peer overlay and a SCADA system enabled with such an overlay in the context of denial of service attacks on shared communications infrastructure.

2 MODEL

2.1 SCADA

A SCADA system is comprised of sensors, which report data to master-stations for processing and analysis. The computational and network capability of sensors varies based on the application. For example, a sensor may be as simple as a pressure gauge with analog outputs, or as sophisticated as an intelligent electronic device with remote communications capabilities (Beaver, Gallup, Neumann and Torgerson, 2002). In our study we assume next generation sensors that have computational and network capabilities sufficient for IPSec, TCP, UDP, and application level peer-to-peer overlay communication. We assume sensors are also actuators, able to respond to control directives from a master-station. We study a simplified model where all sensors interact with a single, common, master-station.

2.2 Network

We are interested in SCADA systems that are supported by non-dedicated communication infrastructure. Our model is characteristic of ordinary wireline networks, being populated with hosts, communication lines, routers, hubs, switches. A master-station is an ordinary host in this topology, as is a sensor. The communication fabric is assumed to be shared; the star topology of master-station and sensors is a virtual one. The physical path between sensor and master-station may be comprised of multiple links, routers, and switches. In our model we include many hosts that are not immediately part of the master-station/sensor network, indeed the survivability architecture we study uses these hosts to provide alternative routes between sensor and master-station.

The particular model we use in our study has as an essential building-block the “campus network” used in other studies of large-scale networks, e.g. (SSFNet 1999, Liljenstam, Liu, and Nicol 2003). The model network topology consists of a number of inter-connected campus networks. One campus network consists of 18 routers and 28 hosts, illustrated in Figure 1. All hosts acting as peer-to-peer nodes. In each campus network 5 hosts additionally act as SCADA sensors. The campus networks are connected in ring with additional chord connections across the ring. These campus networks all belong to a single AS and use a static version of OSPF

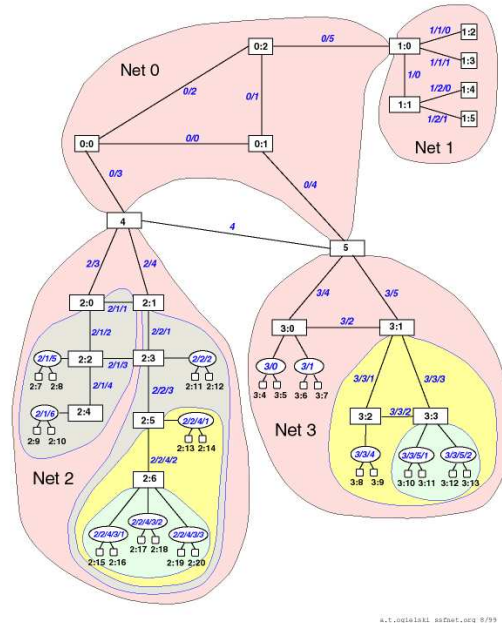


Figure 1: Base Campus Network

for routing between networks and inside networks. This static version of OSPF calculates fixed routes in zero time prior to the start of the simulation. In one campus network a single host is the SCADA master-station for the entire SCADA network.

2.3 Traffic Model

The communication model for SCADA traffic has three principle characteristics. One is of message *type*—sensors send data messages to the master-station, while the master-station sends control messages to sensor/actuators. Another is of message *frequency*—we assume that sensor \rightarrow master-station and master-station \rightarrow sensor messages are periodic. The third is that messages have *real-time deadlines* by which they should be received and acknowledged. We base assumed message characteristics on draft standards for power grid SCADA (IEEE 2003, IEEE 1994) for operations information: a message is considered to be delivered on-time if received within 10 seconds, and the period of sensor \rightarrow master-station messages is 60 seconds. As our present study focuses on network metrics (e.g., probability of message delivery, message latency) and as the time-scale of master-station \rightarrow sensor messages is typically longer, our study looks solely at sensor \rightarrow master-station traffic. We assume that the delay between successive messages from a given sensor has a Gaussian distribution with mean 60 seconds, and variance 1 second². To avoid artificial “waves” of message traffic, the send

time of a sensor’s first message is drawn uniformly at random from a “start-window”.

Again following proposed standards, message sizes are fixed at 89 bytes, composed of a 25 byte header and 64 byte payload. We assume that during normal operations sensors communicate their data to the master-station using the TCP protocol, as this gives reliable acknowledged communication. We later describe how a sensor detects communication failure, and shifts to an alternative form of communication involving a peer-to-peer network, and message flooding using the UDP protocol.

2.4 Threat Model and Detection

It is reasonable to presume that in the near future all devices involved in a SCADA infrastructure will be hardened against direct penetration and that communication will be hardened using IPSec, or something else that prevents spoofing and/or corruption of data. In this context there is a remaining threat due to use of shared communication infrastructure. Denial-of-service attacks of various kinds might still be mounted, a SCADA system is impacted when messages fail to be delivered on time. Late or failed messages reduce a SCADA’s ability to observe and control the system it governs.

A variety of means can be imagined that constrict the flow of SCADA messages. The easiest to imagine is if a high volume of traffic is directed along pathways used by SCADA traffic, in an effort to create congestion and consume bandwidth. This sort of threat might be anticipated and countered by requiring routers to reserve bandwidth for SCADA traffic, but a router may itself become a target of electronic attack (e.g., overwhelm it with spurious control traffic), or even physical attack (on the facility housing the router).

The anticipated threat then is against the shared communication infrastructure, with the effect of causing SCADA traffic to fail from being delivered on time. We assume that the SCADA continuously checks (every 60 seconds) for connectivity failure by including periodic “keep-alive” messages, from sensor to master-station, that report the path’s latency. The sensor maintains an exponentially decayed estimate of Round Trip Time (RTT), computing the n^{th} such, S_n , from the latest RTT estimate r_n and smoothing parameter α , via $S_n = \alpha S_{n-1} + (1-\alpha) r_n$. A large α (we use 0.8 in our experiments) dampens spurious large RTT samples. Recall that the real-time delivery (one-way) requirement for data messages is 10 seconds; when $S_n > 10sec$ (i.e., when the average one-way latency is 5 sec), the sensor considers its primary channel to the master-station as being unusable, and shifts over to peer-to-

peer overlay routing, to be described. It also considers the primary channel to be unusable if 2 RTT measurement instants (20 sec) go by without a response. Even as these other measures are used, the RTT of the primary channel is continuously estimated. When it drops threshold (again, 10 sec) the sensor reverts to the primary channel.

2.5 Peer-to-peer

The key idea we explore is that survivability is enhanced by providing alternative paths for SCADA traffic, paths that depend as little as possible on predictable mechanics of the existing communication infrastructure. If an adversary knows the source and destination of a flow he wishes to disrupt, with a predictable infrastructure he can intelligently guess which devices and links that carry that flow, and target one for disruption. We can confound such an adversary by diverting routing decisions to an application level (peer-to-peer overlay routing), where routing decisions are much less predictable. They can be made even more unpredictable when the overlay structure is itself dynamic and unpredictable.

Some peer-to-peer networks are *structured*, in the sense their topologies are strictly governed. Examples include CAN (Ratnasamy et al. 2001), Chord (Stoica et al. 2003), Tapestry (Zhao et al. 2001), and Pastry (Rowstron and Druschel 2001). Routing can then take advantage of that structure. For example, a peer-to-peer network organized as a two-dimensional mesh can use x-y virtual coordinates to efficiently move a message closer to its destination, with each hop. By contrast the topological properties of *unstructured* peer-to-peer networks are limited to constraints on a node’s connectivity. There is no inherent topological structure to be used to guide routing, so an alternative is to *flood* messages, as follows. When a message is originated, a time-to-live counter is initialized, the message is given a unique ID code, and the message is sent to each of the originator’s peers. Processing such a message, the host consumes the message if it is the intended destination. Failing this, the message’s time-to-live counter is decremented and the message is discarded on value zero. The ID of a surviving message is sought in a cache that records IDs of messages *already* relayed by this hosts, with the message being dropped if its ID is found. A message that survives all these filters is sent to all of the host’s peers (except of the peer that forwarded it to begin with), and the message ID is added to the sent-message-ID cache.

Our studies assume an unstructured network, as these appear to offer a higher degree of survivability than structured networks. However, higher survivability comes at the price of much increased communica-

tion. One of the goals of our study is to observe the magnitude of communication overhead added by an unstructured approach, and the degree to which it helps the network achieve real-time delivery goals.

Peer discovery is a key characterization of a peer-to-peer protocol. In our model, a host wishing to join the network acquires the identity of an existing member, and sends it a query. In a real system the existing member may be discovered using a well-known service, a cached list of neighbors from the last time the host was in the network, or member information obtained via out-of-band means. A queried host may choose to admit the querying host, or not. On rejection the new host is forced to query elsewhere. If accepted, the hosts admit each other to their respective peer tables, and the admitting host sends the querying one its own list of neighbors for use as peers. There is a configurable limit to the number of neighbors a peer will retain (in part to limit the multiplicative factor of flooding); a host that admits a new peer when its own peer table is full must drop an existing peer. Peering may be an asymmetric relationship—the dropped host’s peering table is unaffected by this action.

This basic peer discovery protocol treats all nodes equally and seeks to establish uniform number of neighbors among all peers, and not create centralized (i.e., highly connected) points of failure, to better increase survivability from attacks. Contrast this with other peer discovery protocols that seek improved performance by leveraging a few highly connected peers, as seen in some Gnutella-like systems (Sen and Wang 2004, Lv et al. 2002). This improved performance is at the expense of redundancy since failure of a highly connected peer can break connectivity in the peer-to-peer topology.

We simulate a system in which a host maintains two peer tables, *close*, and *far*. Whether two potential peers are close or far depends whether they are in the same IP subnetwork as each other (e.g. the same class B network, or some like definition of subnetwork). The intuition behind maintaining two sets of peers is that finding an alternate path around a network failure, where the location of that failure is unknown, may increase by choosing topologically diverse sets of peers. For example, choosing only far peers would not find an alternate path around a network failure in the case where that failure is on the gateway path to all of those far peers. However, the use of close peers that may have access to an alternate path around that gateway failure may increase the chances that the overlay successfully delivers the message. The choosing of optimal peers is an open question and is not dealt with here.

An important aspect to modeling a peer-to-peer system is the constant joining and leaving of peers from

the network. A departure from the network may occur when a node disconnects intentionally or due to some failure. For an intentional departure, the node notifies its peers, and they consequently remove it from their peering table. We incorporate the intentional joining and leaving of nodes by specifying an exponentially distributed departure rate λ_d and join rate λ_j . This models nodes joining the overall network from the inactive state at a rate proportional to $1/\lambda_j$ and leaving the network from the active state at a rate of $1/\lambda_d$. For this model, the join and departure rates are both set to 1/300 seconds.

In our model the peer-to-peer connections are implemented using UDP. They have a packet header of 25 bytes with additional data for peer-discovery messages and data routing messages.

2.6 DoS Attack

We model a bandwidth consumption denial of service (DoS) attack on portions of the network. Our bandwidth consumption attack represents a fixed rate source of traffic that overwhelms the resources of a network link, causing increased latency and packet loss along that path. We accomplish this by modifying the IP layer of the simulator to introduce packet losses and additional packet latency on links specified at configuration time, for durations also specified at configuration time.

3 EXPERIMENTAL RESULTS

Using the model above, we constructed two main sets of experiments. The goal of the first set of experiments was to understand some characteristics of the peer-to-peer network under normal conditions. The goal of the second set of experiments was to understand how the model SCADA system and overlay perform under DoS attack. We look at two simple metrics to describe the performance of the system. The on-time delivery ratio is the total number of messages received at the master-station within the 10 second real-time delivery requirement, divided by the total number of messages sent to the master-station. The message latency is the difference between the time the message was sent by the sensor and received by the master station. Using the SSFNet network simulator (SSFNet 1999), we ran simulations for a network comprised of 20 separate campus networks yielding a total of 360 routers, 100 sensors, 1 master-station, and 560 peer-to-peer hosts. Each experiment covered 1800 simulated seconds.

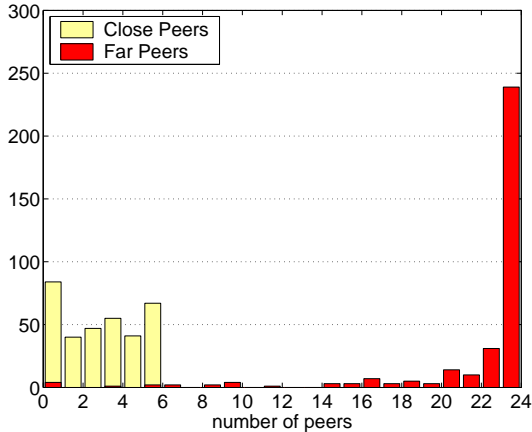


Figure 2: Peering Degree

3.1 Peer-to-peer System

We first examine the peer discovery protocol by observing how well connected peers are in the system. In this experiment, we let the peer-to-peer overlay run under normal operating conditions and record the size of the peering table for each peer at regular time intervals. The maximum number of close peers was set to 6 and the maximum number of far peers was set 24. Figure 2 shows a histogram plot of peering degrees taken from a snapshot of all peering tables at 720 seconds into the simulation. We observe that the peer discovery protocol does a good job of finding far peers, because the histogram shows a large concentration at the maximum of 24. The far peers with lower degrees are likely a result of the constant joining and leaving by peers. We also observe that the peer discover protocol does not perform as well finding close peers, since the peers degrees are not concentrated near the maximum of 6 close peers. The peer discovery mechanism and simulation network topology may explain this characteristic. First, the network topology does not allow for a portion of peers (those in subnet 1 for each campus network) to ever reach the goal of 6 close peers since these peers only have 3 other hosts in their network. Second, the peer discovery protocol finds peers only by contacting other peers. As the size of the network grows, the relative number of close peers for each peer decreases compared to the total number of peers in the network. Therefore when contacting other peers for information that may lead to the discovery of a close peers, it becomes less likely to discover a close peer as the size of the peer-to-peer network increases. Conversely, this also explains why the peer discovery protocol does a better job at discovering far peers.

Now we examine the message delivery performance of the peer-to-peer overlay under normal operating con-

Table 1: Peer-to-Peer Overlay Message Delivery Performance

	Shortest Path	Through Overlay
Delivery Ratio	$\mu=1.0 \sigma=0.0$	$\mu=0.998 \sigma=0.001$
Message Latency	$\mu=0.120 \sigma=0.061$	$\mu=0.280 \sigma=0.275$

ditions. In this experiment, we first set the SCADA sensors to use only the primary channel (shortest path routes from OSPF) and then we set the SCADA sensors to use only the peer-to-peer overlay to reach the master-station. Table 1 shows a comparison of message delivery performance of the shortest path and peer-to-peer overlay with μ and σ indicating the mean and standard deviation respectively across 5 independent simulation runs. The peer-to-peer overlay has a higher message latency mean and variance and a lower on-time delivery ratio. Higher latency may be attributed to topology—the shortest path between two nodes in the overlay network will always be as large as the OSPF path used by the normal routing infrastructure. Higher variance in the latency may be attributed to the dynamic nature of the overlay network, as nodes are continuously joining and leaving. This latter factor also explains the overlay’s lower delivery ratio.

3.2 DoS Attack

For these experiments, links affecting 20% of sensors and 14% of peers in the overlay are subjected to a bandwidth consumption DoS attacks. On the campus network in Figure 1, the link between router 2:0 and 4 is subjected to a bandwidth consumption attack across all 20 campus networks. We selected these links for attack because there is an alternate path to the destination. The simulations were run for 1800 simulated seconds, with the DoS attack starting at time 600 seconds, and ending at time 1200 seconds. In order to study sensitivity to loss rate, we experimentally varied the loss rate for the attacked links between 0.0 and 1.0 with a fixed 0.5 seconds latency increase for all packets. (The loss rate and latency in any given simulation run was fixed, we varied these parameters across runs.) In order to provide a basis for comparison, we subjected a model SCADA system utilizing only the primary communication channel and one that had the ability to use the peer-to-peer overlay. For these experiments, we continue to use a static version of OSPF calculated before the simulation begins, which means that the routing information is not updated during the attack. This was intentionally chosen so the characteristics of the peer-to-peer overlay can be examined in a

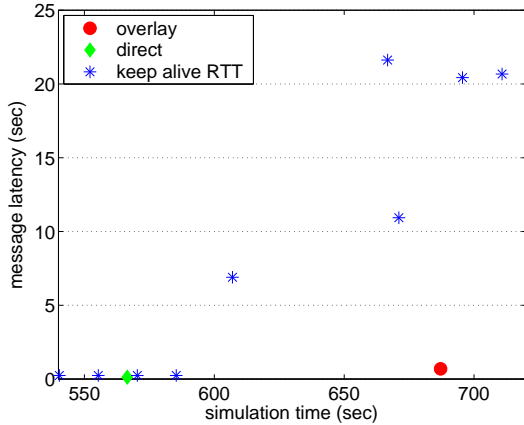


Figure 3: Detection Mechanism: Start of DoS Attack

fixed setting without the added variability of network layer routing updates. Consequently, the primary communications channel performs poorly in this scenario since it is forced to continue using the attacked link.

First we examine the threat detection mechanism that triggers use of the peer-to-peer overlay. We see in Figure 3 a snapshot of the estimated path RTT for a DoS attacked sensor and the selected communications channel versus simulation time for a packet loss rate of 0.3. The estimated path RTT increases dramatically after the DoS attack begins at time 600 seconds, due to TCP’s collision detection and retransmission algorithms. Recall that the threshold time for switching to the peer-to-peer overlay is 10 seconds, we see that the sensor uses the overlay after that threshold is crossed. Similarly in Figure 4, we see a snapshot of the same sensor at the end of the DoS attack at 1200 seconds. After the sensor starts to receive keep alive messages and updates the estimated round trip time, we again see messages sent via the primary communication channel. The delay before recognizing restoration of the primary channel is expected since (i) keep alive messages are sent only every 15 seconds, (ii) after the attack concludes some time is necessary for routing queues and the TCP channel to reach stabilize, (iii) with a delay parameter of $1 - \alpha = 0.2$ the exponential average “remembers” the contribution of large old RTT values for a few iterations of the estimation.

Now we examine the message delivery performance of the peer-to-peer overlay during the DoS attack. Figure 5 shows the on-time message delivery ratio for the affected sensors during the attack period for the system, with and without the peer-to-peer overlay. The error bars have width of the standard deviation, taken over 5 independent replications. The SCADA system without the peer-to-peer overlay shows a large drop in message

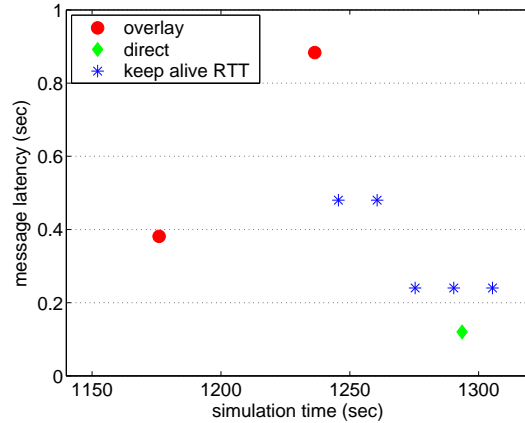


Figure 4: Detection Mechanism: End of DoS Attack

delivery as the packet loss rate increases. TCP congestion control and retransmission algorithms do not handle high packet loss rates well and so exacerbate the lost throughput capability. For the SCADA system that leverages the peer-to-peer overlay, on-time delivery of messages stays above 0.9 for packet losses up to 0.6 and remains at 0.7 for packet losses of 1.0 due to the overlay using alternate paths to the master-station. The decreasing peer-to-peer on-time message delivery ratio for increasing packet loss rates suggests that not all on-time messages took alternate paths. We looked at the peer paths for sensors messages arriving at the master-station and found that a portion of packets do traverse the attacked link in the peer-to-peer overlay (except when packet loss rate is 1.0). Since the peer-to-peer overlay uses simple flooding and does not update its own peering table frequently, it does not detect the attacked link and continues to try and use it. This turns out to be beneficial since UDP does not implement any congestion control or retransmission mechanisms and UDP packets will get through the attacked link with a probability equal to the complement of the loss probability. This suggests that in this type of attack, use of UDP is a good choice for the backup overlay network (a conclusion also reached by Birman et al. (2003), where a similar flooding mechanism is employed.)

Message latency during a DoS attack increases over that of a lightly loaded normal network, for two reasons. First, latency of TCP connections carried over attacked links increases because packet loss induces retransmission. Second, latency of messages carried by the overlay are larger because the path lengths are longer. In order to assess the *increase* in message latency, we plot the result of transforming each latency measurement by subtracting from it the minimal (ideal) path latency observed in a lightly loaded normal network. Figure

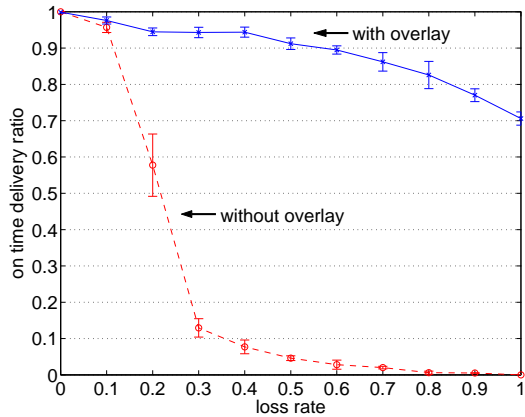


Figure 5: DoS Attack On-time Message Delivery Ratio

6 shows additional message latency for attacked sensors during the attack period in a system without the peer-to-peer overlay. The graph does not contain data for message latency after a packet loss rate of 0.7, because typically no packets were delivered to the master station in these experiments. We see increased mean message latency and high variance due to the congestion control and retransmission algorithms of TCP. Figure 7 shows additional message latency for attacked sensors during the attack period with a system that uses the peer-to-peer overlay. The increased mean message latency and high variance is explained by the different paths and number of hops messages take through the peer-to-peer overlay. This is evident by looking at the mean number of hops messages take through the overlay shown in Figure 8. As we would expect, the number of message hops increases with the packet loss rate as probability of getting through the attacked link decreases. Additionally, the high variance in the number of message hops also helps to explain the high variance in message latency seen for the peer-to-peer overlay.

4 RELATED WORK

Our use of overlay routing for performance or resistance to network failure is hardly novel. Andersen et al. (2001) propose a resilient overlay network (RON) architecture to detect and recover from network outages or degraded performance. Nodes in the RON perform active probing of all other nodes to build application layer link state routing tables. It is shown that RONs provided better performance and quicker recovery from failure than existing wide-area network layer routing protocols. Keromytis et al. (2002) propose a secure overlay services (SOS) network designed to proactively prevent a DoS attack. The SOS network utilizes secure tunneling, consistent routing of a structured peer-to-

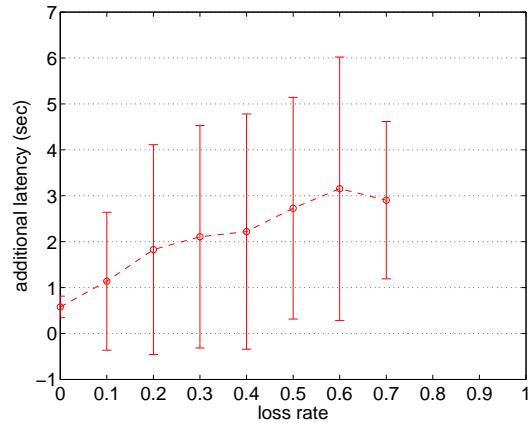


Figure 6: DoS Attack Additional Message Latency without Overlay

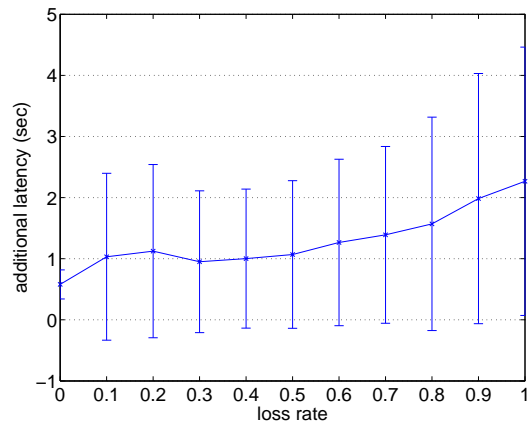


Figure 7: DoS Attack Additional Message Latency with Overlay

peer network, and filtering to reduce the probability of an attack on applications leveraging SOS. Analysis is presented that SOS significantly reduces the likelihood of a successful DoS attack. Wang and Chien (2002) study proxy networks based on peer-to-peer overlays as a means to reduce or eliminate effective DoS attacks. The proxy network provides application location hiding and rapid reconfiguration capabilities to prevent the attacker from targeting an application or host. Criteria for evaluating the effectiveness of an overlay to provide location hiding services are discussed and applied existing structured peer-to-peer networks.

Birman et al. (2003) propose protocols to improve the monitoring and controlling of power systems including peer-to-peer data sharing and bimodal multicast. The peer-to-peer data sharing protocol called Astrolabe provides a mechanism to distribute and query monitor-

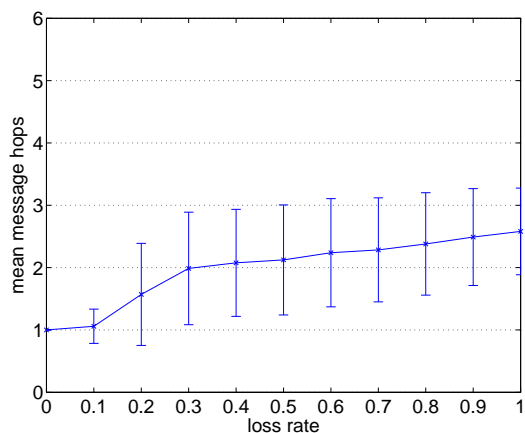


Figure 8: DoS Attack On-time Message Hops with Overlay

ing data across peer-to-peer network. Bimodal multicast provides a scalable way to notify a large number of hosts rapidly. Results from simulations of an actual power control system incident show improved delivery time and consistency for monitoring information.

5 CONCLUSION

We explored the use of a peer-to-peer overlay to improve the survivability of a SCADA system under a denial of service attack on a shared communications infrastructure. We identified a model for SCADA traffic, threat detection, a bandwidth consumption attack, and an unstructured peer-to-peer overlay. Simulations results show that in a bandwidth consumption DoS attack on the shared communications infrastructure of the SCADA system, the peer-to-peer overlay improves on-time message delivery. The peer-to-peer overlay's use of flooding is able to find alternate paths around the network congestion and deliver SCADA messages to the master-station. The results also show a trade off of higher message latency due to the peer-to-peer overlay's use of flooding for message delivery. The choice of peer-to-peer routing protocol, peer discovery protocol, and types of network attacks are all areas of future work.

ACKNOWLEDGMENTS

This research was supported in part by DARPA Contract N66001-96-C-8530, NSF Grant CCR-0209144, and Department of Energy contract DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow

others to do so, for U.S. Government purposes. In addition this project was supported under Award No. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the author(s) and do not necessarily represent the official position of the U.S. Department of Homeland Security.

REFERENCES

- Andersen, D., H. Balakrishnan, F. Kaashoek, and R. Morris. 2001. Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, 131–145: ACM Press.
- Beaver, C. L., D. R. Gallup, W. D. Neumann, and M. D. Torgerson. 2002, March. Key management for scada. Technical Report SAND2001-3252, Sandia National Laboratories.
- Birman, K. P., J. Chen, K. Hopkinson, B. Thomas, J. Thorp, R. VanRenesse, and W. Vogels. 2003, October. Overcoming communications challenges in software for monitoring and controlling power systems. IEEE 1994, March. Ieee standard definition, specification, and analysis of systems used for supervisory control, data acquisition, and automatic control.
- IEEE 2003, March. Draft ieee sa technical report on substation integrated protection, control and data acquisition communication requirements. Technical Report IEEE TR 1525-2003, Substation Committee of the IEEE Power Engineering Society.
- Keromytis, A. D., V. Misra, and D. Rubenstein. 2002. Sos: secure overlay services. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, 61–72: ACM Press.
- Liljenstam, M., J. Liu, and D. M. Nicol. 2003, December. Development of an internet backbone topology for large-scale network simulations. In *Proceedings of the 2003 Winter Simulation Conference*, 694–704. New Orleans, LA.
- Lv, Q., P. Cao, E. Cohen, K. Li, and S. Shenker. 2002. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, 84–95: ACM Press.
- Mirkovic, J., and P. Reiher. 2004. A taxonomy of ddos attack and ddos defense mechanisms. *SIGCOMM Comput. Commun. Rev.* 34 (2): 39–53.
- Nicol, D., S. W. Smith, and C. Hawblitzel. 2001, December. Marianas: Survivable trust for critical infrastructure. Research Proposal NSF-01-160.
- Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Schenker. 2001. A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and proto-*

- cols for computer communications*, 161–172: ACM Press.
- Rowstron, A. I. T., and P. Druschel. 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, 329–350: Springer-Verlag.
- Sen, S., and J. Wang. 2004. Analyzing peer-to-peer traffic across large networks. *IEEE/ACM Trans. Netw.* 12 (2): 219–232.
- Smith, S. W. 2002. Outbound authentication for programmable secure coprocessors. In *7th European Symposium on Research in Computer Security*, 72–89: Springer-Verlag.
- SSFNet 1999. www.ssfnet.org
- Stoica, I., R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. 2003. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans. Netw.* 11 (1): 17–32.
- Wang, J., and A. A. Chien. 2002, December. An analysis of using overlay networks to resist distributed denial-of-service attacks. Technical report, University of California San Diego.
- Zhao, B., J. Kubiawicz, and A. Joseph. 2001, April. Tapestry: An infrastructure for fault-tolerant widearea location and routing. Technical Report UCB/CSB-01-1141, Computer Science Division, U.C. Berkeley, California, U.S.A.
- ence. From 1996-2003 he was Professor of Computer Science at Dartmouth College, where he served as department chair, and at the Institute for Security Technology Studies served as Associate Director for Research and Development, and finally as Acting Director. From 1987-1996 he was on the faculty of the Computer Science department at the College of William and Mary; 1985-1987 he was a staff scientist at the Institute for Computer Applications in Science and Engineering. He has a B.A. in mathematics from Carleton College (1979), an M.S. (1983) and Ph.D. (1985) in computer science from the University of Virginia. His research interests are in high performance computing, performance analysis, simulation and modeling, and network security. He is a Fellow of the IEEE.
- (Mirkovic and Reiher 2004) Nicol, Smith, and Hawblitzel (2001), (Smith 2002) (Beaver et al. 2002).

AUTHOR BIOGRAPHIES

JEFFREY J. FARRIS is a masters student in the Electrical and Computer Engineering Department at University of Illinois, Urbana-Champaign. He received a B.S. in Electrical Engineering from University of Illinois, Urbana-Champaign in 1999. He worked in the software consulting industry primarily focusing on software architecture and application performance testing from 2000-2004 prior to returning to school to pursue a graduate degree. His interests include network security, peer-to-peer networking, image processing, steganography, and digital tamper detection.

DAVID M. NICOL is Professor of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign, and member of the Coordinated Sciences Laboratory. He is co-author of the textbook *Discrete-Event Systems Simulation*, and served as Editor-in-Chief at ACM TOMACS from 1997-2003. He is the General Chair of the 2004 Conference on Principles of Advanced and Distributed Simulation, and the General Chair of the 2006 Winter Simulation Confer-