COVER FEATURE

Denial of Service in Sensor Networks

Unless their developers take security into account at design time, sensor networks and the protocols they depend on will remain vulnerable to denial of service attacks.

Anthony D. Wood
John A. Stankovic
University of Virginia

Sensor networks hold the promise of facilitating large-scale, real-time data processing in complex environments. Their foreseeable applications will help protect and monitor critical military, environmental, safety-critical, or domestic infrastructures and resources.

In these and other vital or security-sensitive deployments, keeping the network available for its intended use is essential. The stakes are high: Denial of service attacks against such networks may permit real-world damage to the health and safety of people. Without proper security mechanisms, networks will be confined to limited, controlled environments, negating much of the promise they hold. The limited ability of individual sensor nodes to thwart failure or attack makes ensuring network availability more difficult.

To identify denial of service vulnerabilities, we analyze two effective sensor network protocols that did not initially consider security. These examples demonstrate that consideration of security at design time is the best way to ensure successful network deployment.

THEORY AND APPLICATION

Advances in miniaturization combined with an insatiable appetite for previously unrealizable information gathering have lead to the development of new kinds of networks. In many areas, static infrastructures are giving way to dynamic ad-hoc networks.

One manifestation of these trends is the development of highly application-dependent sensor networks. Developers build sensor networks to collect and analyze low-level data from an environment of interest. Accomplishing the network's goal often depends on local cooperation, aggregation, or data processing because individual nodes have limited capabilities. Physically small, nodes have tiny or irreplaceable power reserves, communicate wirelessly, and may not possess unique identifiers. Further, they must form ad hoc relationships in a dense network with little or no preexisting infrastructure.

Protocols and algorithms operating in the network must support large-scale distribution, often with only localized interactions among nodes. The network must continue operating even after significant node failure, and it must meet real-time requirements. In addition to the limitations imposed by application-dependent deadlines, because it reflects a changing environment, the data the network gathers may intrinsically be valid for only a short time.

Sensor networks may be deployed in a host of different environments, and they often figure into military scenarios. These networks may gather intelligence in battlefield conditions, track enemy troop movements, monitor a secured zone for activity, or measure damage and casualties. An airplane or artillery[1] could deploy these networks to otherwise unreachable regions.

Although military applications may be the easiest to imagine, much broader opportunities await. Sensor networks could form an impromptu communications network for rescue personnel at disaster sites, or they could themselves help locate casualties. They could monitor conditions at the rim of a volcano, along an earthquake fault, or around a critical water reservoir. Such networks could also provide always-on monitoring of home healthcare for the elderly or detect a chemical or biological threat in an airport or stadium.

Because of their low cost and low overhead, sensor networks can be deployed for civic-event monitoring, then discarded. Longer-lived networks could be periodically refreshed by new deployments, which must integrate themselves into the existing sensor network. The network must be resilient to individual node failure, since at any time nodes could be destroyed, exhaust their power, or fail due to imperfections in large-scale manufacturing processes.

0018-9162/02/$17.00 © 2002 IEEE

Table 1. Sensor network layers and DoS defenses.

Network layer	Attacks	Defenses
Physical	Jamming	Spread-spectrum, priority messages, lower duty cycle, region mapping, mode change
	Tampering	Tamper-proofing, hiding
Link	Collision	Error-correcting code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network and routing	Neglect and greed	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authorization, monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client puzzles
	Desynchronization	Authentication

For many sensor network applications, security is critical. Some face not only a harsh environment but also active and intelligent opposition, which makes the need for battlefield resistance to location, destruction, and subversion obvious. Less obvious, but just as important, are the demands in other arenas.

- *Disasters*. It may be necessary to protect the location and status of casualties from unauthorized disclosure—particularly if the disaster relates to ongoing terrorist activities instead of natural causes.
- *Public safety*. False alarms about chemical, biological, or environmental threats could cause panic or disregard for warning systems. An attack on the system's availability could precede a real attack on the protected resource.
- *Home healthcare*. Because protecting privacy is paramount, only authorized users can query or monitor the network. These networks also can form critical pieces of an accident-notification chain, thus they must be protected from failure.

Protocols and software applications should consider security in their original designs—as must sensor networks, especially with regard to resisting attacks on network availability. Attempts to add security afterwards usually prove unsuccessful.

THE DENIAL OF SERVICE THREAT

Strictly speaking, although we usually use the term to refer to an adversary's attempt to disrupt, subvert, or destroy a network, a denial of service attack is any event that diminishes or eliminates a network's capacity to perform its expected function. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interaction between these factors can cause a DoS. Although attackers commonly use the Internet to exploit software bugs when making DoS attacks, here we consider primarily protocol- or design-level vulnerabilities.

Determining if a fault or collection of faults is the result of an intentional DoS attack presents a concern of its own—one that becomes even more difficult in large-scale deployments, which may have a higher nominal failure rate of individual nodes.

An intrusion-detection system monitors a host or network for suspicious activity patterns such as those that match some preprogrammed or possibly learned rules about what constitutes normal or abnormal behavior.[2] Although we do not deal with IDS strategies here, some of the research problems overlap, particularly in the area of attack response.

Sensor networks destined for harsh environments should already be designed to continue functioning in the presence of faults. This robustness against physical challenges may prevent some classes of DoS attacks. Fault-tolerance may mitigate even node subversion, and efficient protocols will limit opportunities for malicious waste of resources.

Developers must, however, factor the complication of an intelligent, determined adversary into the design separately. For example, they can design sensors to withstand the effects of normal thermal cycles in a desert environment or to cope with transient irregularities in radio propagation. However, this will not be sufficient to thwart an attacker with physical access to the node, who can move or heat and cool the device at will.

An adversary may possess a broad range of attack capabilities. A physically damaged or manipulated node used for attack may be less powerful than a normally functioning node. Subverted nodes that interact with the network only through software are as powerful as other nodes.

Some network deployments are vulnerable to immensely more powerful adversaries. As a prelude to military attack, a wireless sensor network can be aerially deployed in enemy territory. If the enemy already has a wired network and power grid available and can interact with the newly deployed sensor network, it can apply powerful back-end resources to subvert or disrupt the new network. This kind of asymmetry in capabilities presents a daunting security challenge.

A layered network architecture can improve robustness by circumscribing layer interactions and interfaces. A clean division of layers may be sacrificed for performance in sensor networks, however, reducing robustness. Each layer is vulnerable to different DoS attacks, and has different options available for its defense. Some attacks crosscut multiple layers or exploit interactions between them.

Table 1 lists the layers of a typical sensor network and describes each layer's vulnerabilities and defenses.

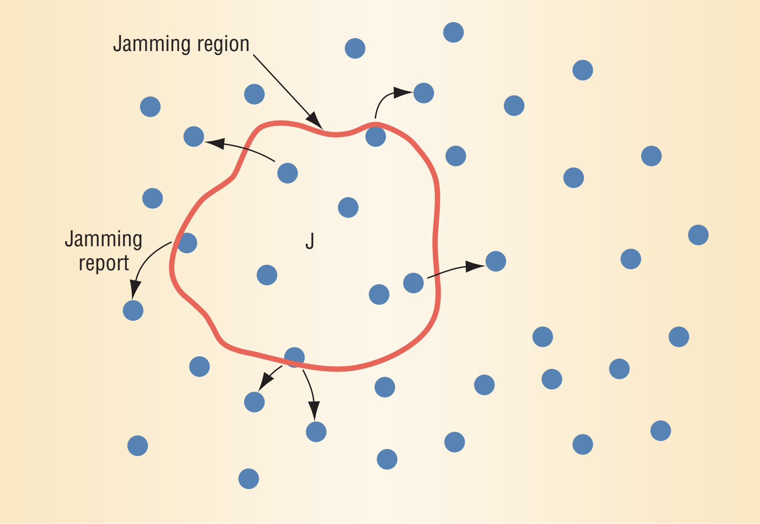

Figure 1. Defense against a jamming attack, phase one. Nodes along the edge of a jammed region report the attack to their neighbors.

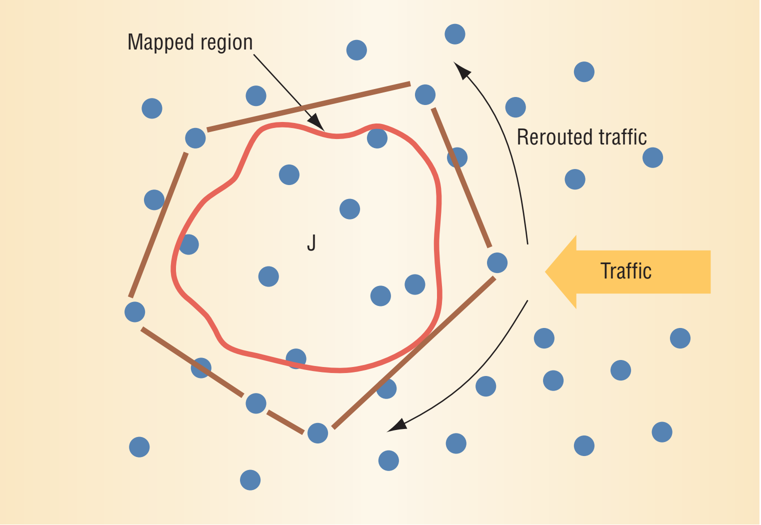

Figure 2. Defense against a jamming attack, phase two. Neighboring nodes collaborate to map the jamming reports, then reroute traffic around the jammed region.

PHYSICAL LAYER

Nodes in a sensor network use wireless communication because the network's ad hoc, large-scale deployment makes anything else impractical. Base stations or uplink nodes can use wired or satellite communication, but limitations on their mobility and energy make them more scarce.

Jamming

A well-known attack on wireless communication, jamming interferes with the radio frequencies a network's nodes are using. An adversary can disrupt the entire network with k randomly distributed jamming nodes, putting N nodes out of service, where k is much less than N. For single-frequency networks, this attack is simple and effective.

A node can easily distinguish jamming from the failure of its neighbors by determining that constant energy, not lack of response, impedes communication. Both effects have similar results, however, since constant jamming prevents nodes from exchanging data or even reporting the attack to remote monitoring stations. Even sporadic jamming can be enough to cause disruption because the data the network is communicating may be valid for only a short time.

The standard defense against jamming involves various forms of *spread-spectrum* communication.[3] To attack frequency hoppers, jammers must be able either to follow the precise hopping sequence or to jam a wide section of the band. Code spreading is a jamming method that mobile-phone networks commonly use. Given that these abilities require greater design complexity and more power, low-cost, low-power sensor devices will likely be limited to single-frequency use.

If the adversary can permanently jam the entire network, effective and complete DoS results. Nodes should have a strategy for combating jamming attacks, however, such as switching to a lower duty cycle and conserving as much power as possible. Periodically, the nodes can wake up and check whether the jamming has ended. By spending energy frugally, the nodes may be able to outlive an adversary, who must continue to jam at greater expense.

When jamming is intermittent, nodes may be able to send a few high-power, high-priority messages back to a base station to report the attack, as Figure 1 shows. Nodes should cooperate to maximize the probability of successfully delivering such messages, which could mean switching to a prioritized transmission scheme that minimizes collisions. Nodes can also buffer high-priority messages indefinitely, hoping to relay them when a gap in the jamming occurs.

In a large-scale deployment, an adversary is less likely to succeed at jamming the entire network, especially if only subverted sensors perform the jamming. As Figure 2 shows, in this scenario a more appropriate response would be to call on the nodes surrounding the affected region to cooperatively map and report the DoS attack boundary to a base station.

To the surrounding nodes, however, the region appears to suffer complete or intermittent failure, and they may be unable to determine that this behavior results from a DoS attack. Fortunately, in a sufficiently dense network, some nodes will be located close to the jamming signal's edge. These nodes can detect the higher-than-normal background noise and report it to unaffected nodes outside the region, even if reception errors prevent the reporting nodes from receiving reliable acknowledgements.

Another and more costly strategy responds to jamming by using any available alternate modes of communication, such as infrared or optical,[1] if the attacker has not jammed them as well.

Tampering

An attacker can also tamper with nodes physically, and interrogate and compromise them—threats that the large-scale, ad hoc, ubiquitous nature of sensor networks exacerbates. Realistically, we cannot expect to control access to hundreds of nodes spread over several kilometers. Such networks can fall prey to true brute-force destruction, but also to more sophisticated analysis.[4] An attacker can damage or replace sensor and computation hardware or extract sensitive material such as cryptographic keys to gain unrestricted access to higher levels of communication. Node destruction may be indistinguishable from fail-silent behavior.

One defense involves tamper-proofing the node's physical package. Its success depends on

- how accurately and completely designers considered potential threats at design time;
- the resources available for design, construction, and test; and
- the attacker's cleverness and determination.

Defense against clever passersby and corrupt insiders is easier and cheaper than defense against well-funded governments.[4] When possible, the node should react to tampering in a fail-complete manner. It could, for example, erase cryptographic or program memory. Other traditional physical defenses include camouflaging or hiding nodes.

LINK LAYER

The link or media access control (MAC) layer provides channel arbitration for neighbor-to-neighbor communication. Cooperative schemes that rely on carrier sense, which let nodes detect if other nodes are transmitting, are particularly vulnerable to DoS.

Collision

Adversaries may only need to induce a collision in one octet of a transmission to disrupt an entire packet. A change in the data portion would cause a checksum mismatch at some other receiver. A corrupted ACK control message could induce costly exponential back–off in some MAC protocols. The amount of energy the attacker needs, beyond that required to listen for transmissions, is minute.

Error-correcting codes provide a flexible mechanism for tolerating variable levels of corruption in messages at any layer. However, these codes work best as counters to environmental or probabilistic errors. For a given encoding, malicious nodes can still corrupt more data than the network can correct, although at greater cost. The error-correcting codes themselves also incur additional processing and communication overhead.

The network can use collision-detection to identify these malicious collisions, which create a kind of link-layer jamming, but no completely effective defense is known. Proper transmission still requires cooperation among nodes, which are expected to avoid corruption of others' packets. A subverted node could intentionally and repeatedly deny access to the channel, expending much less energy than in full-time jamming.

An attacker can tamper with nodes physically and interrogate and compromise them—threats that the nature of sensor networks exacerbates.

Exhaustion

A naive link-layer implementation may attempt retransmission repeatedly, even when triggered by an unusually late collision, such as a collision induced near the end of the frame. This active DoS attack could culminate in the exhaustion of battery resources in nearby nodes. This attack would compromise availability even if the adversary expended no further effort. Random back–offs only decrease the probability of inadvertent collision, thus they would be ineffective at preventing this attack.

Time-division multiplexing gives each node a slot for transmission without requiring arbitration for each frame. This approach could solve the indefinite postponement problem in a back–off algorithm, but it is still susceptible to collisions.

A self-sacrificing node could exploit the interactive nature of most MAC-layer protocols in an *interrogation* attack. For example, IEEE 802.11-based MAC protocols use Request To Send, Clear To Send, and Data/Ack messages to reserve channel access and transmit data. The node could repeatedly request channel access with RTS, eliciting a CTS response from the targeted neighbor. Constant transmission would eventually exhaust the energy resources of both nodes.

One solution makes the MAC admission control *rate limiting*, so that the network can ignore excessive requests without sending expensive radio transmissions. This limit cannot drop below the expected maximum data rate the network supports, though. One design-time strategy for protection against battery-exhaustion attacks limits the extraneous responses the protocol requires. Designers usually code this capability into the system for general efficiency, but coding to handle possible attacks may require additional logic.

Unfairness

Intermittent application of these attacks or abusing a cooperative MAC-layer priority scheme can cause unfairness, a weaker form of DoS. This threat may not entirely prevent legitimate access to the channel, but it could degrade service by, for example, causing users of a real-time MAC protocol to miss their deadlines.

One defense against this threat uses *small frames* so that an individual node can capture the channel only for a short time. If the network typically transmits long messages, however, this approach increases framing overhead. Further, an adversary can defeat this defense by cheating when vying for access, such as by responding quickly while others delay randomly.

Misdirection diverts traffic from its intended destination, perhaps by fabricating malicious route advertisements.

NETWORK AND ROUTING LAYER

Higher layers may not require fully reliable transmission streams, but the network layer provides a critical service nonetheless. In a large-scale deployment, messages may traverse many hops before reaching their destination. Unfortunately, as the aggregate network cost of relaying a packet increases, so does the probability that the network will drop or misdirect the packet along the way.

The absence of pre-existing infrastructure in sensor networks means that most if not all the nodes will serve as routers for through traffic. Since every node is potentially a router, this adds new vulnerabilities to the network-layer problems experienced on the Internet. Routing protocols must be simple enough to scale up to large networks, yet robust enough to cope with failures that occur many hops away from a source.

Neglect and greed

One simple form of DoS attacks the node-as-router vulnerability by arbitrarily neglecting to route some messages. The subverted or malicious node can still participate in lower-level protocols, and may even acknowledge reception of data to the sender, but it drops messages on a random or arbitrary basis. Such a node is *neglectful*. If it also gives undue priority to its own messages, it is also *greedy*.

The dynamic source routing (DSR)[5] protocol is susceptible to this attack. Because the network caches routes, communications from a region may all use the same route to a destination. If a node along that route is greedy, it may consistently degrade or block traffic from the region to, for example, a base station.

Using multiple routing paths or sending redundant messages can reduce the effect of this attack by making it necessary for an adversary to subvert more sensor nodes. Differentiating a greedy node from a failed node can be difficult, however, so prevention is safer than relying on detection.

Homing

In most sensor networks, some nodes will have special responsibilities, such as being elected the leader of a local group for coordination. More powerful nodes might serve as cryptographic key managers, query or monitoring access points, or network uplinks. These nodes attract an adversary's interest because they provide critical services to the network.

Location-based network protocols that rely on geographic forwarding[6] expose the network to *homing* attacks. Here, a passive adversary observes traffic, learning the presence and location of critical resources. Once found, these nodes can be attacked by collaborators or mobile adversaries using other active means.

One approach to hiding important nodes provides confidentiality for both message headers and their content. If all neighbors share cryptographic keys, the network can encrypt the headers at each hop. This would prevent a passive adversary from easily learning about the source or destination of overheard messages, assuming a node has not been subverted and remains in possession of valid decryption keys.

Misdirection

A more active attack, *misdirection*, forwards messages along wrong paths, perhaps by fabricating malicious route advertisements. As a mechanism for diverting traffic away from its intended destination, this DoS attack targets the sender. By misdirecting many traffic flows in one direction, the DoS attack can target an arbitrary victim.

In one variant of misdirection, Internet *smurf* attacks,[7] the attacker forges the victim's address as the source of many broadcast Internet control-message-protocol echoes. The attacker directs all the echo replies back to the victim, flooding its network link. Among sensor network routing protocols, DSR is also vulnerable to this attack. An adversary can simply forge replies to route-discovery requests, including victims in the spoofed route.

A sensor network that relies on a hierarchical routing mechanism can use an approach similar to the *egress filtering* in Internet gateways, which can help prevent smurf attacks. By verifying the source addresses, parent routers can verify that all routed packets from below could have been originated legitimately by their children.

Black holes

Distance-vector-based protocols[8] provide another easy avenue for an even more effective DoS attack. Nodes advertise zero-cost routes to every other node, forming *routing black holes* within the network.[9] As their advertisement propagates, the network routes more traffic in their direction. In addition to disrupting message delivery, this causes intense resource contention around the malicious node as neighbors compete for limited bandwidth. These neighbors may themselves be exhausted prematurely, causing a hole or partition in the network.

Although nodes can detect a black-hole attack more easily than they can detect greed, neglect, or misdirection attacks, a black-hole attack is more disruptive. Other nodes with untainted knowledge of the network topology may suspect inconsistent advertisements.

Authorization

One defense against misdirection and black-hole attacks lets only *authorized* nodes exchange routing information. Traditional wired networks with comparatively few routers often take this approach. Routers may use a public-key encryption infrastructure to sign and verify routing updates. Sensor networks place higher demands on scalability because every node is by design a potential router.

In addition to the computational and communication overhead, designers find that key management is difficult when using public-key cryptography in sensor networks.[10] Nodes form ad hoc relationships upon deployment, they may be mobile, and additional nodes may replenish them during their lifetime. A centralized certification authority would create a single point of failure, greatly hampering the network's scalability. Lidong Zhou and Zygmunt J. Haas, among others, have proposed a mechanism for distributing the certification function among multiple servers.[11]

Nodes can still be subverted with their key material intact. This vulnerability could give an adversary the unrestricted ability to construct valid routing messages, although threshold cryptography with share updating can protect against this possibility.[12]

Monitoring

Nodes can also monitor their neighbors to ensure that they observe proper routing behavior. In one approach, the node relays a message to the next hop and then acts as a watchdog that verifies the next-hop transmission of the same packet.[13] The watchdog can detect misbehavior, subject to limitations caused by collisions, asymmetric physical connectivity, collusion, and so on. Watchdogs inform a quality-rating mechanism, also running at each node, which chooses the most reliable routes for message transmission in much the same way that certain flow-analysis procedures work.[9]

Sensor networks place higher demands on scalability because every node is by design a potential router.

Probing

A more active approach that does not require every node to participate tests network connectivity by *probing*. Networks using geography-based routing, such as Greedy Perimeter Stateless Routing,[6] can use knowledge of the physical topology to detect black holes by periodically sending probes that cross the network's diameter. Subject to transient routing errors and overload, a probing node can identify blackout regions.

A distributed probing scheme can also work.[9] To detect malicious nodes, probes must be indistinguishable from normal traffic. Otherwise, neglectful or greedy nodes could always choose to route probes correctly, escaping detection.

Redundancy

Redundancy can lessen the probability of encountering a malicious node. The network can send duplicate messages along the same path to protect against intermittent routing failure or random malice. If each message uses a different path, one of them might bypass consistently neglectful adversaries or even black holes. A more clever approach uses diversity coding[14] to send encoded messages along different paths, but with lower cost than full duplication.

TRANSPORT LAYER

This layer manages end-to-end connections. The service the layer provides can be as simple as an unreliable area-to-area anycast, or as complex and costly as a reliable sequenced-multicast bytestream. Sensor networks tend to use simple protocols to minimize the communication overhead of acknowledgements and retransmissions. Protocols that provide sequencing share many DoS vulnerabilities with the Internet transmission control protocol.

Puzzles require clients to demonstrate the commitment of their own resources to each connection. Servers distribute the puzzle, which the client must solve before receiving a connection.

Flooding

Protocols that must maintain state at either end are vulnerable to memory exhaustion through *flooding*. As in the classic TCP SYN flood,[15] an adversary sends many connection-establishment requests to the victim. Each request causes the victim to allocate resources that maintain state for that connection.

Limiting the number of connections prevents complete resource exhaustion, which would interfere with all other processes at the victim. However, this solution also prevents legitimate clients from connecting to the victim, as queues and tables fill with abandoned connections. Protocols that are connectionless, and therefore stateless, can naturally resist this type of attack somewhat, but they may not provide adequate transport-level services for the network.

One defense requires clients to demonstrate the commitment of their own resources to each connection by solving *client puzzles*.[16] The server can create and verify the puzzles easily, and storage of client-specific information is not required while clients are solving the puzzles. Servers distribute the puzzle, and clients wishing to connect must solve and present the puzzle to the server before receiving a connection. An adversary must therefore be able to commit far more computational resources per unit time to flood the server with valid connections. Under heavy load, the server could scale the puzzles to require even more work by potential clients.

This solution is most appropriate for combating adversaries that possess the same limitations as sensor nodes. It has the disadvantage of requiring more computational energy for legitimate sensor nodes, but it is less costly than wasting radio transmissions by flooding.

Desynchronization

An existing connection between two endpoints can be disrupted by *desynchronization*. In this attack, the adversary repeatedly forges messages to one or both endpoints. These messages carry sequence numbers or control flags that cause the endpoints to request retransmission of missed frames. If the adversary can maintain proper timing, it can prevent the endpoints from exchanging any useful information, causing them to waste energy in an endless synchronization-recovery protocol.

One counter to this attack *authenticates* all packets exchanged, including all control fields in the transport protocol header. Assuming that the adversary also cannot forge the authentication mechanism, the endpoints could then detect and ignore the malicious packets.

ROTOCOL VULNERABILITIES

In our work, we have examined the vulnerability of two sensor network protocols to DoS attacks. Analyzing these vulnerabilities helps show why developers should consider DoS susceptibility when designing new protocols.

Adaptive rate control

Alec Woo and David Culler describe a series of improvements to standard MAC protocols[17] that make them more applicable in sensor networks. Key mechanisms include

- random delay for transmissions,
- back off that shifts an application's periodicity phase,
- minimization of overhead in contention control mechanisms
- passive adaptation of originating and route-through admission control rates, and
- anticipatory delay for avoiding multihop hidden-node problems.

All of these features impressively improve MAC layer performance, but they still must rely on cooperation among nodes.

For efficiency's sake, Woo and Culler propose giving preference to route-through traffic in admission control by making its probabilistic multiplicative back off factor 50 percent less than the back off factor of originating traffic. This preserves the network's investment in packets that, potentially, have already traversed many hops.

However, this approach exposes a protocol vulnerability by offering an adversary the opportunity to make flooding attacks more effective. High-bandwidth packet streams that an adversary generates will receive preference during collisions that can occur at every hop along their route. Thus, the network must not only bear the malicious traffic, it also gives preference to it.

This surprising interaction shows that an adversary can exploit a reasonable approach to power conservation and efficiency.

RAP

Chenyang Lu's real-time location-based protocols (RAP) provide a real-time communication architecture integrating a query-event service API and geographic forwarding with a novel velocity monotonic scheduling (VMS) policy.[18] As Figure 3 shows, the architecture encompasses several network layers, from a prioritized MAC layer to the query-event API just below the application layer.

The VMS layer stamps packets with a desired velocity, calculated from the distance to travel and the end-to-end deadline. The originator can compute the velocity statically or the network can recompute it dynamically at each intermediate node, based on the distance left and the time taken so far. Nodes schedule packet relay by giving higher priority to higher-velocity packets.

An adversary can exploit the RAP protocol's vulnerabilities by flooding the entire network with high-velocity packets to waste bandwidth and energy. The attacker can achieve high velocities either by making the deadline short or by making the distance extraordinarily large. Packets with short deadlines will be quickly dropped, however, when they inevitably miss their deadlines. So an adversary can inject messages with geographic destinations far away, perhaps outside the entire sensor network. The error may not be discovered until the message reaches the network's edge; until then, the message wastes high-priority bandwidth in the interior.

This attack could succeed even if the network uses a location directory service that could detect these out-of-area packets. For efficiency, a message's originator typically invokes this service to locate the destination node. Once determined, the packet includes the destination so that intermediate nodes only need to make local forwarding decisions. The adversary would avoid contacting the directory service, and the malicious location would go undetected.

In dynamically recomputed velocity scheduling, each node recomputes the velocity for route-through packets. A malicious node could just drop the packet—an attack of neglect. However, the malicious node can escape the attention of monitors and watchdogs by intentionally lowering its velocity so that the packet misses its deadline at the destination—a variant of a misdirection attack. When the network finally detects the missed deadline, it may be impossible to determine where along the path a malicious delay occurred.

Statically computed velocity scheduling may be more amenable to cryptographic protection. Since only the originator computes a velocity, a message authentication code may sign or protect this value. Each node could then check whether upstream routers have tampered with the velocity.

RAP can use clock synchronization so that each node can prioritize all packets based on the time left before their deadlines and their distances. One optimization in the protocol drops packets that miss their deadlines from forwarding queues, making room for other packets. This combination could lead to a very effective DoS if the adversary can attack the clock-synchronization mechanism successfully. A desynchronized node with a sufficiently erroneous clock will always choose to drop packets instead of forwarding them. This attack also amounts to an adversary inducing the node to become a routing black hole.

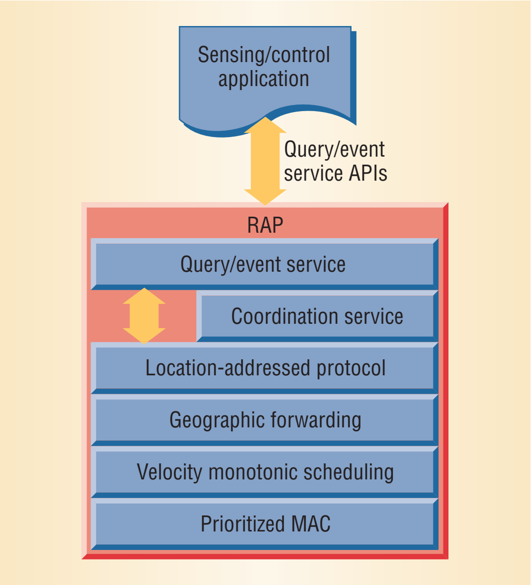

Figure 3. Real-time location-based protocols (RAP) architecture. RAP encompasses several network layers, from a prioritized media-access-control layer to the query-event API just below the application layer.

Attempts to add DoS resistance to existing protocols often focus on cryptographic-authentication mechanisms. Aside from the limited resources that make digital-signature schemes impractical, authentication in sensor networks poses serious complications. It is unclear how to establish trust, or even identity, in large-scale ad hoc deployments of potentially ID-less nodes. Adding security afterward often fails even in systems without these additional constraints.

Design-time consideration of security offers the most effective defense against attacks on availability. Applying defensive strategies can mitigate even problems that seem unsolvable, as in the case of cooperatively mapping jammed regions. Ignoring DoS vulnerabilities can lead to the unexpectedly easy compromise of network resources, as in Adaptive Rate Control's potential preference for malicious traffic.

Security is the linchpin of good sensor network design. Without sufficient protection from DoS and other attacks, sensor networks may not be deploy-able in many areas. They will only be suitable for limited, controlled environments—falling far short of their promise. ■

Acknowledgments

This work was supported in part by the DARPA NEST program under grant F33615-01-C-1905 and by the Office of Naval Research via MURI award N00014-01-1-0576.

References

1. I.F. Akyildiz et al., "Wireless Sensor Networks: A Survey," *Computer Networks*, Elsevier Science., vol. 38, no. 4, 2002, pp. 393-422.
2. A.K. Jones and R.S. Sielken, *Computer System Intrusion Detection: A Survey*, tech report, Computer Science Dept., University of Virginia, 2000.
3. R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley Computer Publishing, New York, 2001, pp. 326-331.
4. R. Anderson and M. Kuhn, "Tamper Resistance—a Cautionary Note," *Proc. 2nd Usenix Workshop on Electronic Commerce*, Usenix, Berkeley, Calif., 1996, pp. 1-11.
5. D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, vol. 353, T. Imielinski and H. Korth, eds., Kluwer Academic, Boston, 1996, pp. 153-181.
6. B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. 6th Ann. Int'l Conf. Mobile Computing and Networking* (MobiCom 2000), ACM Press, New York, 2000, pp. 243-254.
7. CERT Coordination Center, "Smurf IP Denial-of-Service Attacks," CERT Advisory CA-98:01, Jan. 1998.
8. C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. SIGCOMM*, ACM Press, New York, 1994, pp. 234-244.
9. S. Cheung and K.N. Levitt, "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection," *Proc. Workshop on New Security Paradigms*, ACM Press, New York, 1997, pp. 94-106.
10. A. Perrig et al., "SPINS: Security Protocols for Sensor Networks," *Proc. 7th Ann. Intl. Conf. Mobile Computing and Networking* (MobiCom 2001), ACM Press, New York, 2001, pp. 189-199.
11. L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks," *IEEE Network*, vol. 13, no. 6, 1999, pp. 24-30.
12. J. Kong et al., "Providing Robust and Ubiquitous Security Support for Mobile ad hoc Networks," *Proc. 9th Int'l Conf. Network Protocols*, IEEE CS Press, Los Alamitos, Calif., 2001, pp. 251-260.
13. S. Marti et al., "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proc. 6th Ann. Intl. Conf. Mobile Computing and Networking* (MobiCom 2000), ACM Press, New York, 2000, pp. 255-265.
14. E. Ayanoglu et al., "Diversity Coding for Transparent Self-Healing and Fault-Tolerant Communication Networks," *IEEE Trans. Communications*, vol. 41, no. 11, 1993, pp. 1677-1686.
15. C.L. Schuba et al., "Analysis of a Denial of Service Attack on TCP," *Proc. IEEE Symp. Security and Privacy*, IEEE Press, Piscataway, N.J., 1997, pp. 208-223.
16. T. Aura, P. Nikander, and J. Leiwo, "DOS-Resistant Authentication with Client Puzzles," *Proc. Security Protocols Workshop 2000*, Springer–Verlag, New York, 2000, pp. 170-177.
17. A. Woo and D.E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *Proc. 7th Ann. Intl. Conf. Mobile Computing and Networking* (MobiCom 2001), ACM Press, New York, 2001, pp. 221-235.
18. C. Lu et al., "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," to appear in RTAS, 2002.

***Anthony D. Wood** is a graduate student in the Department of Computer Science at the University of Virginia. His research interests include security and event tracking in ad hoc sensor networks and aspect-oriented software development. Wood received a BS in computer science from Virginia Tech. Contact him at wood@virginia.edu.*

***John A. Stankovic** is BP America Professor and chair of the Department of Computer Science at the University of Virginia. His research interests include distributed computing, real-time systems, operating systems, and ad hoc sensor networks. Stankovic received a PhD in computer science from Brown University. Contact him at stankovic@cs.virginia.edu.*