An Internet of Drones

Robert J. Hall, AT&T Labs Research
hall@research.att.com

Drones, flying devices lacking a human pilot on-board, have attracted major public attention. Retailers would love to be able to deliver goods using drones to save the costs of trucks and drivers; people want to video themselves doing all sorts of athletic and adventuresome activities; and news agencies would like to send drones to capture video of traffic and other news situations, saving the costs of helicopters and pilots.

Today, both technological and legal factors restrict what can be achieved and what can be allowed safely. For example, the U.S. Federal Aviation Administration (FAA) requires drones to operate within line-of-sight (LOS) of a pilot who is in control, and also requires drones to be registered.

In this column, I will briefly overview some of the opportunities available to improve public and commercial drone operation. I will also discuss a solution approach embodied in a research prototype, the Geocast Air Operations Framework (GAOF), I am working on in AT&T Laboratories Research. This prototype system has been implemented and tested using simulated drones; aerial field testing with real drones is being planned and will be conducted in accordance with the FAA guidelines. The underlying communications platform, the AT&T Labs Geocast System [1][2][3], has been extensively field tested in other (non-drone) domains with Earth-bound assets, such as people and cars. The goal of the work is to demonstrate a path toward an improved system for the operation of drones, with the necessary secure command and control among all legitimate stakeholders, including drone operator, FAA, law enforcement, and private property owners and citizens. While today there are drones and drone capabilities that work well with one drone operating in an area using a good communication link, there will be increased challenges when there are tens or hundreds of drones in an area.

Note that some classes of drone use are beyond the scope of this discussion:

- **Military drones.** The U.S. military has been operating drones for many years and are the acknowledged world experts in the field. However, its usage scenarios are quite different, and many of its technical approaches are out of scope for this discussion, because they have resources and authority that are unavailable (e.g. military frequency bands) or impractical (high cost drone designs and components) to use in the public/commercial setting. Instead, we seek solutions whose costs are within reason for public and commercial users and which do not require access to resources unavailable to the public.
- **Non-compliant drones.** It will always be possible for someone to build and fly drones that do not obey the protocols of our system. For example, we will not
discuss *defense against drones*, such as electromagnetic pulse (EMP) weapons, jamming, or trained birds-of-prey [4]. However, we hope to work toward a framework for safe and secure large-scale drone use, analogous to establishing traffic laws for cars.

- *Drone application layer issues.* Obviously, drones should actually do something useful once we have gone to the trouble to operate them safely. Often, this takes the form of capturing video or gathering other sensor data. This column does not address the issues involved in transferring large data sets from drone to ground or drone to cloud.

The rest of this column will give background on the communications system underlying the GAOF, the challenges of safe and scalable air operations, and how the GAOF addresses these challenges.

**Background: Multi-tier Geographic Addressing and The ALGS**

It is natural to think of using commercial communications systems, such as the commercial cellular data system (4G/LTE or GSM), to connect drones to servers in the Internet or to users operating smart phones. The difficulty here is that coverage of the cell system is not universal: what happens to the drone when it flies into a gap in coverage? Of course, the link to a remote user will not work, but there may be necessary functions, such as collision avoidance and mutual awareness, that operate between the drone and other drones which must function in order to avoid trouble.

On the other hand, it is attractive to use local wireless ad hoc networking [2] to connect drones to each other for purposes like awareness, collision avoidance, and zone declarations. (See Note 1 for a discussion of link range.) The only problem with that is it does not address the need for efficient access to remote resources, such as a remote non-LOS controller, airspace controllers, or cloud servers that may provide necessary information like restricted-fly zones.

The AT&T Labs Geocast System (ALGS) combines these two ideas into a multi-tier internetwork [1] that allows packets to flow across either tier individually, or both concurrently, depending on availability of the tiers (e.g., whether the drone is in cell coverage, or whether the drone has ad hoc WiFi capability) and on the source and destination of the packet. The ad hoc wireless tier uses WiFi devices in *ad hoc mode* (802.11 IBSS Mode); the long range tier uses a cellular data link (LTE) to connect the drone to a georouter resident in the Internet. When a drone lacks the hardware necessary for one or the other of the two tiers, the ALGS software seamlessly falls back to single-tier use. Note, however, that the architecture does provide naturally for *relaying*. So, for example, if one drone has both tiers available, it can act as a *relay*, transferring long range packets into the short range tier (or vice versa), so that a single-tier drone can receive messages that come via the relay from sources on the other tier. This is useful in many scenarios; for example, one could orbit a two-tier capable drone at higher altitude above an area of operations in a valley.
where cell coverage was nonexistent, allowing remote awareness and control of drones operating at lower altitude within the valley.

The second key concept in the ALGS is geographic addressing. In a geographic addressing (GA) network, packets are addressed to all devices in a specified subset of space. In ALGS, an address is a circle, defined by (a) the latitude and longitude of its center point, and (b) its radius in meters. In many field applications (like drone operations), one wishes to address messages to devices in a physical area, not really knowing in advance which devices are there. For example, to support an airspace awareness display, one wishes to send a query to any and all devices in the area of interest, without knowing if any are there or, if so, their identities. That, after all, is the purpose of the query. By providing this geographic addressing and routing capability at the network level, it can be made as efficient as possible. It can also be provided as a shared service to all applications needing this capability, rather than having each service implement its own GA infrastructure, which would waste network bandwidth and other resources.

For packets that traverse the ad hoc wireless tier, GA also has the advantage of being fundamentally more efficient, due to exploiting one-to-many broadcasts at the physical layer. See Note 2 for further discussion of the efficiency advantages.

In summary, the ALGS provides geographic addressing seamlessly over a multi-tier internetwork that allows both long range communications and short range, low latency communications among drones in mutual proximity.

**Challenge 1: Airspace Awareness**

To fly a drone, an operator must know at all times where it is and what else is nearby that may conflict with it. In addition, it is reasonable to assume that law enforcement or the FAA may wish to establish some control over airspace around particular areas at particular times. Such controllers will need to know locations of all drones flying in the area as well as other information about them, such as velocity, altitude, flight plan, and system health. Finally, the onboard software controlling the drone itself needs to know positions and altitudes of nearby drones and other objects in order to enable automated actions, like avoiding collisions.

*Airspace Awareness* is the general problem of querying all devices in an area for a defined set of information, usually in order to populate and update an *operating picture* of the airspace. When the same picture must be present on all the devices, such as when all drones in an area are attempting to avoid colliding with each other, we term it a *common operating picture*.

Airspace awareness is challenging for at least two reasons. First, the problem of every device telling every other device its status information sounds at first blush to require at least $n(n-1)$ messages per unit time. Such quadratic growth could swamp a wireless network if the number of drones in an area increased sufficiently. Second,
the operating picture changes quickly, because devices move in and out of the area, land and take off, run out of charge, turn on, or purposely sleep their communications for defined periods to conserve power. Thus, it is important not only to keep updating the picture regularly, it is also critical to keep track of how old the information in the picture is; knowing only the position of a nearby drone as of five minutes ago is of limited use in collision avoidance.

Figure 1 shows a simulated awareness display from a GAOF smart phone app. Green dots show positions of drones, with a small data display for each showing ID and velocity. The colored “tails” show movement history, with color coding indicating recency of the information: green indicates the past 15 seconds, yellow 15-60 seconds in the past, orange 60-240 seconds old, etc. The red rectangle is a no-fly zone (see below) that has been declared in the area using the app. The app uses the FCOP protocol [3] to obtain the data needed to populate the display. Two commanded waypoints are marked as well; onboard augmentative control software has planned a route between them that avoided crossing the no-fly zone.
Figure 1. Airspace awareness display from a GAOF smartphone app showing three (simulated) drones (green dots) and a no-fly zone (red rectangle). Multi-colored lines are position histories of the two moving drones, with color coding for recency in time, green being most recent information. The Augmentative Control module has planned a path around the red zone to replace the segment from waypoint WP₁ to WP₂.
GAOF provides a general awareness framework based upon the FCOP awareness protocol [3], which is a distributed algorithm for a common operating picture. Basically, each device wishing to monitor an area formulates a query that describes the desired information and periodically transmits this query in a geographically addressed packet to the area of interest. The devices in the area, under control of the FCOP protocol, respond back to the area surrounding the location of the device issuing the query with response messages containing their information. GAOF’s use of FCOP has a couple of advantages. First, when querying into the ad hoc wireless tier, the protocol reduces the message complexity to $O(n \lg n)$ average case per unit time, which is a major scalability advantage. Second, by responding back to an area (rather than to only the initial querying device), the algorithm further reduces message complexity by having devices listen to all responses, not just to responses to its own queries. Basically, devices know that if they have responded to a query recently to an area containing a different querying device, they need not respond again so soon. (How soon is “soon” depends upon a GAOF system parameter whose value should be guided by expected typical velocities of drones in an area.)

**Challenge 2: Non-Line-of-Sight Control**

The goal of non-line-of-sight (NLOS) control is to allow a drone controller to be physically located arbitrarily far from the drone itself. The military flies drones this way, for example. While the FAA’s rules do not permit non-line-of-sight operation today, it is useful to consider how it could be achieved if it becomes a permitted use. The primary challenges of NLOS control are latency, security, and routing. Long delays, either in moving awareness information to the controller or in moving commands from controller to drone, could lead to loss of control, collisions, or crashing into terrain. Security is critical, since one does not want an unauthorized operator to seize control of the drone via a network connection. Routing is a challenge whenever the drone is in an area with spotty or nonexistent coverage; clearly loss of contact is a significant control challenge.

GAOF addresses the latency challenge using multiple techniques. First, we assume a WayPoint-controlled Drone (WPDrone). That is, control of the drone is communicated by sending it high level commands, such as a target lat/long/altitude, a hover (hold) command, or a landing command. We do not attempt low level control, such as manipulating rotor speeds or control surface pitches. This eliminates the need for millisecond-level latency, with onboard control software dealing with that level. This does not entirely solve the problem, of course, as things can come up quickly, especially if another drone approaches. Collision avoidance is discussed below.

With the technology discussed here, we would secure the drone primarily using standard cryptographic techniques. Each command is securely signed by the controller using a secure session key shared with the onboard GAOF software, and
each command must pass the signature test on reception before it is presented for execution. If privacy is desired on this link, an additional layer of encryption will be added to obscure the waypoints.

To route the commands, GAOF uses an enhanced message addressing primitive termed a logicast. In brief, a logicast is a geographically addressed message that is further filtered by a logical predicate. That is, the address of the message is all devices in the area on which the predicate evaluates to true. In the case of NLOS control commands, the logical predicate is true iff the device’s unique ID matches that in the predicate. This allows the GA system to do the heavy lifting of finding out how to get the packet to the drone, while avoiding delivering the packet to all the drones in the area.

**Challenge 3: Augmentative Control**

There will always be restrictions on where drones can fly and when. We therefore need a way to augment the control of drones by additional commands when appropriate and authorized. These augmentations are applied to modify, cancel, or replace commands received from the controller. They should be in force even when the drone is not within cell coverage or, indeed, when it may not be within range of any communication network; consider, for example, restrictions that would keep drones out of active flight paths near major airports. Major airports may experience overloaded cell systems during prime travel hours, so drones may have only limited network connectivity in the area.

**Augmentative Control (AC)** is challenging in several ways. First, it must be secure; only parties having authorized signing keys can define airspace restrictions or other augmentative control commands. Second, drones may not always be assumed to have an Internet connection such that they can constantly keep up with a remote server that may be issuing augmentative control commands. In fact, it must be possible to issue new commands to a drone in flight, even if the flight area is not within cell coverage. For example, a law enforcement officer in the area near the drone may wish to command it to land due to an emergency situation arising, where the emergency has caused terrestrial communications systems to be unavailable. Third, there must be a clear policy hierarchy of AC commands: what if, for example, a collision avoidance AC module commands a course change into a red zone?

In GAOF, the primary augmentative control mechanism is the zone. A zone is a rectangular area defined by the tuple of \(<OriginLat, OriginLon, length, width, orientation>\). See Note 3 for how this defines a rectangle on the surface of the Earth. In GAOF, all zones extend to all altitudes, though in future, a richer 3-D representation may be developed. There are currently two types of zone:

- A red zone denotes a no-fly zone: drones must not fly into or within red zones.
- A green zone is a geofence: once inside, a drone must not leave a green zone.
Figure 1 above shows a red zone that diverted the flight plan of a drone to avoid it flying over some buildings.

Zones are communicated to drones via a distributed agreement protocol. Any device can send a zone declaration which tells devices the IDs of all zones currently in force nearby. If a device hears such a message and finds within it a zone declaration of which it was unaware, it issues a zone information request to the device, which replies with the information. Through this protocol, the drones in an area eventually come to agree on the current set of zones. A zone deletion is communicated via the same mechanism. When a drone receives a new zone definition, it checks the digital signature of the zone to ensure it is authorized. In this way, all devices can pass around zone information, but only authorized key holders can create or delete zones.

If a GAOF-style platform were to be adopted and deployed in the future on a wide scale, the zone mechanism would be expressive enough to enable these AC example scenarios:

- Airport authorities define a no-fly zone covering sensitive areas in and around the airport;
- Drone operator defines a containing geofence to ensure the drone won’t venture beyond property lines;
- Secret Service creates a temporary no-fly zone over the path of a presidential motorcade route;
- Law enforcement creates a no-fly zone around a particular drone to force it to land, subsequently deleting the zone once it has served its purpose;

The Zones AC component is just one of the AC modules deployed into the GAOF onboard software control architecture. The detailed software architecture of GAOF is beyond the scope of this column, but it is based on a blackboard architecture, where AC modules make suggestions of new flight plans to a single DronePilot module that decides among them based upon a policy definition.

The Zones AC module monitors the position and current flight plan of the drone. If it detects the drone is about to leave a green zone, it suggests a hold/hover several meters before the boundary. If it detects that the ray segment from current position to the next waypoint would cross into a red zone, it uses a graph algorithm to suggest a modification to the flight plan to reach the waypoint but skirting around the red zones. (See Figure 1 for an example of this.) If no such suggestion exists, it instead suggests a stop before crossing the red zone boundary.

AC mechanisms other than zones are useful as well and can be incorporated into GAOF’s onboard software architecture. For example, one mechanism detects when a drone is about to crash due to low-battery and commands a controlled landing instead; another implements the FAA’s drone operation ceiling, commanding the drone to descend if it gets above the limit. Many others are, of course, possible.
Challenge 4: Collision Avoidance

An important special case of augmentative control is avoidance of collisions. This is made possible by the drones collecting a common operating picture containing positions, altitudes, and velocities of objects around the drone. If a collision condition is imminent, perhaps the simplest AC action to suggest (with high priority) is a hover action. Drones incapable of hovering would have to have more complex responses, which are not yet implemented within GAOF. Collision avoidance systems in fixed-wing aircraft, such as TCAS [5], have a long and relevant history.

It is entirely possible to include GAOF components on non-drone objects. Analogous to those flashing red lights on buildings and transmitter towers, such beacons would participate in the operating picture by sending out their positions and possibly zone definitions so that drones could avoid crashing into buildings, towers, bridges, etc.

It is tempting to believe that “the sky is big, and drones are small, so why bother with collision avoidance?” One response is that buildings and other large obstacles are hazards to drones and should be addressed in some way, such as by putting active zone-declaring beacons on them as discussed above. More subtly, if a drone plans its way around the edges of no-fly zones, this tends to concentrate a large cloud of drones onto near-1-dimensional lines, which greatly increases the likelihood of interference, a phenomenon first documented in the context of commercial aviation.

Collision avoidance must work in proximity to a drone even if the area is not within cell coverage. This is a key advantage of the ALGS-based multi-tier architecture, as an ad-hoc-wifi-capable drone can communicate locally even in such conditions. ALGS and FCOP have been tested successfully in ad-hoc-only field conditions to provide a near real-time operating picture to ground units. Local messaging also has lower latency, which is an advantage in collision avoidance as well.

Conclusion

A future of safe, secure, and reliable large scale public drone operations promises to bring many benefits to society, but to achieve this various challenges must be overcome. This column has given a brief tour of four such challenges: airspace awareness, non-LOS control, augmentative control, and collision avoidance. The Geocast Air Operations Framework, based on a multi-tier geographic addressing internetwork for communications, is one option to address these challenges.

The views expressed in this column are solely those of the author and do not necessarily reflect the views of AT&T, IEEE, or any other entity.
Notes

1. We have consistently seen single-link connectivity over ad hoc WiFi at ranges of 100-125 meters in an open field at the surface of the Earth, e.g. when held by a person or mounted in a car. However, we expect link ranges for ground-to-air of hundreds of meters (400+), with imperfect connectivity even in excess of a kilometer. We expect air-to-air to exceed even these, though we of course plan to test to verify this. Electromagnetic waves propagate much better in free space than near the Earth. For slow-moving drones (tens of m/sec), these link ranges seem fine for collision avoidance.

2. For example, airspace awareness among a cloud of drones operating near each other can be accomplished in only $n \lg n$ messages [3], rather than the $n^2$ messages required by traditional IP addressed approaches. In addition to this algorithmic savings, the design of the SAGP wireless geocast protocol [2] does not rely on extra management packets, so spectrum that would normally be wasted on exchanging topology packets to establish routing tables in an IP-based wireless network is instead available to transport more information.

3. The rectangle defined by the tuple <$OriginLat$, $OriginLon$, length, width, orientation> is constructed as follows. The origin lat/lon define a reference point as one corner of the rectangle. Draw a ray segment from there, of length length, having a bearing of orientation from true North. Next, draw a ray from the origin, of length width, having a bearing of orientation+$\pi/2$ from true North. Complete the rectangle defined by the origin and the tips of the two ray segments. Aligning the rectangle such that the first ray segment lies along the positive Cartesian y-axis, the interior of the rectangle comprises the points (0..width) x (0..length).

References


Author Biography. Robert J. Hall earned the PhD and MS degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology. He earned his Bachelor’s degree in E.E.C.S at the University of California, Berkeley. Since 1991 he has been a Principal Investigator at AT&T Laboratories Research, working in the areas of automated software engineering, requirements engineering, modeling and simulation, scalable wireless network protocols, and cloud performance engineering. He is a Fellow of Automated Software Engineering and member of the Steering Committee of the IEEE/ACM International Conferences on Automated Software Engineering. He serves as Editor in Chief of Automated Software Engineering, an international journal, and is an ACM Distinguished Scientist.