Hidden Interfaces to "Ownerless" Networks

Christian Sandvig
University of Illinois at Urbana-Champaign

David Young
Champaign-Urbana Community Wireless Network

Sascha Meinrath
Champaign-Urbana Community Wireless Network

Presented to the 32nd Conference on Communication, Information, and Internet Policy
Washington, DC, USA – September 2004.

Correspondence to: Christian Sandvig
244 Lincoln Hall
University of Illinois at Urbana-Champaign
702 S. Wright St.
Urbana, IL 61801

Tel (217) 333-0141
FAX (217) 244-1598
Email: csandvig@uiuc.edu

Hidden Interfaces to "Ownerless" Networks

ABSTRACT

Experimenters are now striving to develop, apply, and refine mesh networking for wireless data. Cheap, common unlicensed 802.11 "Wi-Fi" equipment forms important testbeds used by small innovators to create the mesh. Unfortunately, the user-driven development of dynamic meshing is currently slowed or foreclosed. While the sum of all deployed Wi-Fi devices has no single owner and was not centrally deployed, it is nonetheless a network and access to it is now constrained by secrecy among manufacturers in the concentrated network card chipset industry. The authors are academics and engineers currently involved in developing mesh networks. We contacted all major manufacturers of Wi-Fi chipsets in the US (2000-04) and requested interface documentation. We had little success and found unsupportable rationales for secrecy. We contend that constellations of private part 15 equipment should be considered as an "ownerless" whole network where interfaces should be compelled using a procedure similar to Sec. 68.110. More broadly, as radios like these become increasingly defined in software, this presents a regulatory crisis: as the basis for fixing spectrum allocation rules was formerly hardware, the increasing configurability of radios may seem to create new rationales for interface secrecy. We find few benefits to interface secrecy, and argue that the benefits of user-driven innovation (like this example of dynamic mesh networking) outweigh them. Finally, the empirical ground of Wi-Fi allows us to reassess the appropriate role of regulation and its past distinctions between manufacturer and user, hardware and software, wired and wireless.

Hidden Interfaces to "Ownerless" Networks

Scholars of telecommunications policy and the historical development of information technology have long argued that control is one—or perhaps *the*—central issue in the development of technological systems of communication (e.g., Neuman, McKnight, & Solomon, 1997 or Beniger, 1986). The major debates now underway in communication law and policy about intellectual property, privacy, technological diffusion, spectrum allocation, competition, and other topics can often be reduced to debates about the location of control within the system: these are debates about who can decide what parts of the network should be doing, where these parts should be located, who can own them, and how many of them there ought to be. In some areas, our expectations about how a communication system should be controlled have been entirely transformed in the last twenty-five years. In that time the default expectations about control have moved from the slow-changing, centralized, state-sanctioned monopoly telephone infrastructure of 1980 to a dynamic, unevenly decentralized, hybrid Internet infrastructure of 2004. This paper is about how these transformations of control are affecting advanced radio technology.

While the political control of radio has certainly changed in recent decades—the introduction of auctions for spectrum licenses being the most obvious example (e.g., see McMillan, 1995)—radio itself is about to face more significant consequences of convergence in ways it has so far avoided. In the telephone network, cable television systems, and computing convergence has meant the replacement of specialized, single-purpose equipment throughout the system (like the electromechanical telephone switch, the cable decoder box, and the time-sharing terminal) with parts that are increasingly nothing more than a sophisticated gloss on top of a general-purpose computer (like the modern digital switch, the digital set-top box, and the personal computer). Digital convergence and the multiplication of processing power throughout these systems have revolutionized both the services that can be offered and how the network itself can be organized.

In radio, improvements in technology have made widespread digital radio practical and cheap. Advancements such as the smart antenna and software-defined radio (SDR) now promise to transform the components in systems of digital radio into a gloss on a general-purpose computer. In 2004, although SDR remains in its infancy, digital radiocommunication in unlicensed bands has exploded in popularity with the deployment of IEEE 802.11 ("Wi-Fi") wireless Internet and related systems. While each radio in the Wi-Fi universe may not be a general-purpose computer, at least it is a cheap consumer device that is often connected to one, usually incorporating the ability to be updated later via modifications to firmware on modifiable non-volatile memory (NVRAM). In other words, because everything increasingly looks like a computer, we have a level of flexibility in our devices that was unheard of in earlier days.[1]

One of the most significant ways that this has affected other systems of communication has been the increasing importance of users as creators of new services and applications—usually termed "user-driven" innovation. While advances in digital radio have occupied many pages of print in scholarly journals about communication, the glamorous topic has been the allocation of the electromagnetic spectrum. Radio, however, is transforming in more ways than this. In contrast to most writing, this paper examines the situation of digital radio once other

[1] Non-Volatile Random Access Memory.

repercussions of digital convergence have had their way. We will specifically address the organization of radio if user-driven innovation is increasingly allowed. We will do this through a detailed empirical analysis of one attempt to insert new features into a radio network—the attempt by a group of users to deploy unlicensed digital radios in a mesh configuration. We will show through a detailed discussion of this attempt that the major bottleneck for user driven innovation is information about the control interfaces to the hardware infrastructure that they use as a platform.

The broader conclusion of this case study is that radio has much more in common with other communication technologies than we often give it credit for. We conclude by conceptualizing data radio services as very analogous to legal and political reasoning about wired systems, with the caveat that decentralized data radio networks are "ownerless." This should shift the focus of regulatory oversight, we argue, from the owner to the equipment manufacturer.

The Role of User-Driven Innovation in Communication Systems

When we consider the development of new communication technologies, we are still prone to consider them as objects developed by "providers" or "firms" and offered to "users" or "consumers." A simple (and common) caricature of economic reasoning portrays a rational individual actor as able to accept or reject a product, but essentially powerless in the design of it. Powerful critiques of this formulation abound—the most elegant may be Hirschman's (1970) observation that economists are too quick to consider exit from a market the only recourse of dissatisfied consumers, while political scientists are too quick to consider collective action to agitate for change (in Hirschman's terms, "voice") as the only recourse of dissatisfied citizens. In fact, citizens sometimes mutely exit their political system while when products do not meet the needs of "users" they can organize and demand change just as a political movement would. More recent scholarship in the sociological study of technology has emphasized that before new technologies stabilize as commonly recognizable products, a complicated negotiation of interests shapes their development, with groups not normally thought of as "producers" having great influence (for an introduction, see Kline & Pinch 1999). In the study of communication technology specifically, studies of engineering innovation (Bardini 2000) problematize the notion of the engineer as distinct from the user and studies of adoption (Kling & Iacono 1995) emphasize the importance of collective action among users, leading to a recognition that uses for technology coevolve with the tools that fulfill them, and that forces far outside the corporate research and development lab shape the form that a communication technology will take. This renders the distinction between "user" and "producer" difficult to sustain, particularly after any detailed scrutiny of people who are usually placed in those categories (Miller, Slater, & Suchman 2004).

This scholarly work dovetails with the recognition from the study of commercial organizations that users are an important source of new technological innovation (Von Hippel 1995) and that the structure of some modern firms is changing to better capitalize on insights produced from those we formerly knew as simply receivers of products—the "consumers" (Neff & Stark 2002). This happens at a time when the open source movement (for a trade review, see Raymond 2001; Williams 2002) has demonstrated a viable alternative structure for the development of advanced computer and communication systems based on user cooperation, and open source institutions have entered into complex symbiotic relationships with more traditional firms (Weber, 2004). In sum, significant scholarship in economics, political science, business, and communication is now emphasizing the importance of user-

driven innovation and attempting to capture and explain it, and events outside the academy in the computer software world are further reinforcing the point.

Examples of User-Driven Innovation

These ideas are at home in several areas of communication policy. For instance, it is possible to reinterpret the development of every significant communication system in these terms. In the clearest example, the Internet is widely understood as an assemblage of functionality contributed by what we would otherwise term "users"—Abbate's (1999) detailed history of the Internet's invention is explicitly framed in the social constructivist tradition mentioned earlier. Applications such as e-mail were not services planned by a central network authority, but innovations created and shared among users, sometimes causing the network's central funders to worry. David (2001) emphasizes the culture of openness and cooperation that led academic users of the network to develop and freely distribute applications (such as the World Wide Web) that proved to have a much wider utility than they originally envisioned. Continuing with more recent developments such as peer-to-peer file sharing, user-contributed innovation remains important on the Internet today. Yet, in earlier communication systems users were also significant. Amateur "users" were crucial innovators in early radio (Douglas 1997); cable television was not born from networks or stations but users who created the notion of a "Community Antenna" (CATV) to improve reception. Even historical events usually framed in terms of competition between providers can be understood as animated by user dissatisfaction, if not innovation. In the US, the MCI challenge to the AT&T long-distance monopoly in the 1970s that would eventually lead to the end of AT&T's sanctioned monopoly in the American telephone network (Temin 1987) can be understood as possible because of the vocal demand for lower prices and new services from the business users of the day.

Recent theoretical developments in the understanding of communication systems are also consonant with these historical moments. For instance, traditional communication policy concepts like "universal service" that are framed as correcting market failure in telecommunications penetration can now also be understood as a policy tool to diversify the kinds of users of a system and thus create additional opportunities for user-driven innovation (Bar & Riis 2000). More fundamentally, convergence has made user-driven innovation increasingly relevant in communication systems because communication infrastructures are increasingly built from digital networks of programmable components that are much more easily modified than, say, an electromechanical switch (Bar & Sandvig 2000). The ownership of communication facilities is now entirely separable from the control and configuration of the code that runs them. While this transformation has already reached telephone switching—where more and more of the telephone network consists of programmable computers—it is about to reach radio—our topic here.

Mesh Networking as User-Driven Innovation

A 1922 guide to radio observed that "It is highly probable that many of the greatest inventions and improvements of the future will come from amateurs who, by experimenting, chance upon undreamed of things" (Verrill, 1922: iii). Today's scholarship on technological discovery also lauds the efforts of users (amateurs), but it does not grant experimentation *by chance* such an important role. Rather, user-innovators often have sound reasons for introducing new features and services.

The authors of this paper are engineers and academics who have worked extensively on the construction of a community-based wireless communication network. This network uses

unlicensed 802.11 "Wi-Fi" equipment in Urbana, Illinois. This project—called the Champaign-Urbana Community Wireless Network (CUWiN)—has been existence since 2000. CUWiN takes essentially the same consumer equipment used in homes and offices, but instead installs it on rooftops to connect neighbors in a high-speed system.

CUWiN has operated some form of unlicensed wireless data networking since 2000. The first multi-hop network connections were built in 2002 and since then the network has grown from three nodes in one cloud to a network of roughly 20 nodes in three different wireless clouds. These clouds allow direct communication between network nodes (usually in houses) but also redistribute Internet connectivity donated by cooperating partners such as the City of Urbana, the Independent Media Center, and other community organizations. Outside Urbana, CUWiN's software has been adopted by the Center for Neighborhood Technology and is the basis for a small mesh network in the North Lawndale neighborhood of Chicago. CUWiN has received donations and grant funding to increase its size in Urbana to 50 nodes by January 2005, and CNT has received grant funding from the NTIA TOP program to expand coverage through selected Chicago neighborhoods. As of this writing, other groups are planning to adopt CUWiN's dynamic mesh software, which is distributed without charge.

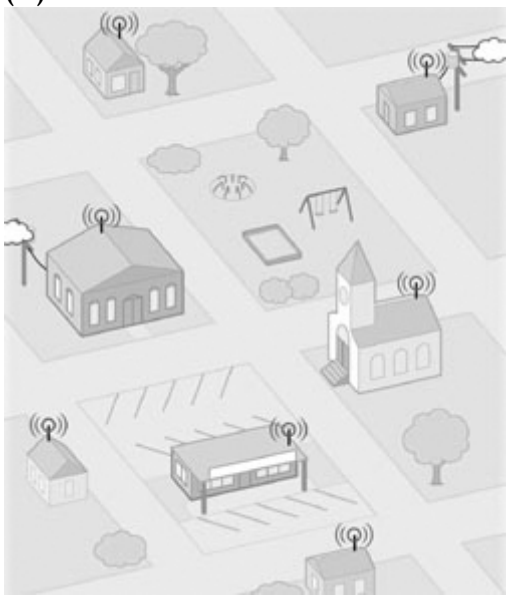
Instead of the "experiment" by chance alluded to in the 1922 quotation above, CUWiN members are motivated by frustrations with existing communication infrastructures—these include poor performance, low speed, upload restrictions on broadband Internet connections, uneven penetration, poor customer support, long waits for installation and high prices. That is, while many CUWiN members are technically skilled, they were *users* of other communication services such as cable modems and digital subscriber line (DSL) until frustration with the existing system drove them to start something new. CUWiN is a cooperatively organized not-for-profit group funded by largely by donations. In these frustrations and this organization it is similar to hundreds of "free," "open," or "community wireless" groups forming across the developed world (for a review, see Sandvig, 2004). While this might seem like an unusual idea, it is not so unusual—these efforts are direct parallels to the hundreds of community-based independent telephone companies formed by residents of small towns and rural areas who were dissatisfied with the service they were (or weren't) offered during the independent era of early telephony (see Fischer, 1992).

In Urbana, Illinois one response to this dissatisfaction could have been competition with identical technology. That is, angry locals could have tried to offer their own competitive DSL service or even local telephone service using the existing wired physical plant. More germane to this paper's topic of user-driven innovation, however, CUWiN is a different sort of project. CUWiN and some other community wireless groups are not attempting to implement the same systems as those run by traditional telecommunications companies that have let them down: instead they are attempting to build a new kind of system—a wireless dynamic mesh network—in a configuration that is unlikely to be produced by industrial research and development. To explain and evaluate this contribution, we will briefly review common configurations of unlicensed wireless data networking equipment. Figure 1 depicts four idealized conceptual examples of wireless networks.[2] The CUWiN project is attempting to move from the first (A)—the most common configuration of Wi-Fi today—to the fourth (D), a dynamic mesh network.

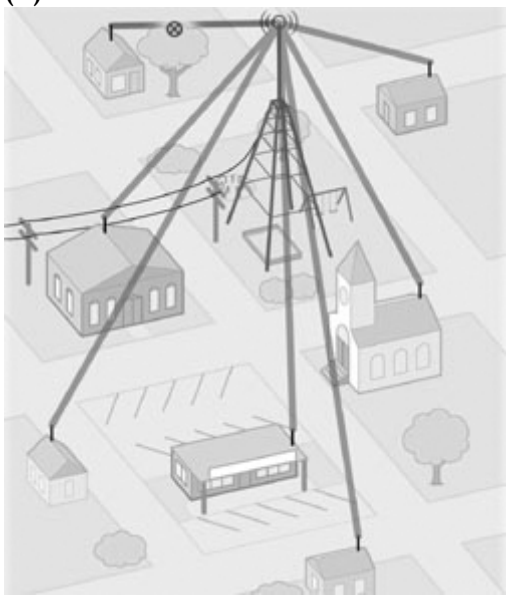
[2] An expanded version of this discussion is available (see Meinrath, n.d.).

FIGURE 1. *Conceptual Examples of Wireless Networks*

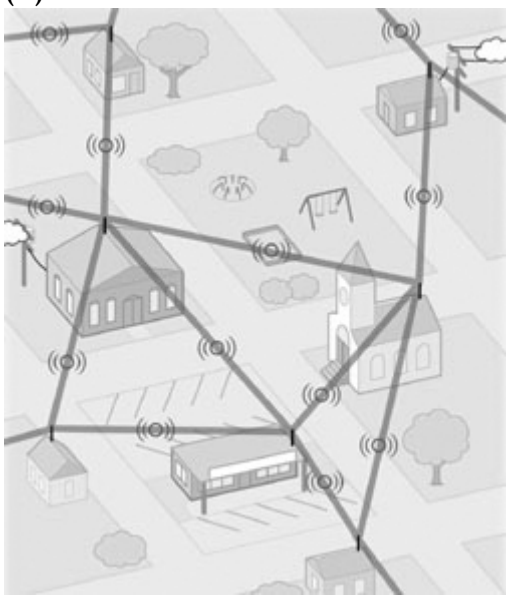
(A) Islands



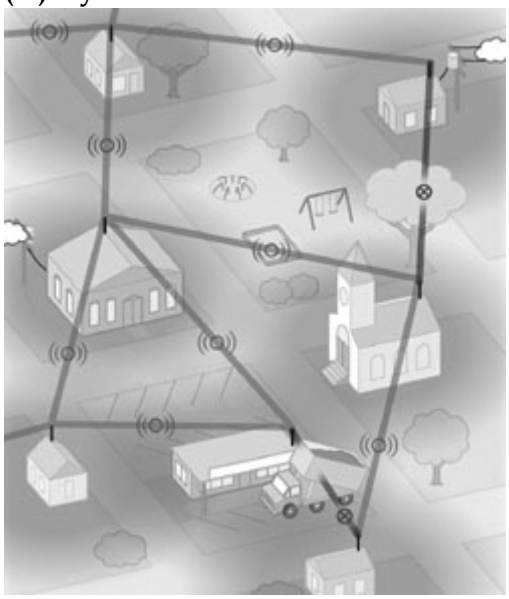
(B) Centralized



(C) Meshed



(D) Dynamic



Note. These images were conceptualized by Bryan Cribbs, Sascha Meinrath, Zachary Miller, Victor Pickard, Ben Scott, and David Young and illustrated by Darrin Drda. Full-color images are archived online at: http://www.communitywifi.org/

Islands. The first diagram, "islands" (A), represents the deployment of unlicensed wireless data networking equipment in 2000—just after the first cheap consumer product for wireless computer networking was released, the Apple AirPort. In the "island" or "hotspot" configuration, network nodes are deployed in a decentralized manner by anyone who wants to purchase one. Computers close to the access point can communicate with the access point, but they must use some other infrastructure to carry their signals farther. Alessandro Ovi memorably described this sort of organic network as, "water lilies." With water lilies of Wi-Fi, the stems usually lead to the Internet. Most Wi-Fi today is still organized as "islands," and the emphasis of these is on serving one or just a few connected computers. (Bar & Galperin term this configuration "cordless Ethernet" [2004].)

Centralized. In the second diagram, a "centralized" (B) or "hub-and-spoke" wireless system connects users with line-of-sight antennas to a centrally located broadcast tower. Clients who cannot see the hub from their building (usually because line-of-site is blocked by trees or other buildings) cannot join the network. These networks are resource-intensive; they require a tower, specialized broadcasting equipment, and maintenance of the hub – a vulnerable point of failure. Yet these relatively expensive systems are the norm for wireless communications today, as in the cellular phone. Even each single "island" mentioned above for figure (A) is really a "hub-and-spoke" network between computers and an access point. Our emphasis of "centralized" (B) as a separate category is meant to emphasize that this configuration is one way of serving multiple users in different buildings.

Meshed. In the third diagram, a "meshed" (C) network has no identifiable center. This is ideally made possible by improvements in routing where each node discovers the other nodes nearby and "explores" possible paths to any given destination. Unlike centralized networks, each new node on the network does not take extra capacity from a central hub. Instead, each new node offers another potential path—in some formulations increasing, instead of decreasing, the available capacity of the system. Mesh systems like this one are currently deployed by commercial vendors, often for municipal applications. In these systems, the location of nodes is often known—for example, nodes may be mounted on light poles throughout a city.

Dynamic. Once a cutting-edge technology, static wireless networks are now seen as difficult to plan, build, manage, and expand. Developers must map out in advance the pathways that network signals will follow to ensure reliable service. This means that if an obstacle – like a growing tree or large truck – blocks a user's connection, or if new users wish to join the network, the network may need to be manually reconfigured to enable signals to reach them. Static networks are relatively inflexible systems that are easily disrupted. The result is often an expensive, inefficient deployment with severe limitations on expansion. The fourth diagram, "dynamic," (D) emphasizes that the network can reconfigure itself to avoid both transient failures (the truck) and longer-term problems (the growing tree). Peer-to-peer file sharing networks such as Kazaa have a comparable topography.

Goals of Mesh Networking

The dynamic mesh design of the network CUWiN has built is closely related to the goals of the cooperative. As a loosely organized group, CUWiN wanted a network that anyone could join or leave at any time. CUWiN wanted the ability to efficiently share bandwidth from a small number of sources of backhaul (Internet connectivity) in order to reduce costs. Members wanted high-speed connections across town so that they could create an alternative to traditional Internet service, phone service, television service, and analog AM/FM radio (e.g.,

using Voice over Internet Protocol and multimedia streaming). In addition, the technology to accomplish this sort of network (for example, the routing) is available but new enough to require considerable technical implementation work. Weber's (2004) analysis of the open source movement notes that to attract the voluntary contribution of effort, tasks need to be well-packaged as a "challenge:" difficult enough to present a significant opportunity for creativity and learning while compartmentalized enough to provide a feeling of individual or small group authorship over a specific portion of the result. Dynamic mesh networking in the first years of this century is exactly that sort of task.

The problem of creating a dynamic mesh network lies chiefly in routing and addressing. Because there is no centralized control and a node can arrive or leave at any time, it is difficult for a single node to know what path to use to forward a communication to get it closer to its destination, or if a given destination is valid. In other words, if you were trying to pass a note to a friend in a crowded concert hall and the friend wasn't next to you in the crowd, how would you know in what direction you should pass the note? While the Internet is often described as though it were an extremely responsive dynamic system of this sort (stories are told about it "routing around" congestion, censorship, and nuclear war), in fact routes on the Internet change relatively slowly and they are often configured and tuned by technicians who painstakingly edit routing files by hand. For the dynamic mesh network to succeed there would need to be no individual tuning.

To solve this problem, "ad hoc" routing protocols have been described in the network engineering literature. These protocols spend some of the network's resources in conversation between nodes to determine routes. To return to the crowded concert hall example, this might be a conversation where you ask the people who are near you if they can see your friends, or if they have seen them lately. The available technology in this area is unsettled, leaving a lot of room for improvement and innovation. An online dictionary entry listed 65 different ad hoc protocols in September 2004, usually by referring to the published journal articles where they were introduced.[3] Computer scientists introduce these protocols either in purely theoretical terms or with a simulation of the performance of the proposed protocol (mathematically or with testbeds of computers programmed to respond as though they were actually connected by unreliable wireless connections). Only a small subset of these protocols have ever been "implemented," meaning the code has been written to allow them to be deployed on wireless devices in the field, and only a very small subset has ever been implemented by more than one party—typically an important test for a new protocol. Finally, only a small subset of the remainder has actually been implemented by more then one party and actually deployed and tested in the field—at the beginning of the CUWiN dynamic mesh project we believe the number was close to one.[4]

The Role of User-Innovators in Dynamic Mesh Networking

Implementing and testing these protocols in the field is a potential important role for any mesh networking innovator. Commercial interests are now moving to design and implement mesh protocols, but these are designed to meet the needs of paying customers whose interests do not align with the members of CUWiN. For instance, the market for mesh networking equipment has focused on fixed, evenly-spaced deployments of a known number of symmetric links in known locations (e.g., a node on every streetlight) to support a wireless

[3] See http://encyclopedia.thefreedictionary.com/Ad%20hoc%20protocol%20list
[4] Leaving aside classified military work.

"cloud" deployed by a municipality.[5] Commercial mobile mesh technology has emphasized high-mobility military and public safety applications. More crucially, commercial development to date has produced very expensive systems, while amateur research and development has produced very cheap ones. In contrast, as they were first conceived, "roofnets" like CUWiN are low mobility (unlike tanks or police cars, the member's homes do not move around continuously) but node locations are unknown and the networks must be dynamic enough to adapt to new members and changing conditions. This combination of total decentralization with low mobility and is rare for a commercial application in this area. To rephrase this, commercial and military deployments usually either know where all of their nodes are and expect them to stay still, or they don't know where any of their nodes are and they expect them to move around a lot. CUWiN expects its nodes to stay relatively still but because each is provided by a volunteer it can make no plans about how many there will be or where they are. As the user-innovation literature predicts, this technical problem is quite different than those encountered in commercial and military research & development labs.

As of 2004, three other groups have developed technology to implement dynamic mesh networks similar to CUWiN's and deployed them in the field. Two commercial companies have produced turnkey mesh products, the LocustWorld *MeshBox* and the 4G *MeshCube*—both of these are small firms (in one case, one person) closely aligned with community wireless groups. One academic project, MIT's *RoofNet*, has also deployed a rooftop mesh. For our argument, the important point is not whether or not these groups are commercial but whether or not they are *users*. Note that CUWiN, like almost all community wireless groups, is closely tied to local commercial firms. Many community wireless groups spin off commercial Wireless Internet Service Providers (WISPs). All of these groups and CUWiN are users of wireless in that they are purchasing hardware (radios and computers) they did not design and attempting to add mesh functionality to it—a feature it was not designed to satisfy.

However, from CUWiN's perspective the other dynamic mesh initiatives from users leave something to be desired. Of the many ad hoc protocols available, all of the competing solutions implement the same routing protocol: Ad hoc On-Demand Vector routing (AODV), which has known scale and performance limitations. When using AODV, the percentage of packets delivered successfully drops quickly in conditions of congestion, and throughput declines steadily as the network size increases. As AODV requires a constant conversation about the available routes, it uses power from batter-powered devices even when they are not communicating, and yet it is best suited for networks where the nodes do move around—these constant conversations about routes are redundant if the network's topography changes slowly.

In this context CUWiN discovered the Hazy-Sighted Link State (HSLS) algorithm for dynamic mesh routing in the network engineering literature and determined that it might perform well under CUWiN's needs. HSLS, developed at BBN Technologies, was described in general terms in a public article (Santivanez & Ramanathan, 2001). CUWiN set out to write a working implementation of HSLS that would transform cheap equipment into a working dynamic mesh. Initially, they did this without the knowledge of the algorithm's authors. However, as the project gained publicity and a Web presence one of the authors of the protocol contacted the group and offered to collaborate. Development has continued with the exchange of questions and answers between the community-based CUWiN and the protocol author.

CUWiN was not alone in its general approach. In Europe, Andreas Tønnesen turned his master's thesis toward the development of a better routing protocol for mesh networks. Where

[5] e.g., offerings from Tropos Networks and MeshNetworks.

CUWiN worked to implement HSLS, Tønnesen implemented Optimized Link State Routing (OLSR) on a testbed in Berlin donated by a small technology firm.

Critical Inputs for Implementing Mesh Networking

What did CUWiN need to implement HSLS and a dynamic mesh? Many analyses of technological development in telecommunications policy focus on competition as a sufficient condition for innovation, however openness is just as essential for the CUWiN effort (and user-driven innovation generally) to succeed. Widespread consumer adoption and competition in the manufacture of Wi-Fi products brought prices down to the point that unfunded groups like CUWiN could conceive of implementing their own network, yet in order to add features to Wi-Fi equipment, CUWiN engineers needed to be able to control it.

As a consumer product, Wi-Fi radios are sold as cards that can be added to a computer system, along with a software device driver that allows the networking features of the operating system to interact with the card. Drivers are often distributed in advance to large operating system producers, allowing an operating system like Microsoft Windows to "recognize" a new card that is inserted and begin using it without the extra step of installing new software to control it.

Device drivers are usually distributed as object code—this is the representation of computer software generated by a compiler. Object code, sometimes called "binary" or "executable," is compiled for a specific operating system and is not readable by a human programmer, unlike source code. In order to understand and control a Wi-Fi device, CUWiN engineers need to understand both the operation of the card and the software that controls it—they need to know what the functions of the card are and how to invoke them, then they need to write their own driver that allows them to manipulate these functions. In this, they are dependent on the documentation provided by the manufacturer and on their own skill at reverse engineering.

To implement the dynamic mesh portrayed in Figure 1, image (D), CUWiN had to find a Wi-Fi card containing a chipset that allows fine-grained control over a number of parameters such as transmit power, carrier-sense threshold, packet fragmentation, and bit rate. (By "fine-grained" we mean that these parameters can be adjusted about as rapidly as it is possible to send a new packet.) A suitable chipset also has to provide enough information about what it is doing to allow the mesh's dynamic adaptation: it must report metrics like the signal strength and the number of attempts before each packet's transmission was acknowledged.

All of these parameters are used for link adaptation, interference control, and transmit scheduling. In a dynamic mesh, different radio paths, types and levels of interference demand different fragment sizes and bit rates. Feedback on transmission attempts and received signal strength allow the estimation of the channel conditions at the receiver. Power control helps control interference between nodes on the same network ("self-interference"), while raising the carrier-sense threshold shrinks the number of nodes to contend with for access to the medium. Raising the carrier-sense threshold to a value that is effectively infinity turns off carrier-sensing altogether and it becomes possible to implement transmit scheduling via TDMA (time-division multiple access: turn-taking based on time slots), a polling protocol (a central node sends tokens to indicate turns to its neighbors), or SEEDEX (neighboring stations tell each other a pseudo-random schedule for alternating between "receiving" and "maybe transmitting" mode; they coordinate transmissions using these random schedules).[6]

[6] SEEDEX is short for "seed exchange."

For a Wi-Fi network card to be suitable the hardware must support these features, but the features also must be adequately documented by the manufacturer so that users can manipulate them.

The Equipment Bottleneck

Wi-Fi equipment had been one of the only bright spots in information technology markets after the downturn of 2000, and the rapid diffusion of Wi-Fi in the developed world has created the perception of a thriving marketplace for Wi-Fi radios. In 2000 and 2001, community wireless groups like CUWiN sought to implement dynamic mesh networking because the current state of the computer science and network engineering literature indicated that such an undertaking should be difficult but possible with readily available equipment.

At the launch of the first consumer equipment based on the IEEE 802.11b standard in 1999, only a few vendors offered chips for wireless local area network products. Market analysts estimated that about $1 billion in venture capital was then invested in wireless local area network companies over the next three years, with about one third of this going to wireless chip manufacturers (Molta, 2003). The Wi-Fi boom produced 1,649 wireless local area network products in the market as of 2004, in every possible format. However, all of these products were based on chipsets made by—at the chip market's most diverse point—only fifteen chip suppliers (AbsoluteValue Systems, 2004) and only a few chips accounted for almost all of the products. In 2001, the two market leaders (Intersil and Agere) had a combined market share of 85% (Nogee, 2002), while in 2002 Intersil alone accounted for 65% of the market (High Speed Internet Access 2002). Asian integrated circuit manufacturers entered the market in 2002 and put extreme price pressure on incumbent US and European chip suppliers (Clendedin, 2003), leading to declining profit margins and a wave of mergers and acquisitions both in wireless product manufacturers and the chipset makers that supplied them (Keenan & Mannion, 2003). In addition, dominant personal computer processor manufacturers *Intel* and *AMD* announced plans to produce most of the circuitry required to provide Wi-Fi themselves and to place it on the motherboard, drastically reducing the potential market for add-in Wi-Fi products (Kewney, 2003).

Even though a large number of Wi-Fi adapters have come to market, a closer analysis shows that there is little diversity in these product offerings at any given moment. We analyzed products offered for sale in August 2004 at the popular online retailer *Network Warehouse* by cross-referencing the manufacturer part numbers of the available products to the WLAN Adapter Chipset Directory (AbsoluteValue Systems, 2004). We checked for 802.11g adapters and found that of the 182 products for sale, most of these were the same product under different brand names. The underlying chipset (not the brand name) is the true test that differentiates how these products work, and we found that innards of these 182 actually contained only six chipsets. Any effort at innovation using purchased equipment tends to work at the forward edge of the technology that is offered for sale (otherwise, by definition, it might not be innovation). That is, new applications often arise just after some critical input or supporting infrastructure makes them possible. Due to the many Wi-Fi card features required to implement dynamic mesh networking (outlined above), just any chipset will not do. Table 1 filters the products brought to market and presents only those chipsets that contain one of the features required to implement the CUWiN design: Orthogonal Frequency Division Multiplexing (OFDM). If one wanted to try to implement a mesh in August 2004 with consumer equipment, the apparent diversity of a market with 1,600 devices is an illusion—the choice would be between only two chips.

TABLE 1. *The 802.11 Adapter Market in August 2004*

1,649	Wireless adapter products brought to market [a]
182	Products in a specialist retailer catalog [b]
128	Products carried in stock and available for purchase [b]
40	Chip manufacturers that reported an intent to enter this market [c]
18	Unique brands of 802.11g products carried in stock [b]
15	Chip manufacturers that actually brought chips to market to supply these products [a]
6	Different chipsets in 802.11g products available for sale [d]
2	802.11g chipsets in 77% of the cards for sale (Atheros, Broadcom) [d]
2	802.11g cards available that support OFDM (Atheros, Broadcom) [e]

Note. "chip" in this table refers to the combined Media Access Control (MAC) and baseband processor used in the wireless adapter, not ancillary chips.
[a] AbsoluteValue Systems, 2004
[b] from an August 2004 survey of the online retailer *Network Warehouse* by the authors.
[c] from Molta, 2003
[d] from a comparison of each part number in the survey described in note d to the WLAN Adapter Chipset Directory (AbsoluteValue Systems, 2004).
[e] OFDM is a required feature for implementing the CUWiN dynamic mesh network design.

The Openness Bottleneck

The equipment bottleneck may seem daunting, but having the necessary equipment is only one necessary condition for this kind of innovation. Even if there were 1,600 chips on the market that would support the mesh, if developers couldn't determine how to control these chips they would all be equally worthless.

As in open source software, most serious amateur software development effort in Wi-Fi occurs on one of the several important Unix-like variants of the Linux and BSD operating systems. These operating systems are the preferred choice for programmers because they are free, because they allow the widest range of customization, and because they are equipped with powerful programming tools (such as a free compiler) by default.

The first way that groups like CUWiN could implement meshing or other new features with this equipment would be to read the interface documentation made available by the hardware manufacturers and write a driver to control the device. Interface documentation does not explain how to build the Wi-Fi device in question—it is not a blueprint—and it does not simply list the features of the device in question—it is not a specification. Instead, it explains how to send signals to the card or chip in question in order to access the available features—it explains how to control the device. However (as we will explain in more detail later), although Wi-Fi card manufacturers do produce interface documentation, they do not make it available to developers like CUWiN.

Frustrated in the most obvious route to implement new features, development groups frozen out from interface documentation have to resort to reverse engineering. Ten years ago, users of open source operating systems like Linux would not have been seen as an important market for consumer-grade network card manufacturers, and no drivers would be available for these platforms to be reverse engineered. However, the increasing popularity of Linux and the

actions of large corporations like IBM in partially embracing the open source and free operating system movement has now changed all that. In 2004, all of the major Wi-Fi card manufacturers release drivers for Linux.

If these drivers were of the kind typical in the open source community they would be released as readable source code and this source code would be freely available to the operating system "packagers" such as Red Hat, Debian, NetBSD, and others. The packagers would then customize the driver if necessary and release it as part of the package's normal distribution, so that a Wi-Fi card plugged into a Linux machine could be "recognized" just as it would under an operating system such as Microsoft Windows.

Instead, card manufacturers have released compiled (object code) drivers that cannot be read, and these drivers are often for only one particular flavor of free operating system—e.g., Linux (or even only *Red Hat* Linux). Where the release of some source code is necessary in order to integrate the functions of the card into the operating system, manufacturers have produced a simple shell of readable source code that calls functions in an unreadable pearl of object code. This unreadable pearl is called the Hardware Abstraction Layer (HAL) and it exists chiefly to hide the interface information for the card and obscure how the driver does its work. Some driver programmers derogatorily call the readable open source container "the shim." In the sense that a shim is a thin piece of metal used to fill a gap between two parts, the manufacturers have released the source code only for a shim used to fill the gap between the operating system and the HAL. The real work of controlling the device still hidden in the HAL, yet the companies can claim to have embraced open source and released an open source driver (the shim). Without either complete interface documentation or the readable code for a driver to work from, knowing how to control the card is exceedingly difficult.

The Openness Problem for Open Source Operating Systems

The problem of inadequate or nonexistent interface documentation plagues those who work on free operating systems. For example, a thread titled "Linux drivers for wireless network cards," on the linuxquestions.org bulletin board in February 2004 began with this posting:[7]

> ***by: fei (newbie)***
>
> My current interest is to write and implement wireless network card [drivers] for linux. I have been emailed several companies to ask to provide [interface specifications] for their products. All I got is "NO". Does anyone know where I can find a generic linux driver for the wireless network cards. Thanks!

The next posting (excerpted below) included links to what information is available online, with the comment:

> ***by: jtshaw (lq addict)***
>
> There is no generic driver because all of the chipsets work differently, and in some cases, support different features…I am currently working on a installation how-to for all the cards I can find any information on, but it is coming pretty slowly.

[7] See http://www.linuxquestions.org/.

This generated a reply from the first poster that sums up the problem of this essay.

> ***by: fei (newbie)***
>
> Thanks a lot! The links are very helpful. But it seems that only Intersil Prism chipset is supported. What about wireless [hardware] using other chipsets? Is there [a] way to get the specifications (technical details) for the other chipsets, in order to write linux drivers?

Note that the above exchange is *not* about adding features to wireless networks, it is simply about providing drivers so that users of open source operating systems can use this hardware. This problem has been addressed by other authors writing on the open source movement and the development of free operating systems (e.g., see Weber, 2004).

In the current arrangement of computer system design and manufacture, hardware production is mostly distinct from the production of both operating systems and software—different people produce these things. Without detailed information about how hardware operates, those writing the operating system have little chance of success, and vice versa. This need for information exchange is usually not seen as a great problem, however, as it is usually in the interests of hardware manufacturers to either release the necessary information to create a driver, or to release a driver themselves. For their part, operating system makers are usually eager to release the information required to integrate hardware with the features of the operating system. If a given piece of hardware is not supported by any operating systems, it cannot be used and will find no customers. If an operating system supports no hardware, the same will be true. This has been a difficult problem for the open source movement because when open source operating systems had few users there was little reason for hardware manufacturers to cooperate with their developers.[8]

For the argument at hand, however, this debate is simply the starting point, and may be a distraction. When the topic is innovation in wireless communication systems, while this situation shares some features with the coordination problems between any hardware and software, it is quite a different case. To begin the explanation of why this is so, consider the reaction of equipment manufacturers when we attempted to obtain interface documentation from them.

Attempting to Obtain Interface Documentation

Starting in 2002, CUWiN developers attempted to obtain interface documentation from nine chip manufacturers.[9] This was both a pragmatic effort and a research project: the chips selected were chosen because at the time we believed they might be useful in producing a dynamic mesh network, and we report these findings here because they are relevant to the regulation of wireless technology generally.[10] First, we attempted to obtain interface

---

[8] This coordination problem for supporters of free operating systems has not gone away, but it has improved dramatically.

[9] Since our first attempts, some of these manufacturers have merged together.

[10] Wi-Fi cards now usually contain three important chips—sometimes made by different companies. The most important is the MAC/baseband chip, chronicled in Table 1. However, it is also helpful (and for some cards, essential) for developers to have information on "ancillary" chips such as a synthesizer. Table 2 includes both baseband and ancillary chip manufacturers, and is therefore not out of a universe of 15 manufacturers. However, we believe the number of total manufacturers is not much greater.

documentation on the Web. If none was available, we contacted the manufacturer directly. The results of this effort are presented in Table 2.

We would characterize our results as largely unsuccessful. Those manufacturers with the largest market share and the most advanced features were the least responsive and provided little to no information about their products. The most open manufacturers were those who were disclosing information about older technology that had already been pushed out of the marketplace for Wi-Fi equipment. With such a small number of test cases it is impossible to determine whether this pattern is statistically significant, and with such a small number of chips on the market there is no way to obtain a larger sample. However, these results are anecdotally suggestive. What are the possible incentives behind such results?

TABLE 2. *Requests for Interface Documentation from 9 Manufacturers*

Disposition	#
3 Provided Complete Documentation…	
…on the Web.	1
…upon request.	1
…upon signing a Non-Disclosure Agreement.	1
3 Provided Very Incomplete Documentation…	
…upon request. (Declined to provide additional detail.)	2
…on the Web. (Did not respond to other requests.)	1
3 Provided No Documentation…	
…and did not provide reasons.	1
…and claimed that the FCC regulation prohibits any disclosure.	2

The Inventives for Interface Openness

An analysis of these firms as rational economic actors in the classical sense (often used in the telecommunications policy literature) does not go very far to explain these results. Most of the concern about closed interfaces in communications assumes that an infrastructure or "platform" provider is attempting to leverage control over a legal monopoly into "downstream" products, services, and applications. That is, a rational firm might try to manipulate an equipment bottleneck like the one we have described into a privileged position in a related market—these firms might want to prevent anyone else from building a meshed wireless internet service with their chips because they want to enter this market themselves. They would not want to open interfaces to anyone (firm or user) because they would view this request not as coming from a customer but from a competitor. Although such things are happening in wireless data networking,[11] that analysis does not explain what we see with our

[11] For instance, Intel's *Centrino* branding campaign allows laptop makers to capitalize on millions of dollars of *Centrino* marketing by labeling certain laptops with this word. However, a *Centrino* label can only be applied to a computer that contains the *Pentium M* laptop processor combined with an Intel internal Wi-Fi chip. There is no difference between a *Centrino* and a *Pentium M* laptop in terms of processing power (both use the *Pentium M*), and laptop makers are already providing third party internal Wi-Fi cards as a standard offering. These third party cards usually used by laptop makers offer better performance and (in some cases) lower prices than the Intel Wi-Fi

openness results in Table 2. None of the chip manufacturers we contacted have expressed any interest in entering a Internet service market or a related market that opening these interfaces to open source developers would damage (e.g., computer operating systems). Indeed, several of these companies are chip manufacturers with extensive product lines—among these products WLAN chipsets are only one offering. These companies are not focused on wireless services, or even on wireless chips—they just make chips.

Another explanation might be that these firms worry that a release of their interface documentation to anyone would eventually make its way to a competitor. This explanation does not suffice when taken with a knowledge of chip design and industry structure. The interface specification is not a blueprint. Release of the interface specification does not tell a competitor how to build the chip, only how to control it and what its features are. Listings of features for all chips are already publicly available and used to generate sales. Furthermore, interface specifications in the current system are already likely to make their way to a competitor. Currently, interface specifications are shared in the form of a "developer kit" (sometimes for a fee in the thousands or tens of thousands of dollars). These specifications are used by operating system manufacturers (like the makers of handheld computers and, to some degree, Microsoft) and software developers to provide support for a particular piece of Wi-Fi hardware. This release of information is a prerequisite to doing business, and because it tends to gather specifications for competing products in one place (e.g., most operating systems support multiple devices) in the high-turnover tech industry, it already entails a significant risk of disclosure. In addition, the most powerful economic logic at hand would tell us that the more widely a manufacturer's hardware is supported, the more hardware it will sell. In sum, manufacturers do not benefit from interface secrecy and the secrecy they have now is not very secret anyway.

In addition, another motivation for secrecy might be the desire to maintain a good working relationship with the dominant maker of operating system software for the personal computer—Microsoft. In April 2004 market research firm IDC estimated that Microsoft controls about 90% of the client operating system market.[12] We have no evidence of pressure from Microsoft on these chip manufacturers, however the fact that these chips are overwhelmingly used in cards plugged into computer running a version of Microsoft Windows suggests that if Microsoft expressed a hostility to open source software, this hostility might influence a manufacturer dependent on integration with Microsoft Windows in a decision to produce open source device drivers or to document interfaces if this is seen as aiding the development of open source.

One might argue that the cost of packaging and releasing this interface documentation information is high and the benefits are low. However, this cost is a requirement of doing business, as stated above, and the interface documents are routinely packaged into "developer kits." When CUWiN did secure interface information from vendors in Table 2, this did not take the form of a question and answer session or a custom-written summary. It simply involved forwarding documents that were already written and available for other developers.

Another explanation may be that these firms are economically rational in the classical sense but that it is impossible to discern, from the outside, all of the information that goes into their decisions. For instance, consider one of the firms in Table 2 refused to provide any

---

offering. Intel recently expanded the program to include labeling of Wi-Fi hotspot service in public places such as airports. Intel is trying to leverage dominance in the laptop central processing market into the sale of Wi-Fi chips and the provision of wireless Internet service.

[12] http://www.microsoft-watch.com/article2/0,1995,1573599,00.asp

documentation at all. After developers could not obtain interface documentation for this chip they reverse engineered its driver and found that the signals sent to control the chip (called "command words") were exactly the same as those of one of its competitor's chips, but the command words were obscured with a layer of trivial encryption. The attempt at obfuscation might suggest to some that this chip manufacturer's rationale for interface secrecy was to prevent the disclosure that its own intellectual property was stolen from a competitor.[13]

The Culture of Secrecy Among Manufacturers

Moving beyond self-interested action in the economic sense, recent scholarship about the development of the Internet (such as Abbate, 1999) has stressed that a "culture of openness" found in government-funded academic computer science programs was extremely influential in producing interfaces and protocols that were freely available (like the GIF standard before the assertion of patent rights) and publicly owned (like TCP/IP). In Abbate's research, this culture of openness was often at odds with the wishes of private firms contracted to produce parts of the ARPANET that would be come the later Internet. Abbate's analysis has been amplified by and developed by other authors. We propose that this suggests that private firms might be characterized as having the opposite climate: a "culture of secrecy." The culture of secrecy in corporations after the recent ascent of the patent portfolio and intellectual property as an important means of generating value in high tech companies means that these firms will by default refuse to release information even when it is economically rational and in their own interest to do so. This is particularly true if the request seems unusual, as one from amateur developers like CUWiN might.

Indeed, those hoping to write device drivers for open source operating systems have been threatened so often by legal action for reverse engineering and the disclosure of interfaces, that the culture of secrecy among wireless hardware manufacturers has produced a climate of fear among developers. As one example, on October 16, 2003 a posting to a developer's mailing list had the surprising title: "Please destroy RealTek 8180's wireless chipset specification document." The list moderator explained,

> Dear list subscribers, as it has been suggested by one person in this list that the document I announced this morning might have been "leaked" without the necessary authorizations of RealTek's hierarchy, I do ask you to destroy the few copies that, as I see from the logs here, you have already downloaded, until I can obtain written and formal authorization from RealTek to continue its distribution. My sincere hope is to see support for the wireless RTL8180…. For this, we need open and trusted cooperation with hardware manufacturers, not "leaked" documentation.

Again we see a parallel with open source development generally—these problems might be encountered from mouse manufacturers as often as from wireless chip makers. However, let us focus on how the CUWiN situation differs from that of an open source operating system developer hoping to include support for a hardware product. The key to this discussion is the last line of Table 2, the response from manufacturers that the disclosure of interface information would be illegal.

---

[13] As we have noted, the interface does not provide enough information to build a chip. It is possible (though unlikely) that two design teams could coincidentally choose identical command words for two chips that were designed independently. However, the rationale for encrypting the command words—hiding them—is not clear.

Claims That Open Interfaces are Illegal

The two manufacturers with the largest market share and the most advanced chips both claimed that any disclosure of interface information would be illegal under FCC rules. Specifically, they claimed that the rules about software-defined radio (SDR) prohibit the disclosure of interface specifications—presumably because any disclosure would allow the user to modify the equipment in a way that would be illegal. This claim, known among wireless developers as "the SDR excuse," is misguided in a number of ways but it is illustrative of the problems facing advanced data radio and user-driven innovation. In addition, it marks the point where innovation in communication systems departs significantly from the problems of simply supporting hardware in open source operating systems.

The licensing regime for allocating the electromagnetic spectrum has in the past depended upon fixing FCC rules in hardware. After a given band of spectrum is licensed, radio manufacturers produce circuits that are tuned to operate only at that frequency. The FCC's allocation of channels is followed by a hardware certification regime. Manufacturers submit their equipment to third-party testers who pronounce it legal to operate in the US. Unlicensed devices (like Wi-Fi cards) are certified by proving that they operate within an unlicensed "park" or band of the spectrum (for Wi-Fi, two parks are at 2.4 GHz and 5.8 GHz). A piezoelectric crystal becomes the enforcer of the spectrum allocation rules, as it is tuned to a specific frequency and cannot easily be changed.

Today's radios, however, are increasingly able to change their characteristics. Software defined radios promise dynamic tuning of the circuit. These days of SDR have not quite arrived: In the most obvious problem with "the SDR excuse," the devices in question in Table 2 are not certified by the FCC as software defined radios, and are therefore not subject to the SDR regulations. In another obvious problem with the SDR excuse, the FCC certifies hardware, not software drivers, and it is not clear how release of information about the command words used in a driver could possibly be illegal. Most likely the use of the SDR excuse is another example of the culture of secrecy among manufacturers.

However, while they are not SDRs, these radios can still be manipulated by users more than ever before. As one example, Orinoco 802.11b Wi-Fi cards sold in Europe, Japan, and the US are identical even though the spectrum allocations for unlicensed operation differ in these countries. The same card can be certified in all three because while the chip can transmit on all of the possible unlicensed frequency bands in the 2.4GHz range, a few bits stored in NVRAM tell the chip only to transmit in the bands appropriate for the country where it is sold. However, a hacker known by the online name "lincomatic," reverse engineered the use of this NVRAM and wrote a small script called "Alchemy" and posted it to the Web. The script allows a user to set any of the values in NVRAM for that card, including the serial number and frequency. For instance, users of a US card can add the 2.462-2.472 GHz range, producing a Wi-Fi card that operates above channel 11. This modification is extremely easy, and illegal.

The technology trend is currently for manufacturers to move away from hardware control to store more and more configurable settings in software and firmware. The SDR excuse reflects a deep ambivalence about this software-controlled future. The thinking might be: If the tuned and certified piezoelectric crystal is no longer the stick with which the government can enforce its spectrum allocation, what will be the bulwark against spectrum anarchy? Here we see the promise of user-driven innovation halted in its tracks because the openness required for innovation is at odds with the mechanism of spectrum regulation and the culture of manufacturers.

CUWiN continues to develop its dynamic mesh software through reverse engineering, but it is unclear that the full capabilities of dynamic mesh will be possible to realize without access to interface documentation from manufacturers. To conclude this paper, we will discuss the implications of this initiative for telecommunications policy generally.

Closed Interfaces are Foreclosing Economic Benefits

The CUWiN experience of the manufacturing and openness bottlenecks suggests that user driven innovation in advanced wireless networking is currently being foreclosed by a lack of interface documentation. While it is interesting to note that by focusing on unique chips we found the hardware market to be radically more concentrated than the way it is usually portrayed, even the very small amount of competition that exists (in the case of dynamic meshable 802.11g, competition between only two firms portrayed in Table 1), may be enough to allow user driven innovation on this infrastructure if the interfaces were open.

The products produced by CUWiN are likely to produce social benefits. While open source developers are often discussed in opposition to commercial firms, cooperative wireless network providers often spawn (or are) commercial firms. For example, CUWiN is a partnership between community organizations, local government, and a software development firm, OJC Technologies. One of CUWiN's members founded a local wireless Internet Service Provider, VoloNet. Like other community wireless initiative, CUWiN's discussions and testing of hardware and software are freely available on the Web and provide useful information to anyone interested in starting a wireless service. In addition, because the state of the art in mesh networking is so unsettled, by developing a freely available implementation of HSLS, CUWiN is providing a useful input to firms (HSLS software) and also conducting applied research that assists all technologists in the evaluation of the many currently competing mesh protocols (like AODV and OLSR).

The Limits of Incentives for Open Interfaces Described by Related Literature

The reverse engineering work done by CUWiN and similar groups sits in good standing within the scholarship on technology. Courts and legal commentators have generally supported reverse engineering in order "to gain access to the functional specifications necessary to make a compatible software program" and, although legal commentators often find the purpose of interoperability to be a more noble calling than to reverse engineer in order to develop a competing program, reverse engineering for purposes of direct competition has also been upheld (see Samuelson & Scotchmer, 2002: 1611). As noted earlier, the earlier research has almost always considered "platform" providers and their relationship to "downstream" goods. The most famous cases in this area relate to game console manufacturers and independent game cartridge producers, and printer manufacturers and sellers of "unauthorized" ink cartridges. Restating and expanding the existing research from the economics of network effects, Samuelson & Scotchmer explain that:

> The developer of a new platform might decide to publish its interfaces or make them available under open license terms - an act that makes reverse engineering unnecessary - in order to make it easy for application developers to adapt existing applications or make new applications for the platform. An important reason to open interfaces is to drive demand for the new platform. (1616)

The theoretical economic and legal literature has focused on comparing two situations: the integrated ownership of "platforms" and "applications" where the applications are or are not freely interoperable (e.g., Matutes & Regibeau, 1988). With the evidence presented above about CUWiN's attempts to secure interface documentation, we hope to have begun probe the limits of these conceptions of manufacturer behavior. While it is true that an economically rational manufacturer would open interfaces, the manufacturers here did not do so, and we found their reasoning to often be idiosyncratic and unsupportable.

The Applicability of Open Interface Rules from Wired Infrastructures

Policy discussions concerning wired infrastructures like the telephone network and cable television networks have considered the costs and merits of open interfaces at great length. These discussions are not usually considered to be at all relevant to the case study discussed here. After all, the universe of all Wi-Fi equipment is not owned by a single service provider and there is no restriction on deploying more of it. However, we find that the parallels between wired systems are quite relevant. We have demonstrated that the market for Wi-Fi chipsets is concentrated, and we argue that a useful parallel is the state of the digital telephone switch when it became, instead of the instrument for one company to provide telephone service, a platform upon with competing companies (CLECs) could provide telephone service. Both advanced digital radio today and the telephone company of ten to twenty years ago were at critical moments of dramatically increased programmability. Wi-Fi equipment forms a platform just like the digital switch of the 1990s, but Wi-Fi devices form a kind of "ownerless" network that myriad users attempt to interoperate with in various ways to provide diverse services. As a crucial piece of wireless architecture, the public interest is not served by the restriction on application development that stems from the current culture of secrecy among manufacturers. As long as this infrastructure remains concentrated in the hands of two firms (or less) and its interfaces remain secret, we will foreclose user-driven innovation benefits because innovative users are too small a market segment to be able to drive features in products using only the power of demand. Because these innovative users like CUWiN are not well financed enough to operate their own competing chip foundries, classes of innovation like the implementation of new dynamic meshing protocols will be unfulfilled.

Let us develop this parallel to wired infrastructures briefly to show what might be done to produce openness. During much of telephone history, innovation at the network's edge occurred at a snail's pace: before the Hush-A-Phone (1956) and Carterfone (1968) decisions, "unauthorized foreign attachments" (such as a plastic cup to your handset) were forbidden. Spurious claims that "network integrity" demanded a near-total ban on interconnection were overturned; each user was found to have a "right to reasonably use his telephone in ways which are privately beneficial without being publicly detrimental" (DC Cir 1956). This "harm principle of interconnection" was developed in an era of telecommunications when a sanctioned monopoly exercised near-total control over communication by telephone in the US. In that context, it was a way of limiting the reach of AT&T and the FCC into homes and businesses. Note that this decision could have been framed as a ruling about the rights of third parties (in this case, the Hush-a-Phone Corporation) to participate in the manufacture of telecommunications equipment (the famous Hush-a-Phone attachment). While this rationale could have been used, invoking competition and lower prices, instead, the decision was phrased in terms of subscriber rights.

AT&T's effort to retain control of the telephone network's interfaces produced the Direct Access Adaptor (DAA): This was an isolation transformer required to allow the connection of, say, an answering machine--adding expense and blockading functionality, just like the Hardware Abstraction Layer (HAL) implemented in object code in Wi-Fi device drivers by current manufacturers. The FCC found that requiring users to rent a DAA was illegal.

As Nall (1993) points out in an excellent review of subscriber equipment regulation, while it was initially skeptical, the Commission later came to actively pursue a policy of promoting competition in customer premise equipment—products at the network's edge. In the 1968 *Carterfone* decision the Commission first allowed customers to interconnect equipment that was not manufactured by the Bell System. Change was slow but consistently for more interconnection. By the 1976 *Mebane Home Telephone Company* ruling, the Commission linked the *Hush-a-Phone* and *Carterfone* decisions to find a "broad principle" allowing customers to interconnect equipment for private benefit (Nall 1993: 137). In an investigation a year later into the effects of this equipment interconnection in the nine years since *Carterfone*, the Commission found that the interconnection rules led to lower prices, greater choice, more payment options, new features, ease of maintenance, improved reliability, and technology that was more configurable.[14] This logic was extended in the *Computer II* decision of the early 1980s which detariffed customer equipment offered by the telephone company and aimed to completely separate regulated services from equipment. From *Computer II*, this policy is known as the "Customer Premises Equipment (CPE) Unbundling Rule," but this unbundling arose gradually and built on the earlier decisions about foreign attachments, as Nall shows. After the divestiture of AT&T in 1984, the unbundling requirements were extended to other equipment located on customer property.[15] The CPE Unbundling rule was modified only slightly in the *Computer III* ruling, and later extended to cellular telephone service.

Compelling Open Interfaces in Wireless Data Networking

The overall principle of these rulings in the wireline world was to "isolate terminal from transmission offerings"[16]—the "broad principle" *allowing* interconnection for private benefit was later transformed into rules that *compel* open interfaces in order to promote competition with incumbent local exchange carriers. In wireline telephone infrastructure, 47 CFR Part 68 governs the connection of terminal equipment to the telephone network. The Administrative Council for Terminal Attachments publishes technical criteria that must be made public in order to allow a third-party terminal to interconnect. Section 110 goes further, and compels providers to provide additional "technical information concerning interface parameters" upon request if this information is needed for interconnection and these details are not already public.

A forum like The Administrative Council for Terminal Attachments may be a suitable remedy for present problems outlined with wireless chip manufacturers. There are already calls for a common vocabulary and API across manufacturers (SDR Forum, 2003) that are

[14] The original wording was "improved…ease of making changes." (Nall 1992: 138).
[15] For instance, Network Channel Terminating Equipment (NCTE) (Nall 1992: 143).
[16] *In re* Amendment of § 64.702 of the Commission's Rules and Regs. (Second Computer Inquiry), *Final Decision,* 77 F.C.C.2d 384, para. 141 [hereinafter Computer II], *modified by Memorandum Opinion and Order,* 84 F.C.C.2d 50 (1980) [hereinafter Computer II, *MO&O*], *aff'd and clarified by Memorandum Opinion and Order on Further Reconsideration,* 88 F.C.C.2d 512 (1981), *aff'd sub nom.* Computer & Comm. Indus. Ass'n v. FCC, 693 F.2d 198 (D.C. Cir. 1982), *cert. denied,* 461 U.S. 938 (1983), *aff'd on second further recon., Memorandum Opinion and Order,* 56 Rad. Reg. 2d (P & F) 301 (1984). para 180

similar in structure to the information provided by the Council for telephones. A rule similar to section 68.110 could ensure that enough information will be disclosed to allow competition.

Toward Open Interfaces, Again

This paper has presented an empirical puzzle. The manufacturers of Wi-Fi chipsets do not open their interfaces even though there seems to be little rationale for their secrecy. Contrary to most understandings of this market, there are only a few chip manufacturers. These two bottlenecks (in openness and in the number of equipment makers) obtain at a critical moment for radio technology—a moment when the radio is programmable and configurable as never before. This new programmability presents the potential for of transformative user-driven innovation, such as the case of dynamic mesh networking presented here. This new programmability echoes the consequences of convergence in wired networks one to two decades ago. On wired networks such as the telephone network, a regulatory drive toward open interfaces has since the 1950s guaranteed more and more openness in interfaces and interconnection with positive results. On the wired telephone network, the programmable switch enabled both new feature development and new competition.

Advanced radio is ready for this transformation, but while interfaces remain secret, user-driven innovation is foreclosed. Manufacturers use the newfound configurability itself as an excuse for secrecy, implicitly arguing that disclosure of interface documentation will allow users to sidestep certification and perhaps create devices that operate illegally. However, radio chipset manufacturers already document these interfaces and share this information selectively outside their firm. There is little additional cost to compel complete openness, some precedent for wired networks, potentially some benefit to the firm, and potentially large benefits to the development of the system as a whole, and to society.

Aside from these pragmatic conclusions, this situation emphasizes the looming regulatory challenge of increasing configurability in radio, namely, that the decisions of radio regulators will no longer be enforced by hardware. Yet a call for interface secrecy stands in opposition to the last 50 years of wireline interface regulation. This begs us to reconsider the political assumptions and distinctions made between hardware and software in wireless, and to rediscover the overlooked infrastructure parallels between wired and wireless.

Sources Cited

(2002, August). "U.S. Bancorp piper Jaffray see wireless local area network reaching an inflection point." *High-Speed Internet Access 18*(8): 15-16.

Abbate, J. (1999). *Inventing the Internet.* Cambridge, Mass.: The MIT Press.

AbsoluteValue Systems, Inc. (2004, February 2). WLAN Adapter Chipset Directory. Melbourne, Florida: AbsoluteValue Systems, Inc. http://www.linux-wlan.org/docs/wlan_adapters.html.gz (Accessed 1 Sep 2004.)

Bar, F. & Riis, A. M. (2000). Tapping User-Driven Innovation: A New Rationale for Universal Service. *The Information Society 16*(1): 1-10.

Bar, F. & Sandvig, C. (2000). Rules From Truth: Post Convergence Policy for Access. Paper Presented to the 28th Annual Telecommunications Policy Research Conference. Alexandria, Virginia, USA. http://www-rcf.usc.edu/~fbar/Publications/Rules_from_Truth.pdf

Bardini, T. (2000). *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*. Stanford, CA: Stanford University Press.

Beniger, J. R. (1986). *The control revolution: Technological and economic origins of the information society*. Cambridge: Harvard University Press.

Clendedin, M. (2003, July 21). "Price battle looms as Asia firms ready low-cost WLAN ICs." *Electronic Engineering Times 1279*: 74.

David, P. (2001). The Evolving Accidental Information Super-Highway. *Oxford Review of Economic Policy 17*: 159-187.

Di Cosmo, R. (2003, October 16). "Please destroy RealTek 8180's wireless chipset specification document." *Linux-Kernel Mailing List Archives*. http://www.ussg.iu.edu/hypermail/linux/kernel/0310.2/0075.html

Douglas, S. (1997). *Inventing American Broadcasting 1899-1922*. Baltimore: Johns Hopkins University Press.

Fischer, C. (1992). *America Calling: A Social History of the Telephone to 1940.* Berkeley, Calif.: University of California Press.

Hirschman, A. O. (1970). *Exit, Voice, and Loyalty: Responses to Decline in Firms, Organizations, and States*. Cambridge, Mass.: Harvard University Press.

Keenan, R. & Mannion, P. (2003, November 10). "Pure play chip comms in jeopardy." *Electronic Engineering Times 1295*: 1.

Kewney, G. (2003, February 27). "Intel's Centrino 'a serious threat to WLAN chip makers.'" *The Register*. http://www.theregister.co.uk/2003/02/27/intels_centrino_a_serious_threat1/

Kline, R., & Pinch, T. (1999). The Social Construction of Technology. In D. MacKenzie & J. Wajcman (Eds.), *The Social Shaping of Technology* (2nd ed., pp. 113-115). Philadelphia: Open University Press.

Kling, R., & Iacono, C. S. (1995). Computerization Movements and the Mobilization of Support for Computerization. In S. L. Star (Ed.), *Ecologies of Knowledge: Work and Politics in Science and Technology* (pp. 119-153). Albany, NY: State University of New York Press.

Linuxquestions.org. (2004). "Linux drivers for wireless network cards." *Linuxquestions.org Forum Archive*. http://www.linuxquestions.org/questions/showthread.php?threadid=140875

Matutes, C. & Regibeau, P. (1988). "Mix and Match" Product Compatibility Without Network Externalities. *RAND Journal of Economics 19*: 221.

McMillan, J. (1995). Why Auction the Spectrum? *Telecommunications Policy 19*(3): 191-199.

Meinrath, S. (n.d.). "Wirelessing the World: Socio-Historical and Technical Factors Affecting the Battle over (Community) Wireless Networks." Unpublished manuscript. Urbana, IL: Institute of Communications Research, University of Illinois at Urbana-Champaign.

Miller, D., Slater, D., & Suchman, L. (2004). "Anthropology." In M. Price & H. Nissenbaum (eds.), *The Academy and the Internet,* 84-105. New York: Peter Lang.

Molta, D. (2003, October 2). "Generation W." *Network Computing14*(20): 23.

Nall, D. A. (1993). Cable Television Subscriber Equipment: Lessons from the Common Carrier Experience. *Federal Communications Law Journal 46*(1).

Neff, G. & Stark, D. (2002). "Permanently Beta: Responsive Organization in the Internet Era." In P. E. N. Howard and S. Jones (eds.), *Society Online: The Internet in Context*, 173-188. Thousand Oaks, Calif.: Sage.

Neuman, W. R., McKnight, L. W., & Solomon, R. J. (1997). *The Gordian Knot: political gridlock on the information highway*. Cambridge, Mass.: MIT Press.

Nogee, A. (2002, May 13). "Show me the money." *Electronic News 48*(20): 6.

Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* (rev. ed.) Sebastopol, Calif.: O'Reilly.

Sandvig, C. (2004). An Initial Assessment of Cooperative Action in Wi-Fi Networking. *Telecommunications Policy 28*(7/8): 579-602.

Santivanez, C. & Ramanathan, R. (2001). "Hazy-Sighted Link State (HSLS) Routing: A Scalable Link State Algorithm." BBN Technical Memorandum No. 1301. Cambridge, Mass.: BBN Technologies.

Samuelson, P. & Scotchmer, S. (2002). The Law and Economics of Reverse Engineering. *Yale Law Journal*, *111*: 1575-1663.

Verrill, A. H. (1922). *The Home Radio: How to Make and Use it.* New York: Harper & Brothers Publishers.

Von Hippel, E. (1995). *The Sources of Innovation*. New York: Oxford University Press.

Weber, S. (2004). *The Success of Open Source*. Cambridge, Mass.: Harvard University Press.

Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. Sebastopol, Calif.: O'Reilly.

AUTHOR BIOGRAPHIES

Christian Sandvig is an Assistant Professor in Speech Communication at the University of Illinois at Urbana-Champaign where he studies communication technology and public policy. He received the Ph.D. from Stanford University (2001) and a portion of his dissertation received the TPRC first prize for graduate student research (2000). In 2002 he was named a "next-generation leader in science and technology policy" in a junior faculty competition co-organized by the American Association for the Advancement of Science. Sandvig served as Markle Foundation Information Policy Fellow (2001-2002) at the Programme in Comparative Media Law and Policy, Oxford University. He remains a research associate in Socio-Legal Studies (a research centre of the Oxford Law Faculty). Sandvig's current research project on Wi-Fi is funded by a grant from the National Science Foundation.

David Young is the technical lead for the Champaign-Urbana Community Wireless Network (CUWiN). Young received his B.S. in Computer Science from Cornell University in 1999. He has been a consulting software engineer with OJC Technologies since 1999. Since 2002, he has been the principal architect and programmer for CUWiN. As an avid open-source developer for the NetBSD operating system, Young has contributed 802.11 device drivers, a link adaptation module, an extensible radio capture format, feature enhancements and bug fixes. His current work on wireless networking is supported by a grant from the Open Society Institute.

Sascha Meinrath is the coordinator for the Champaign-Urbana Community Wireless Network (CUWiN). As coordinator and a co-founder of CUWiN, he contributes expertise in research methodology, community organizing, and project management. He received his undergraduate degree in Psychology from Yale University and is now finishing both his Masters in Psychology and his Doctorate of Philosophy in the Institute of Communications Research at the University of Illinois, Urbana-Champaign. Meinrath co-founded the Urbana-Champaign Independent Media Center Foundation and currently works as a project manager for two software development companies. He is the current treasurer for the Global IndyMedia Network and was recently elected to the Board of Directors of WEFT 90.1 FM. His current work on wireless networking is supported by a grant from the Open Society Institute.

ACKNOWLEDGEMENT

This material is based on work supported by the National Science Foundation under Grant No. #0308269.