Early Review of The BitChat Protocol

Edoardo Biagioni
University of Hawai'i at Mānoa
esb@hawaii.edu

Abstract

The BitChat application and protocol were announced in July, 2025 by Jack Dorsey [1]. Corresponding iOS and Android apps are available for free (and ad-free) download on their respective stores.

Inspired by the pure peer-to-peer nature of bitcoin, by prior projects in mesh networking, and by the original Internet Relay Chat (IRC), BitChat provides a seamless secure mesh networking chat application for devices that are in direct range of Bluetooth Low Energy (BLE) communication or where messages can be forwarded by intermediate relays. When devices are no longer within the same mesh, one-to-one communication is provided among users that have already exchanged keys over an Internet-based peer-to-peer communication system called Nostr.

This paper describes the architecture and protocols of BitChat, including its use of BLE, of the Noise protocol framework, and of Nostr.

This paper also compares BitChat to other historical communication protocols based on wireless mesh communications. Specifically, this paper includes many comparisons to the AllNet project, which the author has led for over a decade.

I. Introduction

Firechat [2] was a peer-to-peer messaging app that became famous in 2014, when it was used by protesters whose Internet access was cut off by the authorities they were protesting against.

By leveraging peer-to-peer (p2p) communications among mobile devices, that is, by having each device forward to all other reachable devices whatever messages it receives, ubiquitous connectivity can be obtained as long there is a sufficient density of devices. Specifically, given any technology's maximum communication range, all-to-all communication is available as long as devices form a connected graph with no link longer than the communication range. While this is known not to scale to large sizes [3], it can be very effective within group of up to a few thousand devices.

In such a situation each device acts as both an endpoint and a router, so this is known as wireless ad-hoc networking. Very often such devices are mobile, and then it does not make sense to use the common strategy (used, for example, with IP networks) of assigning addresses based on the router to which a device is connected. Instead, and assuming that efficiency is not paramount, messages are simply flooded to all reachable devices, and some other means is used to determine which unicast message is intended for which device.

Chat applications notably require very little bandwidth, so efficiency is less important than keeping connected as many devices as possible, ultimately supporting communications among as many people as possible.

One of the weaknesses of Firechat is that it did not encrypt unicast messages, leaving its users vulnerable to eavesdropping. It may be desirable for even broadcast messages to at least be signed, to give confidence in the authenticity of the sender of any given message.

Many protocols and apps since Firechat have addressed the needs of this space, including Serval [4], Gotenna [5], Briar [6] (which uses Tor [7]), Bridgefy [8], Meshtastic [9] (which uses LoRa [10]), and many others. One distinction among different systems is whether additional hardware is required to create the mesh network. Additional hardware can provide greater communication ranges and get around the limitations of standard networking protocols, which often lack strong support for mesh networking. On the other hand, if the mesh networking is done directly between commonly-used mobile devices without needing external hardware, then many more people have access to it simply by installing the necessary software. And since the density of participating devices may then be greater, such a network may achieve higher connectivity than one using specially-designed hardware if, due to its convenience and affordability, more people and devices participate in the ad-hoc network.

Not only are wireless networking protocols generally not designed for ad-hoc p2p communication (with the outstanding exception of 802.11/WiFi in IBSS mode), many mobile operating systems also only support the most commonly used modes of such protocols. Software that wishes to use direct peer-to-peer communication must then often run with super-user (root user) privileges, which again limits the number of potential users and devices. And finally, although at the physical layer radio communications are invariably broadcast and connectionless, protocols such as Bluetooth and its descendants are only designed for connection-oriented point-to-point communications.

In addition to all these software obstacles, mobile device operating systems such as iOS, in the name of conserving energy, put apps to sleep whenever they are not in the foreground, to the point of even closing their sockets and the corresponding network connections. Closing sockets fails to save much energy, but does keep the apps from responding to incoming traffic except when using the Apple Push Notification service (APNs) [11], which requires a centralized secure server and an Internet connection.

BitChat [12] is a recently developed wireless mesh chat app sponsored by billionaire Jack Dorsey. It is designed to flood data to as many nodes as are reachable over Bluetooth Low Energy (BLE), which has a range of up to 100m under normal circumstances, and up to 1,000m if a lower bit-rate is used and the environment presents few obstacles.

This paper looks at the design decisions of BitChat at this early stage. Specifically, Section II looks at the overall architecture of the project, Section III focuses on the parts of the Noise Protocol Framework that are used by BitChat, Section IV describes BitChat's low-level use of BLE, Section V talks about the Nostr protocol that supports communication when connected to the Internet and out of BLE relay range, Section VI describes the iOS and Android apps and their strengths and limitations, and Section VII looks at project management issues that might influence the eventual success of this project.

The author of this paper has been the lead researcher and developer for the AllNet protocol and app [13] [14]. Throughout this paper we compare the design decisions of BitChat to those of AllNet. A summary of our findings is in Table I.

	BitChat	AllNet
Created	2025	2011
Downloads	10^6	10^1
Development	active	not active
Platforms	iOS, Android	Linux, Mac, iOS, Windows
Wireless	BLE	WiFi on Linux
Internet	Nostr	distributed hash table
Encryption	ChaChaPoly	AES and RSA
Authentication	SHA256	SHA512
Delay-Tolerant	no	yes
Data Retention	limited	persistent
Main Use	local broadcast	one-to-one
Other Use(s)	geohash, one-to-one	signed broadcast: local and remote

TABLE I
SIGNIFICANT FEATURES OF BITCHAT AND ALLNET.

II. BITCHAT ARCHITECTURE

As explained in the BitChat whitepaper [15], communications in BitChat are structured in four layers, shown in Figure 1:

- Application Layer, describing the messages (including BitchatMessage and DeliveryAck) used by the BitChat app.
- Session Layer, responsible for flooding messages, message types, and fragmentation. This layer includes a TimeToLive field to prevent infinite loops.
- Encryption Layer, responsible for building and using "secure channels" between this device and other devices. The functionality of this layer is discussed in Section III.
- Transport Layer, the layer that uses BLE to communicate in a peer-to-peer fashion among devices within range of each other. This layer is discussed in Section IV.

One significant difference between AllNet and BitChat is that AllNet supports delayed delivery of messages, thus implementing a Delay Tolerant Network (DTN). BitChat either delivers a message quickly or drops it, putting very little effort into delivering older messages.

Bitchat application	*application layer*
Message framing and state	*session layer*
Noise protocol framework	*encryption layer*
BLE, Wi-Fi Direct, etc	*transport layer*

Fig. 1. BitChat architecture in the BitChat whitepaper.

III. The Noise Protocol Framework and Mutual Authentication

The Noise protocol framework is “a framework for crypto protocols based on Diffie-Hellman key agreement” [16]. Noise is a family of protocols, each one identified by a sequence of letters and digits. The specific Noise protocol used by BitChat is `Noise_XX_25519_ChaChaPoly_SHA256` [15], where:

- the `XX` refers to the XX handshake pattern described below,
- the `25519` specifies that the Diffie-Hellman exchanges uses the standard elliptic curve 25519 [21]
- the `ChaChaPoly` specifies the use of the ChaCha20-Poly1305 algorithm for the Authenticated Encryption with Associated Data (AEAD) cipher [22].
- the `SHA256` specifies that cryptographic hashing uses the standard hash function SHA-256 [23].

The XX pattern of key exchange “is used for a full handshake if the parties haven’t communicated before” [17]. This is an exchange of three packets. The first packet, from Alice to Bob, carries an ephemeral key that Alice generated. The response contains a different ephemeral key that Bob generated, used to complete the Diffie-Hellman key exchange, as well as Bob’s long-term static key, encrypted with the symmetric key derived from the Diffie-Hellman exchange. The final packet, from Alice to Bob, contains Alice’s long-term static key, encrypted with the Diffie-Hellman derived key, and completes a second Diffie-Hellman exchange with Alice’s static key and Bob’s ephemeral key.

This authentication only needs to be done once for each pair of people who will use BitChat to communicate. The key is not considered valid until the individuals involved have verified the key fingerprint out-of-band.

The AllNet protocol has a similar key exchange, but a shared secret is included in the exchange to filter out any attacker who might also wish to exchange keys with Bob or Alice, so no additional verification is needed. On the other hand, AllNet uses a custom protocol, which may be less secure than a standard protocol such as used by BitChat.

IV. BitChat’s use of BLE

The fundamental nature of wireless radio transmission is connectionless broadcast. While every antenna is somewhat directional, fundamentally the radio waves spread in nearly every direction, and any sufficiently sensitive and discriminating receiver can, in the absence of noise, listen in to every transmission. The Ethernet protocol [18], which was inspired by the wireless Aloha protocol [19], carries as the first meaningful bytes of each packet a destination address so that non-recipients may stop listening to incoming packets as soon as possible.

On top of this connectionless medium, Bluetooth creates a connection mechanism so that devices may mutually authenticate to each other and transmit data, often encrypted, along these authenticated channels. The connection process often includes manual intervention, as in connecting a device to a wireless headset, but may also happen automatically.

Bluetooth Low Energy (BLE) is the newest version of Bluetooth, improving on energy efficiency and range.

In the iOS implementation of BitChat, the file `bitchat/Services/BluetoothMeshService.swift` implements all of the services needed for connection establishment and data transfer:

- peer discovery and connection: each device both advertises itself as a BitChat node and listens for other nodes’ advertisements. A node that has received another node’s advertisement then connects to it, and a version negotiation brings to two devices to agreement on which version to use. The connection is tracked over time to make sure the peer is still reachable. Should the connection go away, it is re-established if possible. Connection status changes are noted in the app’s user interface.
- message routing and relay, by forwarding each received message to all other devices that one is connected to. As each message is forwarded, its Time To Live field (TTL) is decremented. Messages with a TTL of 0 are processed locally but not forwarded. This field, similar to the analogous field in the IP header, prevents unlimited use of network resources by a packet that might be routed in a loop.

There are several layers of abstraction in BitChat message transfer. At the lowest layer (physical, radio waves) the information is broadcast to all within range. At the next layer (data link, BLE protocol), all communication is between individual peers. On the next layer (network, flooding) all received messages are forwarded to all peers reachable through the mesh network. Each such message is authenticated, and private messages are also encrypted, all with the Noise protocol. Above these layers, the session and application layers create outgoing messages and display incoming messages to the user while tracking which users are reachable and maintaining encryption and authentication state.

For comparison, the AllNet protocol has over the years made several attempts to use different versions of Bluetooth (including BLE) to transmit messages. However, in a futile attempt to avoid the complexities of Bluetooth connection management, the designers of AllNet tried to use connectionless modes of Bluetooth transmission, particularly the advertising and scanning, to transmit AllNet data directly, with no success to date. This lack of success is to be compared to the very successful use of the IBSS mode of 802.11/WiFi for ad-hoc communication among Linux implementations of AllNet.

V. Nostr

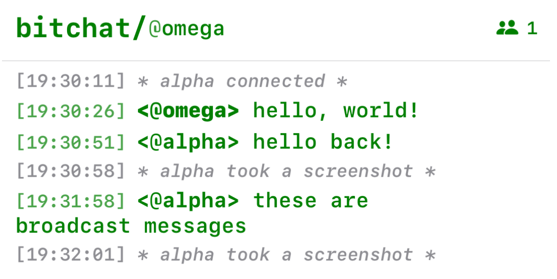
When devices are no longer part of the same wireless mesh but do have access to the Internet, BitChat still supports communication among users that have exchanged keys in the past, as well as broadcast communication in chatrooms accessible by a location-selecting feature known as a geohash [20]. As well as attempting to send messages along the BLE mesh, BitChat sends them to a distributed peer-to-peer network of relays called Nostr [24]. Receivers query relays to receive any messages addressed to their own public keys or current geohash. Although there is nothing to prevent relays or end-nodes from accessing private messages sent to others' public keys, lack of the corresponding secret key means the messages cannot be decrypted.

Nostr is claimed to be used by 18 million people [25], as a general peer-to-peer broadcast layer for the Internet for a variety of applications. It is designed to resist censorship and outside control, and so is popular with people who dislike centralized applications or centralized control.

The geohashes and resulting chat rooms are monitored by different websites, including notably `bitchat.lat`.

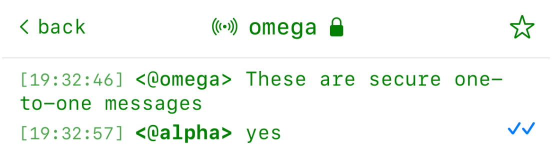
The internet communication functionality in AllNet is provided by the AllNet Distributed Hash Table (DHT). When any AllNet device is connected to the Internet, it tries to connect to other AllNet devices. If successful, the devices coordinate to divide the space of possible AllNet addresses among themselves, so that each message is stored on a limited number of devices. While in theory this may scale better than the Nostr scheme of letting devices store data on any relay they wish, in practice Nostr has grown to millions of users whereas AllNet has never had more than a handful. AllNet also lacks chat room functionality equivalent to the BitChat geohashes, although it does support signed broadcasts over the Internet.

VI. The App



```
[19:30:11] * alpha connected *
[19:30:26] <@omega> hello, world!
[19:30:51] <@alpha> hello back!
[19:30:58] * alpha took a screenshot *
[19:31:58] <@alpha> these are
broadcast messages
[19:32:01] * alpha took a screenshot *
```

Fig. 2. Broadcast messages on omega's screen.



```
[19:32:46] <@omega> These are secure one-
to-one messages
[19:32:57] <@alpha> yes
```

Fig. 3. One-to-one messages on alpha's screen.

The design of the app's UI is mostly based on textual interactions, and at time of writing, there is no way to send media such as video, audio, or pictures. Clicking on the "bitchat" label in the upper left corner brings up a description of how the app works, including that a triple tap on that same label will delete all the saved data and passwords and contact information. In general, the app seems designed for immediate interaction rather than long-term information storage and retrieval, and it is easy for recent interactions to no longer appear in the window. This is in contrast to AllNet, which, following the example of email, requires explicit user interaction to delete received messages.

Also unlike AllNet, the BitChat app reports when contacts are within range or no longer in range. Instead, AllNet has a `trace` function that can be activated manually to see which devices are reachable. While in 2011 it seemed wise to have apps be parsimonious with network usage, in 2025 saving bandwidth seems to be much less of a concern, especially since BitChat only tracks users reachable through the mesh network.

The main screen of the app shows broadcast messages sent across the BLE ad-hoc mesh network, or, if a geohash is used, recent messages within that chatroom. Messages sent from this screen are, by default, broadcast to all. In the upper right of this screen the user can see how many devices are participating in the mesh or how many users are in the chatroom. Clicking on that upper right part of the screen gives a choice of connecting to or messaging an individual. Unlike the broadcast messages, which are only signed, these individual conversations are both signed for authentication and encrypted for privacy.

The key exchange to establish a connection with an individual is seamless and nearly automatic. After this exchange, however, there is no guarantee of who the contact is. After making the individual key exchange individuals can optionally compare each other's key fingerprints and, if these fingerprints match, tell the app that the contact is verified. For security, this operation must be performed out of band, that is, with an authentication mechanism other than BitChat.

One other benefit of exchanging keys is that when the Internet is available, communication then becomes possible over the Internet using Nostr relays.

At a recent talk the author encouraged attendees present in the room to download and try out the app. The result was a brief mesh exchange among 10 different participants.

VII. Organization of the Project

The BitChat project has been very active. The first log entry for the iOS project was on July 2nd, 2025, and for the Android project on July 8th, 2025. In the three months and a week since then there have been over 800 updates to the iOS project and over 350 updates to the Android project. The last full month for which we have data, September 2025, had 85 updates to the iOS project and 51 updates to the Android project. Both projects have over 30 different authors, and there is almost no overlap between the authors of the two projects. This is very different from the AllNet project, begun as an academic project in 2011, and having 4 contributors so far, all of them enthusiastic colleagues and students of the author. The organizational skills, charisma, and perhaps abundant funds of the founder of BitChat undoubtedly have all contributed to this meteoric number of contributions, and if maintained, augur well for the future of this project.

One notable difference between the AllNet project in 2011 and the BitChat project in 2025 is that building blocks such as Noise and Nostr are available and can be leveraged, without having to be developed from scratch. Like BitChat, Nostr also has been supported by Jack Dorsey, and the use of Nostr by BitChat is likely related to this connection.

VIII. Conclusion

While still in its infancy, BitChat has been designed very carefully and thoroughly to support confidential peer-to-peer communication that is as resistant as possible to censorship and interference.

At this time there is little evidence that the general public in western democracies is interested in disconnecting from centralized services that have the ability to monitor or interfere with or communications. Nonetheless, the allure of censorship resistance and peer-to-peer decentralized communications continues to appeal to many technically inclined and privacy-oriented people.

Accurately predicting the future is challenging, but what can be said with confidence is that BitChat has many of the technical and organizational underpinnings that might make it very successful in the coming years.

IX. Acknowledgements and Disclaimer

The author is very grateful for anonymous reviewer feedback on an earlier version of this paper, and apologizes that the tight timeline for submission to this venue does not allow inclusion of all their excellent suggestions.

While the author has tried to provide an objective comparison and context for comparing BitChat to AllNet, clearly the author has been much more involved with AllNet, and some bias may have crept in.

References

[1] Jamie Crawley, "Jack Dorsey Unveils Bitchat: Offline, Encrypted Messaging Inspired by Bitcoin", Coindesk, July 8, 2025, `https://www.coindesk.com/tech/2025/07/08/jack-dorsey-unveils-bitchat-offline-encrypted-messaging-inspired--by-bitcoin`.

[2] Archie Bland, "FireChat – the messaging app that's powering the Hong Kong protests", The Guardian, 29 September 2014, `https://www.theguardian.com/world/2014/sep/29/firechat-messaging-app-powering-hong-kong-protests`

[3] Gupta and Kumar, "The Capacity of Wireless Networks", IEEE Transactions on Information Theory, V. 46, N. 2, March 2000.

[4] "Serval – Reclaim your phone", `http://servalproject.org` (unavailable, accessed through `https://archive.org`).

[5] "goTenna Uses Smart Protocols + Radio Waves To Send Messages Off-Grid", `https://gotenna.com`, retrieved 2016.

[6] "Briar – Secure P2P Messenger Releases First Version, Receives New Funding", May 9, 2018.

[7] Dingledine, Mathewson, and Syverson, "Tor: The Second-Generation Onion Router", Proceedings of the 13th USENIX Security Symposium, August 2004.

[8] Monit Khanna, "Hong Kong Protestors Are Using An App That Doesn't Need Internet, And Bypass Chinese Snooping", India Times, Septewmber 3, 2019, `https://www.indiatimes.com/technology/news/hong-kong-protestors-are-using-an-app-that-doesn-t-need-internet-and-bypass-chinese-snooping-374969.html`

[9] "Meshtastic – Open Source, decentralized mesh networking ecosystem", `https://github.com/meshtastic`

[10] LoRa Alliance, "A Technical Overview of LoRa and LoRaWan", 2015 white paper.

[11] Apple Inc., "Sending Notification Requests to APNs" `https://developer.apple.com/documentation/usernotifications/setting_up_a_remote_notification_server/sending_notification_requests_to_apns`, retrieved August 2025.

[12] Jack Dorsey, "bitchat: A decentralized peer-to-peer messaging app that works over Bluetooth mesh networks," July 2025, `https://github.com/permissionlesstech/bitchat`.

[13] Edoardo Biagioni, "AllNet Project – Own your network, Own your chat, Own your social network", 2011-present, `https://alnt.org/`.

[14] Edoardo Biagioni, "Proposal Summary", December 27, 2011, `https://alnt.org/exec.html`.

[15] Jack Dorsey, "BitChat Protocol Whitepaper", July 2025, `https://github.com/permissionlesstech/bitchat/blob/main/WHITEPAPER.md`.

[16] Trevor Perrin, "The Noise Protocol Framework", July 2018, `https://noiseprotocol.org/noise.html`.

[17] "Noise Protocol Framework", `https://en.wikipedia.org/wiki/Noise_Protocol_Framework`, retrieved August 2025.

[18] Robert Metcalfe, "Packet Communication", December 1973, MIT/LCS/TR-114.

[19] Norman Abramson, "The Aloha System – Another alternative for computer Communications", Proceedings of the 1970 Fall Joint Computer Conference, AFIPS Press.

[20] G. Niemeyer, "geohash.org is public!", `https://web.archive.org/web/20080305223755/http://blog.labix.org/#post-85`, February 2008.

[21] Daniel J. Bernstein, "Curve25519: New Diffie-Hellman Speed Records". In Yung, Dodis, Kiayias, et al. (eds.), "Public Key Cryptography - PKC 2006" Lecture Notes in Computer Science. Vol. 3958. New York: Springer. pp. 207–228. (doi:10.1007/11745853_14. ISBN 978-3-540-33851-2), 2006.

[22] Nir, Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 8439, June 2018, IRTF. `https://datatracker.ietf.org/doc/html/rfc8439`

[23] Secure Hash Standard (SHS), FIPS 180-4, latest revision August 2015, `https://doi.org/10.6028/NIST.FIPS.180-4`.

[24] Fiatjaf (pseudonymous), "nostr - Notes and Other Stuff Transmitted by Relays", Nov 2019, `https://fiatjaf.com/nostr.html`.

[25] Michael del Castillo, "Meet @Fiatjaf, The Mysterious Nostr Creator Who Has Lured 18 Million Users And $5 Million From Jack Dorsey", Forbes, May 30 2023, `https://www.forbes.com/sites/digital-assets/2023/05/30/bitcoin-social-network-nostr-creator-fiatjaf-/`.