

Table of contents

1	Bluetooth Proximity Logs	2
1.1	Proximity Logging Mechanics	2
1.2	Proximity Sample Data	3
1.3	Linking Proximity to Surveys	4
1.4	Passive Mobile Sensor Telemetry & Communication Logs	5
1.4.1	1. Raw Bluetooth Proximity Scans (Raw_Telemetry_Data/Bluetooth/)	6
1.4.2	2. SMS/Text Messaging Metadata (Raw_Telemetry_Data/SMS/)	6
1.4.3	3. Voice Call logs (Raw_Telemetry_Data/PhoneCall/)	6
1.4.4	4. Email/Mail Metadata (Raw_Telemetry_Data/Mail/)	6
1.4.5	5. Location/GPS Telemetry (Raw_Telemetry_Data/Location/)	7
1.4.6	6. Address Book/Contacts Scrapes (Raw_Telemetry_Data/Contacts/)	7
1.5	Technical Appendix: Sensor Calibration & Footnotes	7

1 Bluetooth Proximity Logs

This section describes the continuous Bluetooth physical proximity datasets located in the [Weekly Data/](#) directory.

Download Academic Calendar (PDF)

Unlike the self-reported survey waves, the Bluetooth proximity data is automatically and passively logged by smartphones carried by the students. The data is divided into 34 weekly files (`week8.csv` through `week41.csv`), representing the first academic year of the study.

1.1 Proximity Logging Mechanics

The smartphones were configured to run a continuous background service that scans for other nearby participants' Bluetooth hardware beacons. When two devices detect each other within physical range (typically ~10 meters), a **proximity episode** is recorded.

Each episode is described by 10 standardized columns:

Column	Type	Description
<code>btnamej</code>	Character	Bluetooth pseudonym of device j (e.g., <code>socs043</code>)
<code>btnamei</code>	Character	Bluetooth pseudonym of device i (e.g., <code>socs018</code>)
<code>episode</code>	Numeric	Sequential counter of the contact episode between this dyad
<code>freq</code>	Numeric	Number of Bluetooth packets exchanged during the episode
<code>avg_RSSI</code>	Numeric	Average Received Signal Strength Indicator (signal strength; closer to 0 indicates closer distance)

Column	Type	Description
sd_RSSI	Numeric	Standard deviation of RSSI during the episode
s_timestamp	Numeric	Start Unix timestamp of the episode
e_timestamp	Numeric	End Unix timestamp of the episode
dur	Numeric	Duration of the episode in seconds
waitingtime	Numeric	Elapsed time (in hours/days) since the last contact between this dyad

1.2 Proximity Sample Data

Below is a sample of the first few rows of `Weekly Data/week8.csv` to illustrate the structure of the automated logs.

Missing Weekly Data in Public Repository

Due to size limitations and IRB restrictions, the raw high-resolution Bluetooth proximity files (`Weekly Data/week*.csv`) are excluded from the public git repository. However, the compiled longitudinal networks are available, and the codebook is built with representative sample data if the raw files are not present.

```
library(readr)
library(dplyr)

# Load a sample week or use mock data if file is missing (e.g. on GitHub Actions)
file_path <- "Weekly Data/week8.csv"
if (file.exists(file_path)) {
  week_df <- read_csv(file_path, n_max = 8, show_col_types = FALSE)
} else {
  # Generate representative mock data for compilation/GitHub Pages rendering
  week_df <- tibble(
    btnamej = c("socs043", "socs018", "socs012", "socs043", "socs031", "socs018", "socs009",
    btnamei = c("socs018", "socs012", "socs043", "socs031", "socs009", "socs009", "socs018",
```

```

episode = c(1, 1, 2, 1, 1, 2, 1, 2),
freq = c(15, 8, 22, 5, 14, 11, 7, 19),
avg_RSSI = c(-74.5, -82.1, -68.4, -88.0, -79.2, -75.0, -84.3, -71.1),
sd_RSSI = c(4.2, 5.1, 3.8, 6.0, 4.5, 3.9, 5.2, 4.1),
s_timestamp = c(1315483200, 1315483600, 1315484000, 1315484600, 1315485000, 1315485500,
e_timestamp = c(1315483500, 1315483900, 1315484500, 1315484800, 1315485400, 1315485900,
dur = c(300, 300, 500, 200, 400, 400, 200, 500),
waitingtime = c(2.4, 1.1, 0.5, 3.2, 1.8, 0.9, 4.0, 1.2)
)
}

knitr::kable(week_df, caption = "Sample Bluetooth Proximity Episodes (Week 8)")

```

Table 1.1: Sample Bluetooth Proximity Episodes (Week 8)

bt- namej	bt- namei	episode	freq	avg_RSSI	sd_RSSI	s_times- tamp	e_times- tamp	dur	waiting- time
socs043	socs018	1	15	-74.5	4.2	1315483200	1315483500	300	2.4
socs018	socs012	1	8	-82.1	5.1	1315483600	1315483900	300	1.1
socs012	socs043	2	22	-68.4	3.8	1315484000	1315484500	500	0.5
socs043	socs031	1	5	-88.0	6.0	1315484600	1315484800	200	3.2
socs031	socs009	1	14	-79.2	4.5	1315485000	1315485400	400	1.8
socs018	socs009	2	11	-75.0	3.9	1315485500	1315485900	400	0.9
socs009	socs018	1	7	-84.3	5.2	1315486000	1315486200	200	4.0
socs012	socs031	2	19	-71.1	4.1	1315486500	1315487000	500	1.2

1.3 Linking Proximity to Surveys

To link these Bluetooth pseudonyms to standard student ego IDs used in demographic and network survey databases, use the mapping table:

Download Bluetooth-to-EgoID Mapping (.csv)

The mapping contains a simple two-column crosswalk:

```

library(readr)
library(dplyr)

```

```
# Load Bluetooth mapping key or use mock data if file is missing
map_path <- "Metadata and Mappings/BT_egoid_mapping.csv"
if (file.exists(map_path)) {
  map_df <- read_csv(map_path, n_max = 5, show_col_types = FALSE)
} else {
  map_df <- tibble(
    bname = c("socs001", "socs002", "socs003", "socs004", "socs005"),
    egoid = c(1001, 1002, 1003, 1004, 1005)
  )
}

knitr::kable(map_df, caption = "Bluetooth-to-EgoID Crosswalk Sample")
```

Table 1.2: Bluetooth-to-EgoID Crosswalk Sample

bname	egoid
socs003	46584
socs004	13896
socs005	26127
socs006	69065
socs007	16495

Using this table, researchers can map `bnamej` and `bnamei` to standard numeric student `egoid` values, allowing them to integrate high-resolution face-to-face physical contact networks with rich longitudinal surveys!

1.4 Passive Mobile Sensor Telemetry & Communication Logs

In addition to the cleaned, aggregated Bluetooth proximity episodes, the **NetSense Complete Raw Data** contains continuous, passive mobile sensor telemetry streams. These datasets offer an exceptionally rich, longitudinal, multi-channel view of participant behavior across several academic semesters (2011 to 2015).

To protect participant privacy under Institutional Review Board (IRB) guidelines, these raw streams are kept in a secure, Git-ignored symbolic directory (`Raw_Telemetry_Data/`). Below is the technical methodology and data schema describing each of these passive telemetry channels.

1.4.1 1. Raw Bluetooth Proximity Scans (Raw_Telemetry_Data/Bluetooth/)

Unlike the aggregated episode-level dyads, the raw Bluetooth directory contains continuous, second-by-second background scan records of the wireless environment: - **Mechanics:** Participant smartphones executed continuous background scans for Bluetooth beacons. It captured any nearby broadcasting Bluetooth hardware. - **Raw Schema:** [Scanning Ego Phone Number], [Unix Timestamp], [Friendly Device Name], [Hardware MAC Address], [RSSI Signal Strength] - **Privacy Masking Note:** The raw logs capture un-de-identified MAC addresses and “Friendly Device Names” (e.g., Ego’s MacBook Pro, Alter’s Laptop), which frequently contain participants’ real first and last names. In the clean files, these are programmatically scrubbed and mapped to standard numeric egoid values using the hardware crosswalk key.

1.4.2 2. SMS/Text Messaging Metadata (Raw_Telemetry_Data/SMS/)

Tracks the continuous exchange of text messages sent and received by participants’ devices: - **Mechanics:** Passenger smartphone operating systems passively recorded SMS transaction events (both inbound and outbound), logging connection endpoints and character volumes. - **Raw Schema:** [Sender Phone Number], [Unix Timestamp], [Receiver Phone Number], [Communication Category ID], [SMS Character Length] - **Sociometric Utility:** This allows researchers to model the exact temporal rhythm of text-messaging communication (burstiness, circadian patterns, response latency) and correlate text traffic with self-reported friendship closeness.

1.4.3 3. Voice Call logs (Raw_Telemetry_Data/PhoneCall/)

Logs voice communications, detailing phone conversations and connection attempts: - **Mechanics:** Records incoming, outgoing, and missed call events directly from the phone’s communication stack. - **Raw Schema:** [Caller Phone Number], [Unix Timestamp], [Callee Phone Number], [Call Direction/Status (Inbound/Outbound/Missed)], [Call Duration (Seconds)] - **Sociometric Utility:** Voice call durations represent a highly intensive dimension of social tie maintenance, capturing active verbal interactions which contrast with passive Bluetooth co-presence or short text messages.

1.4.4 4. Email/Mail Metadata (Raw_Telemetry_Data/Mail/)

Logs email transactions sent and received through participants’ official Notre Dame NetID email accounts: - **Mechanics:** Passively captures metadata headers of incoming and outgoing emails routed through the university’s mail exchange servers. - **Raw Schema:** [Sender ID/Phone Number], [Unix Timestamp], [Sender Email Address], [Receiver

Email/Listserv Address] - *Example:* 5550100101, 1320120074, sender.1@nd.edu, listserv@listserv.nd.edu - **Sociometric Utility:** Traces academic and listserv email networks, allowing researchers to study how official, institutional communications overlay on peer-level face-to-face and text messaging networks.

1.4.5 5. Location/GPS Telemetry (Raw_Telemetry_Data/Location/)

Captures high-frequency geographic coordinate streams tracking physical mobility: - **Mechanics:** Devices logged cellular, Wi-Fi, and GPS coordinates at regular background intervals. - **Raw Schema:** [Participant Phone Number], [Unix Timestamp], [Latitude], [Longitude], [Altitude], [Accuracy (Meters)] - **Sociometric Utility:** Enables spatial density mapping, co-location clustering, and the calculation of spatial co-presence matrices (identifying dyads who spend time in the same physical locations like dormitories, dining halls, or lecture theatres).

1.4.6 6. Address Book/Contacts Scrapes (Raw_Telemetry_Data/Contacts/)

Represents periodic backups/scrapes of the local address book directories stored on the participant devices: - **Mechanics:** Periodically backed up the list of saved phone contacts, capturing the active, known personal directory of each participant. - **Raw Schema:** [Ego Phone Number], [Unix Timestamp], [Contact Email/NetID], [Contact Name String] - *Example:* 5550100102, 1355893215, contact.15@nd.edu, :contact.15@nd.edu - **Sociometric Utility:** Maps the cognitive “potential” network of each participant—showing whom they have officially saved in their directory, which serves as a powerful baseline comparison against self-reported survey nominations and actual contact events.

1.5 Technical Appendix: Sensor Calibration & Footnotes

For researchers seeking deeper technical specifications regarding the NetSense mobile sensing infrastructure, the following notes summarize the hardware and software calibrations documented in the primary study engineering reports:¹

¹These technical details are compiled from and cross-referenced with Rachael Purta’s dissertation *Characterizing Bluetooth Low Energy Beacons for Studying Social Behavior* (Notre Dame, 2019) and the engineering study logs.

1. **Classic Bluetooth Duty Cycle:** The passive co-presence logs used in NetSense (2011–2014) operated under the **Classic Bluetooth standard** (pre-Bluetooth Low Energy v4.0). Under this protocol, smartphones continuously maintained a background scanning service while broadcasting their hardware pseudonym in a “discoverable” state. Indoor signal propagation operated in the 2.4 GHz band with a standard physical range of **10 to 50 meters**.
2. **RSSI to Distance Attenuation Limits:** While Received Signal Strength Indicators (RSSI) are logged in the raw files (typically ranging from -30 dBm to -100 dBm), direct distance translation using classic path loss formulas is highly volatile. Multipath signal reflections off indoor structures and structural body shielding (e.g., carrying the phone in a pocket versus in-hand) create high signal variability. Thus, RSSI values in the cleaned files are treated as **categorical proximity bins** (identifying close co-presence vs. distal background detection) rather than precise meter-level distances.
3. **The Episode Aggregation Window:** Raw signal dropouts are common in continuous mobile scans. To prevent brief 1-second signal interruptions from falsely splitting a continuous physical encounter into multiple separate records, raw scans were processed through a **greedy temporal aggregator**. The aggregator greedily combined separate contact events into a single continuous **proximity episode** if the elapsed gap between detections was less than a determined threshold of **25 seconds**.