

# Wie das BACNet funktioniert

Raphael Kreft

27. April 2021

## Inhaltsverzeichnis

<b>1</b>	<b>Das Secure Scuttlebutt Prinzip</b>	<b>1</b>
<b>2</b>	<b>Feeds und Events im BACNet</b>	<b>1</b>
2.1	Der Masterfeed . . . . .	1
2.2	Feeds . . . . .	1
<b>3</b>	<b>Aufbau der Aktuellen BACNet Software</b>	<b>1</b>
3.1	Der Kern ist die Datenbank und Ihre Software . . . . .	1
3.2	Datenübertragung und Synchronisierung . . . . .	2

In diesem Artikel wird beschrieben wie das BACNet funktioniert, welches im FS2020 in Projekten der Vorlesungsteilnehmer implementiert wurde. Dazu beleuchten wir zunächst kurz das Grundprinzip des BACNet: Secure Scuttlebutt und beschäftigen uns dann mit der genauen Funktionsweise des BACNet.

## 1 Das Secure Scuttlebutt Prinzip

Das BACNet funktioniert nach dem Secure Scuttlebutt Prinzip. Nachrichten werden nicht per Pakete übertragen und wieder vergessen, sondern in Append-Only-Logs organisiert.

Teilnehmer des Netzwerks erstellen solche Append-Only-Logs, auch Feeds genannt. Jeder Feed hat eine feste Quelle und folgende Eigenschaften:

- Der Feed besteht aus Events welche aneinander gehängt werden
- Jedes Event enthält die Nutzdaten und Metadaten und verweist auf das vorherige Event
- Zu jedem Feed gehört ein Schlüsselpaar, welches genutzt wird um events zu signieren bzw verifizieren

Nutzer können nun solche Feeds abonnieren und erhalten und halten die Daten dieser Feeds lokal auf Ihrem Rechner. Somit entsteht ein dezentrales Netzwerk. Dabei ist es egal auf welchem Weg die Daten übertragen und zwischen den einzelnen Teilnehmern synchronisiert werden. Somit kann neben einer normalen UDP/TCP Verbindung, auch USB-Stick-Austausch oder Radiowellenübertragung genutzt werden.

## 2 Feeds und Events im BACNet

Jede Lokale Datenbankinstanz stellt im BACNet einen Netzwerkteilnehmer dar. Sie besitzt genau einen Masterfeed und beliebig viele Feeds.

### 2.1 Der Masterfeed

- Wird standardmässig immer zwischeneinander synchronisiert
- Enthält Events bezüglich welchen Feeds man abonniert/vertraut und welche man selber bereitstellt.
- Wird beim Erstellen einer Datenbank immer als erstes erzeugt

### 2.2 Feeds

- Können beliebig viele für jede Applikation erstellt werden
- Jeder Feed besitzt einen eigenen Schlüssel bzw Schlüsselpaar
- Jeder Feed gehört zu genau EINEM Masterfeed, das erste Event zeigt immer darauf
- Man muss Feeds vertrauen um bei synchronisation Daten von Ihm abzugleichen

## 3 Aufbau der Aktuellen BACNet Software

### 3.1 Der Kern ist die Datenbank und Ihre Software

Das Zentrale Element des BACNet sind die Datenbankinstanzen, die Datenbank und die darum Entwickelte Software werden von Allen anderen Gruppen genutzt um Events zu speichern oder abzufragen. Die Gruppe LogMerge stellt hierbei ein Interface zur Verfügung über welches die Datenbank auch einfach abgefragt/gespeist werden kann, dabei werden Events verifiziert und gefiltert, dafür wird bei über die DatenbankSoftware bzw feddctrl abgefragt welchen feeds man vertraut bzw welche Feeds man teilen möchte.

**LogMerge ist eine API an die Transportschicht:**

- abgefragt werden können events von Feeds nach sequenznummer(berreich)

- Eingefügt werden können alle Events denen man vertraut
- Ob einfügen oder auslesen, LogMerge überprüft und filtert die Events wie oben beschrieben

**Events erstellen und in die Datenbank einfügen(Applikationssicht):** Die LogMerge Gruppe bietet auch ein Event-CreationTool an über welches Events und Feeds(=erste Events) erstellt werden können und sollten. Applikationen nutzen dann Die DatenbankSoftware um die Events in die DB einzufügen.

**Kontrolle des Netzwerks** Die Gruppe FeedCtrl nutzt die angebotene Funktion der Datenbanksoftware und LogMerge(es existieren extra Tabellen für Feedctrl in der DB) um eine API bereitzustellen, welche es ermöglicht anderen Feeds zu vertrauen, welche zu erstellen, zu blocken usw. Intern werden hier Events erstellt und gespeichert, sowie die Datenbank abgefragt.

### 3.2 Datenübertragung und Synchronisierung

Die Datenübertragung geschieht auf der Transportschicht: Die Gruppen der Transportschicht nutzen das Interface von LogMerge um Events abzufragen und zu speichern. Zur Übertragung werden die Events dann in das Binärformat PCAP gebracht.

**Synchronisierung** Die eigentliche Datenübertragung findet während der Synchronisierung statt. Hier findet ein per Protokoll definierter Informationsaustausch zwischen zwei Netzwerkteilnehmern statt. Hier wird mitgeteilt auf welchem Stand der jeweilige Teilnehmer ist und fehlende Informationen werden angefragt und ausgetauscht. Viele Gruppen haben dies gehardcoded in Ihrem Code, wer etwas, mehr Komfort haben möchte, kann die API der Gruppe Logsync dazu nutzen.