

# Desynchronized Channel Access with Carrier Sensing for large-scale wireless IoT Networks

**Tobias Tuchscherer**  
**TU Chemnitz, Professur Kommunikationsnetze**

**tobias.tuchscherer@etit.tu-chemnitz.de**  
**t.tuchscherer@mcloud-systems.com**



1. Einführung - IoT Kanalzugriffsmechanismen
2. IoT Verkehrscharakteristik
3. Desynchronisationsansätze
4. Neue Desynchronisationsmethode
5. Performanzanalyse
6. Zusammenfassung und Ausblick

- Simple Kanalzugriffsmechanismen, wie **ALOHA** (ohne Carrier-Sensing)
- Notwendigkeit nach besseren Zugriffsmechanismen mit steigenden Netzgrößen bzw. unter kritischen Netzauslastungssituation für LPWAN Protokolle
  - Reduzieren von Kollisionen und dadurch Sendewiederholungen
  - Gewährleistung von ausreichend Throughput und stabiler Netzperformanz
- **Slotted ALOHA**
  - Unterteilung der Zeitachse in feste Zeitslots, auf die zufällig zugegriffen wird
  - Kollisionen nur wenn ein Zugriff auf den selben Zeitslot von zwei Übertragungen getätigt wird
  - Dadurch niedrigere Anzahl an Kollisionen im Vergleich zu reinen ALOHA , aber: zusätzliche Nachrichten und Nachrichten-Overhead notwendig
- **CSMA**
  - Lauschen auf dem Kanal vor einer Übertragung → Back-Off, falls eine Übertragung nicht erfolgreich ist (Kanal belegt)
  - Kein zusätzlicher Overhead erforderlich, aber Energieverbrauch im Listening Modus höher als im Sleep Modus → höhere Energieeffizienz als reiner ALOHA, aufgrund von reduzierten Sendewiederholungen

- Batterielebenszeit kritisch für Profitabilität des Dienstes (**OPEX**)
- Essenziell sind günstige, low-power Endgeräte, die **energieeffiziente Kommunikationsprotokolle** nutzen
- Im Durchschnitt niedrigeres Verkehrsvolumen und Sitzungszeit pro Endgerät als bei Mensch-zu-Mensch Kommunikation
- Asymmetrisch: mehr Uplink (Status- und Messberichte) als Downlink Verkehr (Konfigurationsänderungen, Firmware/Software Updates)
- Periodisches Uplink-Verkehrsmuster (vorallem die legal regulierte Messungen übertragen) im Intervall von typischerweise 15, 30 oder 60min → bursty Verkehr
- Konstanter Verkehr der sich über den Tag verteilt (im Vergleich zu tagaktiven Mensch-zu-Mensch Kommunikationsverkehr) mit stündlichen Spitzen

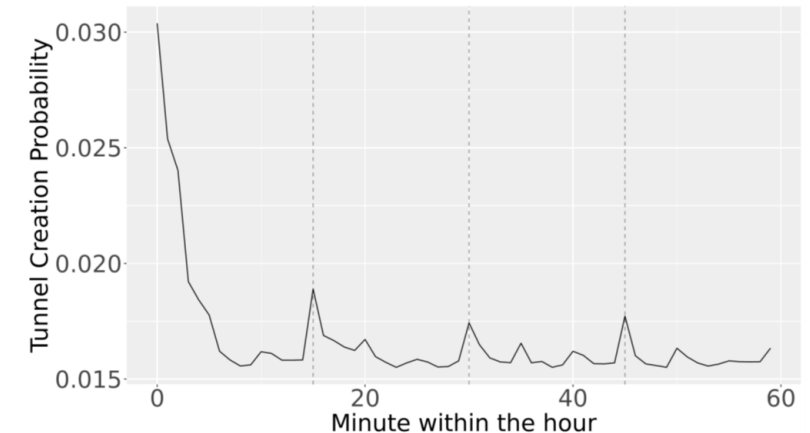


Fig. 1: IoT Verkehrscharakteristik über eine Stunde Zeitintervall aus: Simon Raffeck, Stefan Geissler, Michael Krolikowski, Steffen Gebert, and Tobias Hoßfeld. 2022. Data Usage in IoT: A Characterization of GTP Tunnels in M2M Mobile Networks. In NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. IEEE Press, 1–6. <https://doi.org/10.1109/NOMS54207.2022.9789901>.

# 3. Desynchronisationsansätze

- **Desynchronisationsansätze DESYNC-TDMA** nach J. Degeys, I. Rose, A. Patel and R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks"
  - Biologisch inspirierter Kanalzugriff Desynchronisationsalgorithmus
  - Annahme: vollständig verbundenes Kommunikationsnetz mit einer festen Anzahl an Knoten  $n$  mit derselben Periodizität  $T$
  - **Gleichmäßige Verteilung** der Sendeevents auf der Zeitachse über derselben Periodizität  $T$
  - Initialisierungsphase: „Firing“ Nachricht von jedem Knoten im Netzwerk
  - Lauschen auf den Kanal vor und nach einer Übertragung
  - Knoten optimieren ihre Sendezeitpunkte lokal optimal zwischen den direkten Phasennachbarn („Ausmittlung“ zwischen den Sendezeitpunkten)

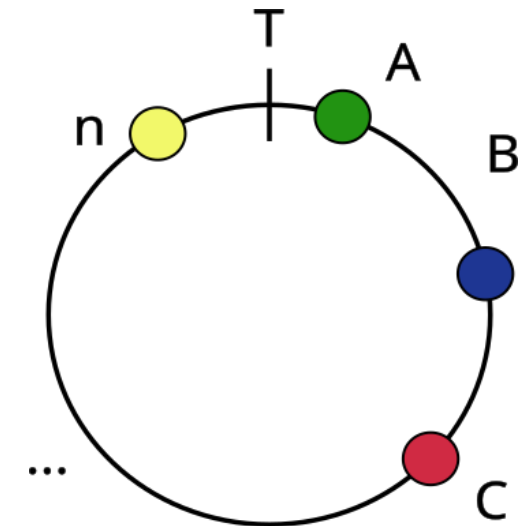
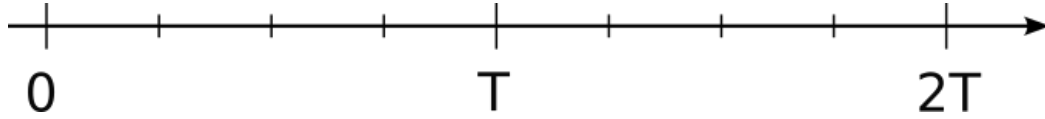
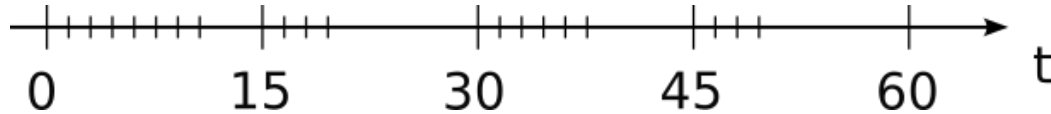


Fig 2: Desynchronisierter Kanalzugriff mit  $n$  Knoten und derselben Periodizität  $T$

# 4. Neue Desynchronisationsmethode



**DESYNC-TDMA: Gleichverteilung** der  
Sendezeitpunkte aller Knoten über die einheitliche  
Periodendauer



**Idee:** Anpassung des Desynchronisationsverhaltens  
nach tatsächlichen IoT-Verkehrscharakteristika

- **Unser Ansatz:** alle aktiven Knoten passen ihre Sendezeitpunkte so an, dass sich möglichst **große leere Intervalle** ohne Sendeevents auf der Zeitachse ergeben
- **Vorteile neue Methode:**
  - Unterschiedliche Periodizitäten möglich
  - Für neue Sensoren im Netz: große leere Intervalle für RX-Fenster
  - Erwartung einer stabileren Netzperformanz bei realistischem Netzbetrieb (variable Anzahl an Knoten, eventbasierte Sensoren, Sensoren, die die Methode nicht nutzen)

# 4. Neue Desynchronisationsmethode

- Verschiedene Übertragungszyklen bei DESYNC-TDMA problematisch:
  - Verschiedene Nachbarn beim Desynchronisieren in folgenden Zyklen möglich, was zu synchronisiertem Übertragungsverhalten führen kann:

$$T_A \ll T_B \cong T_C$$

- Bewegliche Knoten führen zu leeren Stellen auf der Zeitachse → Desynchronisationsschritt nötig; und Desynchronisation des Knotens in der neuen Zelle

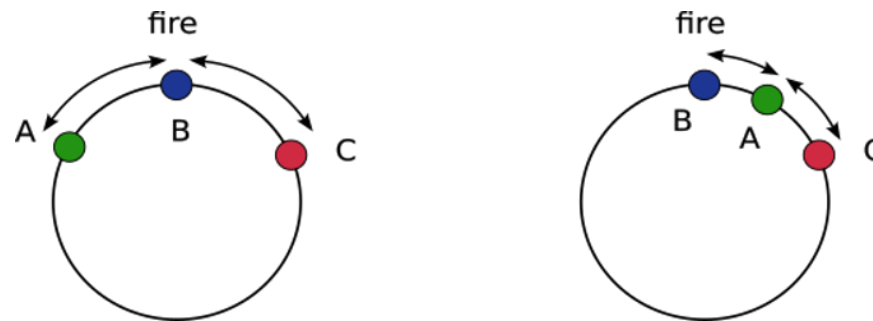


Fig 4: Desynchronisationsversuch über zwei Zyklen, bei dem Knoten A deutlich geringere Periodizität aufweist

# 4. Neue Desynchronisationsmethode

- Definiere Trigger-Funktion:  $\phi_i(t) \in [0,1]$  eines Knotens  $i$ , die mit  $\phi_i(t) = 1$  Sendestart Sensor  $i$  (Bsp.:  $\phi_i(t) = 0.5$  bedeutet Sensor ist die Hälfte der Sendeperiode durchlaufen)
- Für einen initiale Uplink Übertragung zielt ein Knoten  $a$  idealerweise auf den Zeitpunkt, welcher am weitesten entfernt von allen anderen Sendeevents ist, dieser kann mit  $T_{max}$  als maximal auftretende Transmissionsperiodizität (typischerweise 60min) folgend ausgedrückt werden:

$$t^* = \max_{t \in (0, T_{max})} \sqrt{\sum_i^N |\phi_i - \phi_a(t)|^2 + |T_{max} - \phi_a(t)|^2 + |\phi_a(t)|^2}$$

- Nach initialen Uplink versucht der Knoten ebenfalls das leere Zeitintervall zu maximieren



# 4. Neue Desynchronisationsmethode

- Dabei wird die Distanz  $\tilde{\Delta}_i$  zwischen den Übertragungsstart des vorherigen Knotens  $i - 1$  und der Übertragungsstart des Knotens  $i$  gemessen
- Algorithmus terminiert, wenn eine definierte Guardband-Schranke  $GB$  erreicht wird:  $\tilde{\Delta}_i \leq GB$
- $GB$  wird beeinflusst durch z.B. Paketgrößen, duty cycle, Clockgenauigkeit der Endgeräte, Frequenzbandlimitationen

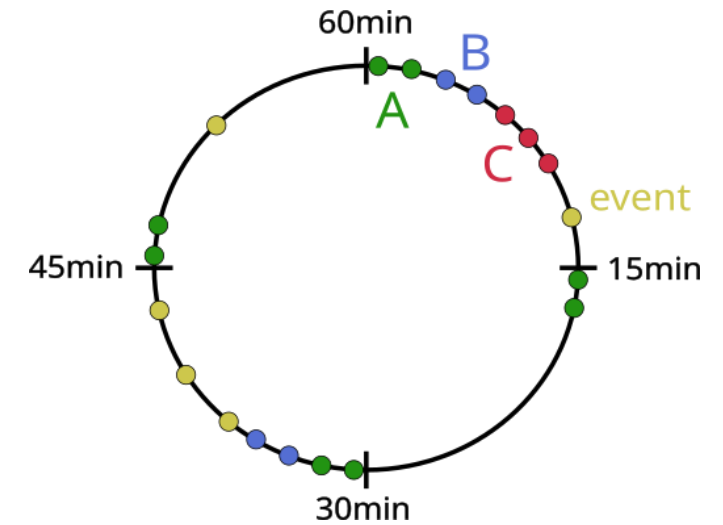


Fig 3: Desynchronisierter Zustand für die vorgestellte Methode

- Implementierung Algorithmus in einer LPWAN-Netzsimulator (LoRaWAN)
- Verschiedene Tools zum Simulieren von LoRaWAN-Netzen sind verfügbar, aber funktionell unvollständig im Hinblick auf der vorgestellten Methode
- Erstellen einer vollständigen Performanzanalyse über einer LoRaWAN Simulation
  - Metrik: Kollisionsrate, Throughput, Radio Ressource Utilization
  - Einzelne und multiple Gateway Szenarien
  - Hohe Anzahl an Knoten (>1000)
  - Verschiedene Paketgrößen
  - Hidden Station Problem soll untersucht werden, in Hinblick auf den Einfluss der implementierten Methode
  - Mischbetrieb
  - Geringerer Energieverbrauch als reines CSMA, bei vergleichbarer Netzperformanz

- Algorithmus für Kanalzugriff unter Zuhilfenahme adaptiver Empfangsfenster für Carrier-Sensing
- Im Netzbetrieb: Desynchronisation der Sender mit alternativer Kanalbelegung als DESYNC-TDMA, angepasst an tatsächliche IoT Verkehrscharakteristika
- Vereinfachende Annahmen von DESYNC-TDMA können vernachlässigt werden: Unterschiedliche Paketlängen, Sendeperioden, adaptive Anzahl von Knoten bzw. eventbasierte Sender möglich
- Mit jeder erfolgreichen Sendeperiode lässt sich die Größe des Empfangsfenster für das notwendige Carrier-Sensing reduzieren, im Hinblick auf energiesparsameren Betrieb



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

**Fragen?**

- Eine TX-Einheit  $TU$  besteht aus: ein TX-Fenster, je zwei RX-Fenster vor (zusätzliche Annahme in LoRaWAN) und nach der Übertragung und zwei Guardbands
- **Initialer Kanalzugriff eines neuen Knotens:**
  - In der ersten Übertragungsperiode öffne RX-Fenster bei  $t = 0s$
  - Lauschen auf den Kanal bis der Geräuschpegel deutlich sinkt ( $\tilde{t}_e = 0$ )
  - Kapazität  $c$  der potentiell offenen TX-Einheiten bis  $T$ :

$$c = \left\lfloor \frac{|T - \tilde{t}_e|}{TU} \right\rfloor$$

- Initiale Kanalzugriffszeit soll dann bei  $t_a^*$  erfolgen:

$$t_a^* = \tilde{t}_e + g \cdot TU$$

mit der gleichverteilten Zufallsvariable  $g$  aus den Intervall  $m = |T - \tilde{t}_e|$  mit  $g \in G(m)$

1. Bei erfolgreichem Zugriff:

$$t_a^{new} = \frac{T - \tilde{t}_e}{2} + \tilde{t}_e$$

Im nächsten Zyklus: Verkleinern des RX-Fensters zum Messen von  $\tilde{t}_e^{new}$

$$t_a^{*new} = \alpha \cdot t_a^{new} + (1 - \alpha) \cdot t_a^*$$

- Algorithmus terminiert, falls gilt:

$$|t_a^{*new} - \tilde{t}_e^{new}| \approx 2 \cdot GB$$

- Korrektur von Zeitdrift, anhand von gemessenen  $\tilde{t}_e$

2. Im Falle einer Kollision: Backoff-Strategie abhängig von der Anzahl der Sendewiederholungsversuche innerhalb der Sendeperiode:

- Falls  $r \in \mathbb{N}$  Anzahl der Sendewiederholungen, ziehe  $r$  gleichverteilte Zufallsvariablen  $m_1, m_2, \dots, m_r \in G(c)$  mit:

$$c = \left\lfloor \frac{|T - t_a^{*new}|}{TU} \right\rfloor$$

- Übertragungsversuche zu den Zeitpunkten  $t_a(m_1), t_a(m_2), \dots, t_a(m_r)$
- Nach erster erfolgreichen Übertragen beginnt Algorithmus mit Schritt 1