



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Fakultät Elektrotechnik und Informationstechnik

Institut für Nachrichtentechnik

PROJEKT VOM ACTUAL TOPICS

zum Thema

Python basierte Simulation des Helfer-Knotens im drahtlos
vermaschten Netzwerk

vorgelegt von Xiang, Zuo
im Studiengang Informationstechnik, Jg. 2014
geboren am 21.08.1991 in China

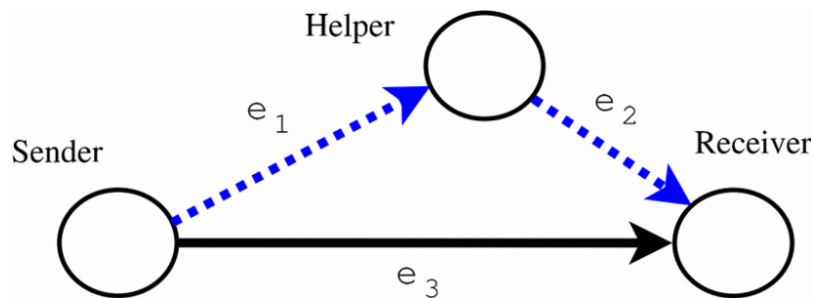
Betreuer: Dipl.-Inf. Frank Wilhelm
Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Dr. h.c. Frank H.P. Fitzek
Tag der Einreichung: 21.03.2009

Aufgabenstellung

Python-based Simulation of the Helper Node in Wireless Mesh Network

Description

The broadcast nature of the wireless channel allows neighboring nodes to overhear transmissions of the main route. Considering that wireless channels suffer from significant packet losses, this overhearing opens interesting possibilities for exploiting neighboring nodes to forward a packet opportunistically in order to enhance overall performance.



On the other hand the sender and helper cannot send at the same time. If the helper cannot provide new information to the receiver it is just stealing the sending time of the sender.

The task is to build a basic simulation of this scenario using Kodo Python. This simulation can then be used to compare different sending policies of the helper node.

Prerequisites

- * Python programming
- * Kodo-Python <https://github.com/steinwurf/kodo-python>
- * Plotting

Supervisor

- * Dipl.-Inf. Frank Wilhelm <frank.wilhelm@tu-dresden.de>



Python basierte Simulation des Helfer-Knotens im drahtlos vermaschten Netzwerk

The broadcast nature of the wireless channel allows neighboring nodes to overhear transmissions of the main route. Considering that wireless channels suffer from significant packet losses, this overhearing opens interesting possibilities for exploiting neighboring nodes to forward a packet opportunistically in order to enhance overall performance.

picture

On the other hand the sender and helper cannot send at the same time. If the helper cannot provide new information to the receiver it is just stealing the sending time of the sender.

The task is to build a basic simulation of this scenario using Kodo Python. This simulation can then be used to compare different sending policies of the helper node.

| | |
|----------------------|---|
| Betreuer: | Dipl.-Inf. Frank Wilhelm |
| Hochschullehrer: | Prof. Dr.-Ing. Dr. h.c. Frank H.P. Fitzek |
| Tag der Einreichung: | 21.03.2009 |

Bearbeiter: Xiang, Zuo

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation | 1 |
| 2 | Methode | 3 |
| 2.1 | Aufgabenbeschreibung | 3 |
| 2.1.1 | Szenario der Simulation | 3 |
| 2.1.2 | Simulationsarbeit | 3 |
| 2.2 | PlayNCool Algorithmus | 4 |
| 2.3 | Vorstellung der Simulationswerkzeuge | 5 |
| 2.3.1 | Kodo | 5 |
| 3 | Ergebnisee | 7 |
| 3.1 | Ergebnisee | 7 |
| 4 | Zusammenfassung und Aussicht | 9 |
| 4.1 | Szenario | 9 |
| 4.2 | Vorstellung der Simulationswerkzeuge | 9 |
| 4.2.1 | Kodo | 9 |
| | Literaturverzeichnis | 11 |

Abbildungsverzeichnis

| | | |
|-----|-----------------------------------|---|
| 2.1 | Szenario der Simulation | 3 |
| 4.1 | Szenario der Simulation | 9 |

Tabellenverzeichnis

Quelltextverzeichnis

1 Einleitung

1.1 Motivation

2 Methode

2.1 Aufgabenbeschreibung

2.1.1 Szenario der Simulation

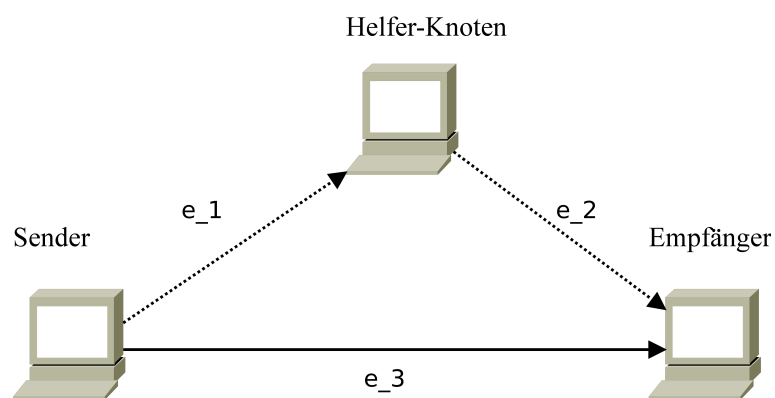


Abbildung 2.1: Szenario der Simulation

Weil in der Abbildung 2.1 beschriebenes Szenario der häufigste Fall in jeder Verbindung des vermaschten Netzwerks ist, sollte es in dieser Aufgabe detailliert analysiert und simuliert.

Die Paketverlustwahrscheinlichkeiten zwischen Sender, Helfer-Knoten sowie Empfänger werden jeweils durch e_1 , e_2 sowie e_3 charakterisiert. Um die Analyse sowie Simulation zu vereinfachen, wird in dieser Aufgabe angenommen, dass Verbindungen des Helfer-Knotens symmetrisch sind ($e_1 = e_2$).

2.1.2 Simulationsarbeit

Wie in der Einleitung erwähnt, dass die heuristische Weitersendung des Helfer-Knotens zur Verbesserung der Netzwerkleistung beitragen könnte. Zwei Situationen werden in dieser Aufgabe mit Python simuliert und auch verglichen.

- (1) Datenübertragung mit Netzkodierung(RLNC) ohne Helfer-Knoten

(2) Datenübertragung mit Netzwerkkodierung(RLNC) mit Helfer-Knoten

Bei der zweiten Situation gibt es viele unterschiedliche Strategien, in dieser Aufgabe wird der PlayNCool Algorithmus[2] mit Kodo-Python implementiert und simuliert.

Weil zufällige Netzwerkkodierung bei den Simulationen verwendet wird, können Sender, Helfer-Knoten und Empfänger auch als Encoder, Recoder sowie Decoder genannt.

Wenn das Helfer-Knoten Pakete zu senden beginnt, liegen verschiedene Varianten für die Disposition zwischen Sender und Helfer vor. In dieser Simulation wird eine vereinfachte Annäherung von TDMA verwendet, nämlich Sender und Helfer besetzen das Kanal alternativ.

Bei jeder Simulation werden Paketverlustwahrscheinlichkeiten(nämlich e_1 , e_2 , e_3) als Variablen eingestellt. Um einen Vergleich durchzuführen, wird die Anzahl von Paketen bekommen, die insgesamt und auch nur von dem Helfer gesendet werden. Danach werden Daten mehrerer Simulationen verarbeitet und in Grafik mit Konfidenzintervall dargestellt.

2.2 PlayNCool Algorithmus

PlayNCool ist ein opportunistisches Protokoll mit Netzwerkkodierung für lokale Optimierung vom Routing im drahtlos vermaschten Netzwerk. [2]

Eine wichtige Leistungsmerkmal des Algorithmus ist, dass das Helfer-Knoten nicht sofort sondern nach gewisser Zeit Pakete transportiert, um nützlichere Informationen dem Empfänger anzubieten. Wegen der Vermeidung von Konkurrenz des drahtlosen Kanals an der Anfangsphase, die Leistung der Verbindung wird verbessert.

Für die Erzeugung der kodierten Pakete mit der Generationsgröße g wird RLNC[1] verwendet. Die Anzahl der Paketen am Helfer-Knoten, die von dem Sender transportiert werden, wird als r definiert. Dieser Wert sollte als der Schwellenwert für den Anfang der Paketübertragung zwischen Helfer und Empfänger.

Laut der PlayNCool Politik wird r durch folgende Formel berechnet. [2]

$$r = -g \cdot \frac{C}{D - (1 - e_3) \cdot C} \quad (2.1)$$

$$C = -1 + e_2 + e_3 - e_1 \cdot e_3 \quad (2.2)$$

$$D = (2 - e_3 - e_2) \cdot (e_3 - e_1 \cdot e_3) \quad (2.3)$$

2.3 Vorstellung der Simulationswerkzeuge

2.3.1 Kodo

3 Ergebnisee

3.1 Ergebnisee

4 Zusammenfassung und Aussicht

4.1 Szenario

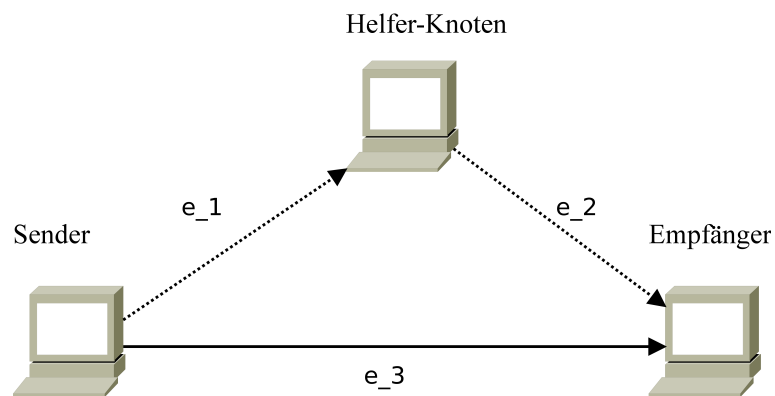


Abbildung 4.1: Szenario der Simulation

Weil in der Abbildung 2.1 beschriebene Szenario der häufigste Fall in jeder Verbindung, deshalb wird diese Szenario detailliert analysiert und weiter simuliert.

Der Paketverlustwahrscheinlichkeiten zwischen Sender, Helfer-Knoten sowie Empfänger werden jeweils durch e_1 , e_2 sowie e_3 definiert. Um die Analyse sowie Simulation zu vereinfachen, wird es hier zuerst angenommen, dass die Verbindung des Helfer-Knotens symmetrisch ist ($e_1 = e_2$). Die komplexere Situation Ohne Symmetrie wird auch analysiert aber wegen der Zeit noch nicht simuliert.

4.2 Vorstellung der Simulationswerkzeuge

4.2.1 Kodo

Literaturverzeichnis

- [1] *Introduction to Network Coding*. Steinwurf, 2012. URL: http://kodo-docs.steinwurf.com/en/latest/nc_intro.html (siehe S. 4).
- [2] Peyman Pahlevani, Daniel E Lucani, Morten V Pedersen und Frank HP Fitzek. „Playncool: opportunistic network coding for local optimization of routing in wireless mesh networks“. In: *Globecom Workshops (GC Wkshps), 2013 IEEE*. IEEE. 2013, S. 812–817 (siehe S. 4).

