

2018-05-06

Energieeffizientes Routing in Ad-Hoc Netzen

└─ Vorarbeit
 └─ Whoami
 └─ Whoami

Whoami

u Marcel Ebbrecht

u Viele Jahre LAN, WAN, WLAN, TCP/IP

u n Jahre Informatik

u marcel.ebbrecht@gmail.com

u <https://github.com/marcelebbrecht/powerrouting>

- Bla
- Bla
- Geht nur mich und Prüfungsamt was an
- Bei Fragen, etc.
- Software, Thesis und Präsentation wenn durch, Commit an INETMANET

Energieeffizientes Routing in Ad-Hoc Netzen

└ Vorarbeit

└ Warum sind wir hier?

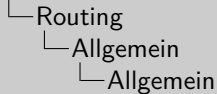
└ Warum sind wir hier?

Warum sind wir hier?

- Was ist das Problem und warum ist das relevant?
- OLSR und AODV im Wesentlichen
- Optimierung nach Ladezustand
- Nach dem Vortrag gerne mehr Videos und Auswertungen

- Problem wird beschrieben, Relevanz erläutert
- Grundeigenschaften und Funktionen, die für das Verständnis der Anpassung wichtig sind
- Die eigentliche Arbeit, Videos, Auswertungen, Vergleiche, Fazit
- Sehr viel vorhanden, im Vortrag größtenteils AODV, ist zwar nicht so sauber implementiert wie OLSR, aber in Videos schneller erkennbar

Energieeffizientes Routing in Ad-Hoc Netzen



Allgemein

- TCP/IP Layer 3
- IPv4 (IPv6)
- Statisches Routing
- Dynamisches Routing (reaktiv oder proaktiv)

- Vermittlung zwischen IP-Subnetzen
- Die Arbeit betrachtet IPv4, Aussagen gelten aber auch für IPv6, Protokolle können das
- Pflege von Routeninformationen (Ziel, Gateway, Metrik) per Hand, kleine statische Netze
- Bereitstellung über Routingprotokolle, wenn gebraucht oder permanent
- Die Güte einer Verbindung, i.d.R. kommt diese in der Routingtabelle des Systems als Zahl an, je kleiner, desto besser. Simple Metrik: HopCount (Anzahl der Knoten bis zum Ziel) oder differenzierter -> Das entscheiden die Routingprotokolle, gr. und/oder dyn. Netze

Energieeffizientes Routing in Ad-Hoc Netzen

└ Routing

└ Ad-hoc On-demand Distance Vector

└ Ad-hoc On-demand Distance Vector

Ad-hoc On-demand Distance Vector

- Reaktiv
- Geringer Overhead / Hohe Latenz
- Loop avoidance
- Precursor-Listen
- HopCount maßgebend

- Fordert Routen an, wenn sie gebraucht werden (Name!!!)
- Hierdurch kein Traffic, wenn keine neuen Routen gebraucht werden, dafür aber Wartezeit beim Bedarf nach neuer Route
- Durch Sequenznummern wird das DV-Prot. typische Problem vermieden (im Prinzip werden nur Infos verarbeitet, die eine höhere Nummer haben)
- Speichert pro Route Informationen über Hosts, die diese Route nutzen
- HC einzig maßgebend für Routenwahl durch OS (!= Protokoll!!!)
- Gute Anwendung: Sensornetz - ein Ziel, wenig Overhead (Strom)

Energieeffizientes Routing in Ad-Hoc Netzen

└ Routing

└ AODV Ablauf

└ AODV Ablauf

AODV Ablauf

- Route benötigt -> RREQ
- Route bekannt -> RREP
- Route nicht mehr verfügbar -> RERR
- Sammelt Informationen aus empfangenen Nachrichten
- Entfernt Routen durch Timeout oder RERR

- Versand per Broadcast, TTL wird schrittweise gesteigert bei Timeout
- Ein Host der eine Route kennt, schickt RREP per Unicast an den Originator des RREQ
- Stellt ein Router den Ausfall einer Route fest, ob durch Detektion oder RERR, wird er die Precursors dieser Routen informieren (Hier UC, BC geht auch), mehrere Ziele in einem RERR
- Gewinnt Routinginformationen aus AODV RREQ (Broadcast) und weitergeleiteten RREPs (Unicast) von Nachbarn, Refresh
- Nach gewisser Zeit verschwinden inaktive Routen

Energieeffizientes Routing in Ad-Hoc Netzen

└ Routing

└ Optimized Link State Routing

└ Optimized Link State Routing

Optimized Link State Routing

- Proaktiv
- OLSR Paket mit Messages
- MID, HELLO, TC
- 1HNB, 2HNB, MPRSet („optimized“)
- Willingness

- Name!!! Permanenter Austausch von Informationen, auch ohne Anforderung von Routen -> Overhead immer gegeben, dafür Routen nach Vorlaufzeit sofort verfügbar
- Mehrere Messages per Paket möglich -> spart Traffic
- Minimale zu unterstützende Menge ... zu aufwändig wie es genau funktioniert; MID/TC BC TTL255 Routenfindung, HELLO Sensing und MPRs mitteilen
- 1 direkte Nachbarn, 2 direkte Nachbarn von 1 ohne Host selbst, MPRSet minimale Menge an 1HNB um alle 2HNB zu erreichen (jeder Router eigenes Set, regelmäßige Neuberechnung)
- Bereitschaft zu Routen, 7 hoch, 1 minimum, relevant für Aufnahme ins MPRSet

Energieeffizientes Routing in Ad-Hoc Netzen

- └ Wo ist das Problem
 - └ Warum wichtig
 - └ Warum wichtig

Warum wichtig

- Steigender Bedarf nach Bandbreite und Verfügbarkeit
- Viele mobile Geräte
- ABER: Routing braucht Strom
- Akkulation begrenzt
- Routingverfahren berücksichtigen es nicht

- Wird langfristig mit den zentral arbeitenden Systemen (Funkmast) unnötig aufwändig
- Nahezu jeder hat ein Smartphone in der Tasche, dass als Router dienen könnte -> Public MESH wird möglich
- Bla
- Bla
- Die Last muss verteilt werden um die Akzeptanz zu steigern -> Lastverteilung nach Akkustand!

Energieeffizientes Routing in Ad-Hoc Netzen

└ Omnet

└ Ganz kurz: Omnet

└ Ganz kurz: Omnet

Ganz kurz: Omnet

• Freie Software, C++, nette Community

• Ereignisbasiert, Modular

• Framework INET kann alles, wirklich alles, also viel

• Inetmanet gibt MANET (AODV okay/buggy, OLSR korrekt)

• Unbrauchbar: Auswertung -> PERL/Gnuplot (R Versionskonflikte)

- Keine Warum willst du das-Fragen, einfach Antworten
- Alles austauschbar, gute Struktur, relativ niedrige Einstiegsschmerzen
- Von PAN bis Geostationär (manchmal buggy)
- Fork
- Kleine Datenmengen teils unterschiedliche Ergebnisse, große Datenmengen lahm/instabil, bestimmte Funktionen gar nicht verfügbar
- richtig, richtig cooles Spielzeug

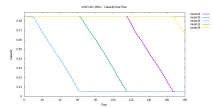
Energieeffizientes Routing in Ad-Hoc Netzen

└ Simulation

└ AODV Vanilla

└ AODV Vanilla

AODV Vanilla



- Man sieht, wie es beginnt und wie der Akku nach und nach geleert wird, erst dann Umschaltung
- Nur senden verbraucht Strom (besser erkennbar, Rest in der Natur nicht beeinflussbar)
- OLSR zeigt selbes Verhalten
- AODV Implementation manchmal instabil (Vanilla), scheinbar nicht 100% korrekt implementiert, Instabilität durch feste Seeds umgangen

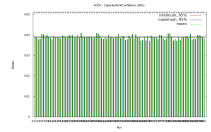
Energieeffizientes Routing in Ad-Hoc Netzen

└ Simulation

└ AODV Vanilla

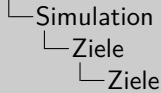
└ AODV Vanilla

AODV Vanilla



- 100 Simulationen mit verschiedenen Zufallszahlen
- Ergebnisse sehr stabil

Energieeffizientes Routing in Ad-Hoc Netzen

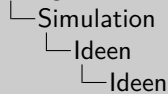


Ziele

- Lastverteilung nach relativem Akkustand
- Kompatibilität zu anderen Systemen
- Steuerung der Empfindlichkeit und Grad der Anpassung

- Bla
- Gemischte Netze sollen funktionieren
- Parameter um die ladungsabhängige Anpassung zu steuern

Energieeffizientes Routing in Ad-Hoc Netzen

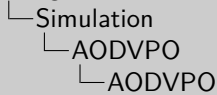


Ideen

- Es muss ein vorhandener Parameter eingesetzt werden
- AODV: HopCount künstlich anheben (begrenzt möglich)
- OLSR: Willingness ändern

- Im Prinzip sollen sich die Router schlechter machen, als sie sind, gerade in der heutigen Zeit eher selten zu beobachten
- Begrenzt durch maximalen HopCount ($8\text{Bit} = 255$) - größeres Netz, schwächere Anhebung
- Prinzipiell keine Grenze in der Größe des Netzes gesetzt. Dafür weniger fein steuerbar

Energieeffizientes Routing in Ad-Hoc Netzen

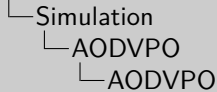


AODVPO

- Trigger $t = 0.2$ bedeutet, dass bei $C(i) \in \{0.8, 0.6, \dots, 0.2\}$ eine Anpassung stattfinden soll
- Allgemeiner: Es wird eine Anpassung ausgelöst, wenn gilt:
 $(C(i) * 100) \bmod (100t) = 0$
- $C(i) \in [0, 1]$ entspricht dem relativen Ladezustand
- Anpassung meint das Versenden von RERR, dadurch Ausfall und Neuberechnung

- Hoher Trigger bringt schnellere Reaktion, dafür mehr Ausfall (Paketverlust)

Energieeffizientes Routing in Ad-Hoc Netzen



AODVPO

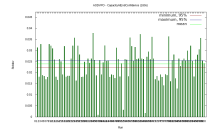
- Implementierung relativ einfach
- Funktioniert nicht bzw. zufällig
- Diagnose: AODV fehlerhaft
- Therapie: Wartezeit $W(P)$ bevor RREP erzeugt wird

- 3 Methoden erstellen, Einfügen in Hopberechnung an mehreren Stellen
- Warum nicht? Logs passen...
- Wertet nicht den HC aus, sondern nimmt erstes empfangenes RREP ... macht aber begrenzt Sinn ...
- Dann funktioniert es ... meistens ... weitergehendes Reparieren von AODV außerhalb des Umfangs dieser Arbeit ... gute paar tausend Zeilen Code

Energieeffizientes Routing in Ad-Hoc Netzen

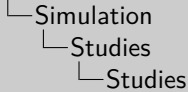
- └ Simulation
 - └ AODVPO Live
 - └ AODVPO Live

AODVPO Live

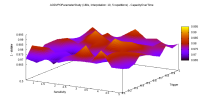


- Größere Streuung durch Zufall und Fehlfunktion
- Alter Wert ca 0,4, immer besser!!!
- Wie sehen dann eigentlich optimale Parameter aus?

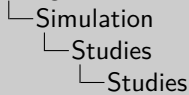
Energieeffizientes Routing in Ad-Hoc Netzen



Studies



- Äußerst zufrieden, schaute mir mal die Simulation mit Abweichung fast Null an
- s zwischen 0.5 und 5, t zwischen 0.1 und 0.9 laufen lassen - macht man t kleiner oder s größer kommen sehr extreme werte raus
- Erkannte ein weiteres Problem: Es wurde nur ein gewissen Prozentsatz an Paketen erfolgreich übertragen, nämlich:



- Ursachen: Kein Zustandkommen von Routen durch ständige RERR oder zu hohem HopCount (oder instabiles Verhalten bei Pufferüberlauf HopCount 8Bit) oder Simple Fehlfunktion Protokoll AODV

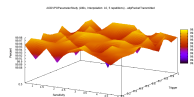
Energieeffizientes Routing in Ad-Hoc Netzen

└ Simulation

└ Studies

└ Studies

Studies



- Um jetzt einen sinnvollen Zusammenhang zu finden, muss man das kombinieren. Prinzip: Ein hoher Loss oder Abweichung disqualifiziert das Setting

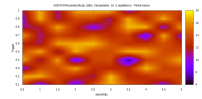
Energieeffizientes Routing in Ad-Hoc Netzen

└ Simulation

└ Performance

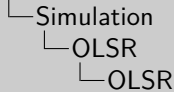
└ Performance

Performance



- Das brachte $t = 0,2$ und $s = 3$ für die o.g. Ergebnisse

Energieeffizientes Routing in Ad-Hoc Netzen



OLSR

■ Prinzipiell stabil

■ Statt HopCount-Penalty neue Willingness
 $W_{new} = \lfloor \max(1, (7 - C(i) \cdot (1 - s) - B)) \rfloor$ berechnet

■ Fließender Übergang

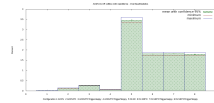
■ Keine Hacks

- Sauberer implementiert in Vanilla, weniger Ausreißer
- keine Extremen Werte, dafür nicht so genau im Zeitpunkt, dafür keine Unterbrechungen
- Man kann in diesem Szenario (bei OLSR läuft akku leer) sogar den Overhead senken, da man die Intervalle größer machen kann in denen Kontrollnachrichten geschickt werden

Energieeffizientes Routing in Ad-Hoc Netzen

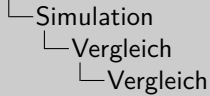
- └ Simulation
- └ Vergleich
- └ Vergleich

Vergleich

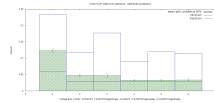


- Intervalle bei OLSRPO verdoppelt was den Stromverbrauch reduziert, trotzdem bessere Werte

Energieeffizientes Routing in Ad-Hoc Netzen



Vergleich



- OLSR Vanilla und AODVPO Trigger Happy entfernt, da zu extreme Werte das Bild verzerren bzw. die anderen Werte unlesbar machen

Energieeffizientes Routing in Ad-Hoc Netzen

└ Abschluss

└ Fazit

└ Fazit

Fazit

- Funktioniert in den gezeigten Szenarien
- Funktioniert in gemischten Netzen (Kompatibel)
- Funktioniert in größeren Netzen (Skaliert)

- Haben wir gesehen
- Nicht so gut, aber grundsätzlich funktioniert es, bei OLSR besser als bei AODV, was vielleicht wieder an der Implementierung liegt
- dito