Telematics Project - Wireless Embedded Systems (WS 08/09)

Piotr Narozny - narozny@inf - 3891503

31. März 2009

1 Vorwort

Dieser Technical Report beschreibt die Erstellung eines Routingverfahrens für das Telematics Project "Wireless Embedded Systems"des Wintersemesters 08/09 an der Freien Universität Berlin am Institut für Informatik. Projektleiter und -betreuer war Dipl.inf. Bastian Blywis.
Die konkrete Aufgabe bestand darin ein Routingverfahren für Sensorknoten basierend auf dem Scatterweb Framework zu implementieren und zu testen. Zu Anfang der Veranstaltung gab es eine kurze Einführung in die Tematik und eine Wiederholung der Kentinisse bezüglich der Routingverfahren aus den Vorlesungen Telematik und Mobilkommunikation an selber Universität. Kurz danach sollte man sich auch schon für ein Routingverfahren entscheiden und nach einigen "First Steps", die dazu dienten sich mit dem eingesetzten Sensorknoten und dem Framework vertraut zu machen, auch beginnen das gewählte Verfahren umzusetzen.
Hierbei sei angemerkt, dass ich mich zu anfangs als einziger der Gruppe für das Routingverfahren "Link State Routing“ (kurz LSR) entschieden habe, jedoch mitten im Semester auf das hier weiterhin beschriebene "Dynamic Source Routing“ -Protokoll (kurz DSR) gewechselt bin.
Die o.g. Sensorknoten wurden uns zur Verfügung gestellt, sowie das erforedrliche Zubehör um sie zu flashen. Die Knoten können sowohl per USB-Kabel, als auch mit 3 AAA-Batterien betrieben werden. Der dafür erforderliche Switch befinden sich auf den Knoten. Um die Software auf den Knoten flashen zu können befindet sich zusätzlich eine JTAG-Schnittstelle. In meinem Fall war ein MSB430 USB-Debug-Interface MSP-FET430-UIF zwischengeschaltet. Mit Hilfe des seriellen Anschlusses und der empfohlenen Software HTerm, war ich im Stande die Ausgaben des Sensorknotens auf der Konsole zu verfolgen und Input zu geben. Auf der Platine des Sensorknotens ist der Mikrocontroller MSP430 verbaut. Dieser hat wahlweise einen zusätzlichen Funkausgang für Antennen. In meinem Fall wurde keine Antenne angeschlossen, wodurch die Reichweite teilweise sehr begrenzt

war.
Die Implementierung wurde zunächst auf einem Macintosh OS X begonnen, jedoch nach enormen Anlaufschwierigkeiten mit dem Compiler auf einer Windows XP Professional Maschine erarbeitet. Zunächst gab es enorme Probleme beim Flashen der Software. Das wurde nach Austauschen des USB-Debug-Interface überwunden. Als Entwicklungsplattform war Eclipse GANYMEDE (3.4) mit CDT-Plugin im Einsatz. Zum Übersetzen der Software wurde der frei erhältliche MSP-GCC Compiler verwendet. Der In- und Output der Sensorknoten konnte wie oben erwähnt mit dem Programm HTerm nachvollzgen werden. Das war das wichtigste Hilfmittel zum Debuggen der entwickelten Software.
Das zur Verfügung gestellte Scatterweb Framework entstand an der Universität und stellte eine Wireless-Kommunikation zwischen den Sensorknoten sicher. Hier musste nichts mehr angepasst werden. Eine bereits implementierte Funktion Netsend(...) wurde benutzt um die manipulierten Daten zu verschicken. Die Verarbeitung in den einzelnen ISO/OSI-Schichten wurde intern gehandhabt und musste nicht weiter analysiert werden.
Die Anwendungsszenarien sind vielfältig und werden von der Arbeitsgruppe Telematics & Computer Systems wie folgt auszugsweise benannt:

- Überwachung in der Umwelt
- Gebäudeüberwachung
- AdHoc-Prozessüberwachung

2 Technologie

Nachfolgend stelle ich kurz und knapp die relevanten Routingverfahren vor.

2.1 Link State Routing Protocol

Das LSR-Protokoll basieren ganz anders als die Distanzvektorprotokolle auf dem Austausch kompletter Netzwerktopologien untereinander. Dieses proaktive Protokoll verwaltet die gesamte Netztopologie des Netzwerks in jedem erreichbaren Knoten. Der Routingalgorithmus arbeitet folgendermassen:

- Nachbarn werden mittels eines HALLO-Paketes ermittelt.
- ECHO-Pakete werden versendet um relevante Daten wie Antwortzeit und Entfernung zu analysieren und die daraus entstehenden Kosten für diese Verbindungsgüte zu ermitteln.
- LINK-STATE-Pakete mit allen Daten der gesammelten Netztopologie werden erstellt und periodisch geflutet.

- Berechnung der küzesten Route zwischen den Knoten mittels eines kürzesten-Wege-Algorithmus wie dem Algorithmus von Dijkstra oder Bellman-Ford.

Falls dies Routingtabelle oft geändert wird liegt es nahe das LSR-Protokoll zu verwenden, da so schnelle Reaktionen auf eventuelle Topologieänderungen möglich sind, da die kürzesten Wege nach Aktualisierung der Tabellen immer wieder neu berechnet werden.

2.2 Dynamic Source Routing Protocol

Das letztendlich von mir implementierte Routingprotokoll DSR wird in multi-hop wireless AdHoc-Netwerken mit Sensorknoten verwendet. Das Netzwerk kann sich vollständig selbst organisieren und bedarf keiner gesonderten Überwachung.
Die beiden Hauptarbeitsweisen des Protokolls bestehen aus dem Ermitteln und Warten einer bestehenden Route zwischen den Sensorknoten im Netzwerk. Die einzelnen Knoten auf einer Route müssen nicht die komplette Netzwerktopologie kennen um zum nächsten Knoten zu gelangen, sondern es werden Mengen von Zieladressen in jedes Paket gespeichert und eventuell weitergeleitet oder modifiziert und gespeichert. Jeder Knoten horcht den Netzwerkverkehr ab und aktualisiert seine Routingeinträge.

3 Implementierung

3.1 verwendete Datenstruktur

Wie vorgeschlagen besteht die Datenstruktur aus drei elementaren Datenpaketen. Den Daten-, Anfrage- und Antwortpaketen die hin- und hergeschickt werden. Die Anfrage- und Antwortpakete bestehen jeweils aus dem Header, wo die Informationen gespeichert werden, einem Start und Ziel der Route (Source und Target), sowie einem Identifikator für das gerade bearbeitete Paket.
Ein Anfragepaket wird versendet und man erhält ein Antwortpaket, falls die Route erfolgreich aufgebaut wurde.
Des weiteren wird ein Datenpaket verwendet, indem die Informationen über den gewählten Pfad stehen. Dieser besteht aus dem payload und dem Datenheader.

3.2 Übermittlung und Cache

Bei der Übermittlung von Daten wird zunächst ein Anfragepaket erstellt, mit dem der Pfad zum eigentlich Ziel ermittelt wird. Dieses Paket wird per Broadcast versendet. Knoten, die zum ersten mal auf einer Route liegen, werden in den Header aufgenommen. Bereits besuchte Knoten werden nicht gespeichert, sondern übergangen, da auf einem Pfad kein Knoten mehrmals vorkommen darf. Pfade enthalten per Definition keine Kreise. Ist die Route zum Zielknoten nun erfolgreich aufgebaut worden, wird ein Antwortpaket versendet, das auf der zuvor ermittelten Route des Anfragepaketes, den ursprüngichen Startknoten findet. Ist das

Antwortpaket erfolgreich zurückgekommen wird das eigentliche Datenpaket mit den Nutzdaten versandt. Da sich im Test erwiesen hat, dass Datenpakete öfter mal verlorengehen, wird das gleiche Datenpaket höchstens 5 mal verschickt. Falls ein Datenpaket nach den fünf Versuchen nicht übermittelt werden konnte wird es verworfen. Eine Überprüfung des Zustandes des lokalen Speichers erfolgt in festen Abständen und sichert somit ein Überlaufen des Cache durch sequentielle Freigabe der Ressourcen.

4 Ausnahmen

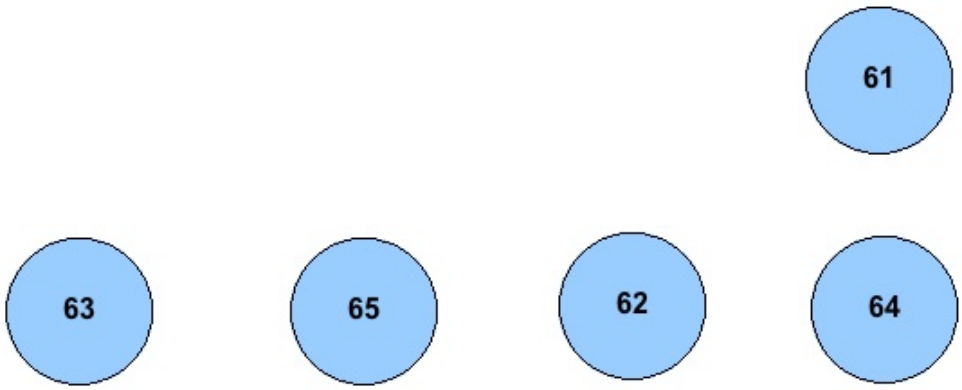
Anmerkung: Auf Grund des Wechsel der Routingverfahren von LSR zu DSR mitten im Semester, habe ich mich auf die minimale Fuktionsweise des Routing beschränkt.

- Pfade, die während der Antwortphase nicht mehr existieren werden nicht erkannt und somit wird auch nicht darauf reagiert. D.h. es gehen u.U. Pakete verloren auch nach mehrmaligen Sendeversuchen.
- Nachbarknoten werden nicht informiert über eventuell nicht mehr vorhandene Knotenverbindungen.
- Eine Verbindungsgüte zwischen den einzelnen Knoten wird nicht gespeichert und ausgewertet.
- Ein Clustering der Knoten zu Knotengruppen findet nicht statt.
- Es gibt keinen globalen Cache der die bereits vorhandenen Routen zwischenspeichert.

5 Versuchsauswertung

Das Testen fand in verschiedenen Umgebungen statt. Hauptsächlich habe ich in einem leerstehenden Seminarraum (ca. 7 x 7 m) mit wenig Mobiliar getestet. Weitere Tests wurden in einem vollgestellten Keller durchgeführt (ca. 12 x 4 m), jedoch vorzeitig abgebrochen.
Die insgesamt drei Testaufbauten decken die verschiedenen Netztopologien ab.

- Abbildung 1 zeigt einen L-förmigen Pfad.

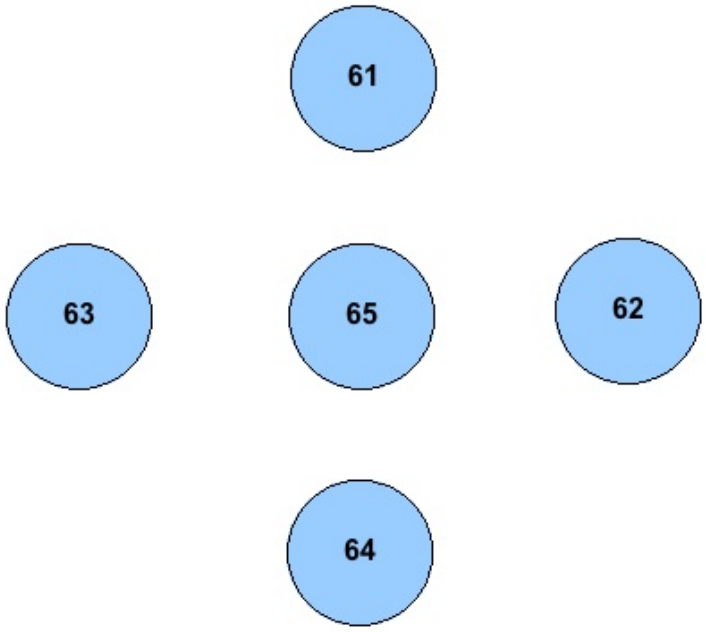


Auf dieser Topologie wurden zwei verschiedene Versuche ausgeführt. Versuchsaufbau 1 bestand darin Pakete von Knoten 61 nach 64 zu schicken. Bei Versuchsaufbau 2 wurden Pakete von Knoten 61 zum Knoten 63 verschickt.
Es wurden pro Versuchsaufbau 3 Versuche mit jeweils 10 Paketen versendet mit folgendem Ergebnis:

no.	Versuch 1	Versuch 2
1	7	3
2	10	4
3	7	4

Beim ersten Versuch konnte man eine Erfolgsquote von 80% beobachten.
Beim zweiten Versuch kann man kaum von einer „Erfolgs“quote sprechen. Hier habe ich knapp 40% gemessen.

- Abbildung 2 zeigt einen stern-/kreuzförmigen Aufbau.
  Auf dieser Topologie wurde ein Versuch ausgeführt. Hier wurden Pakete von Knoten 63 zu 62 verschickt, wobei fast jeder Knoten jeden anderen in seiner Reichweite hat. Lediglich Knoten 64 hatte keine direkte Verbindung zum Knoten 61. Eine direkte Verbindung von Knoten 61 zu 64 bestand jedoch!

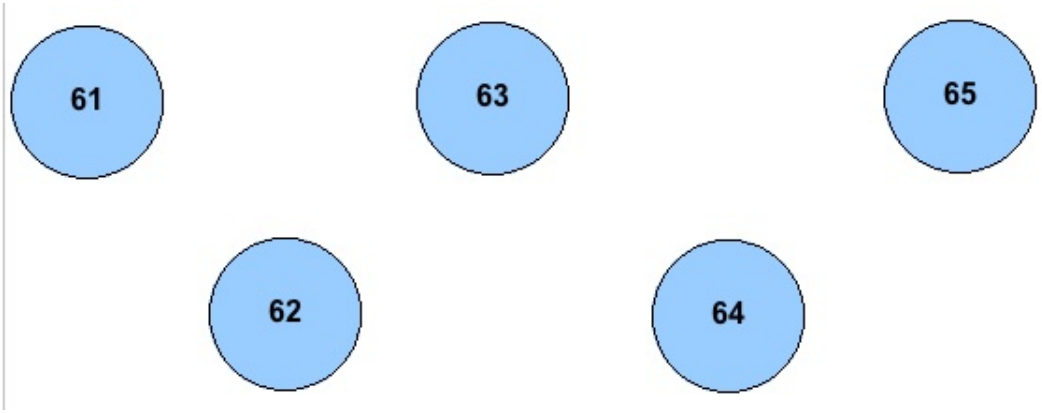


Es wurden 3 Versuche mit jeweils 10 Paketen versendet mit folgendem Ergebnis:

no.	Versuch
1	11
2	10
3	9

Eine Erfolgsquote von 100% ist zu beobachten, was höchstwahrscheinlich darauf zurückzuführen ist, dass Pakete kopiert und mehrfach versendet wurden.

- Abbildung 3 zeigt einen alternierenden Pfad.
  Auf der dritten Topologie wurde ein Versuch ausgeführt. Hier wurden Pakete von 61 nach 65 verschickt, wobei jeder Knoten nur seine direkten Nachbarn auf dem alternierenden Pfad kannte.



Es wurden 5 Versuche mit jeweils 20 Paketen versendet mit folgendem Ergebnis:

no.	Versuch
1	9
2	0 !?!?
3	8
4	7
5	7

Auch hier kann man kaum von einer „Erfolgs“quote sprechen. Ich habe 31% Durchsatz gemessen. Diesen schlechten Wert kann ich nur auf Implementierungsfehler meinerseits zurückführen!

6 Schlusswort

Auf Grund der Tatsache, dass ich mitten im Semester das von mir gewählte Routingverfahren gewechselt habe und an einen Zeitpunkt angekommen bin, wo ich die Implementierung und Fehlersuche eingestellt habe, um mit diesem Technical Report zu beginnen, sind sicherlich noch etliche Tweaks unaufgedeckt, die zu einem gesteigerten Durchsatz geführt hätten.