

ServiceCast: Eine Architektur zur dienstorientierten Kommunikation in selbstorganisierenden Sensor-Aktor-Netzen

Andreas L. Kuntz
Institut für Telematik
Universität Karlsruhe (TH)
akuntz@tm.uka.de

Martina Zitterbart
Institut für Telematik
Universität Karlsruhe (TH)
zit@tm.uka.de

ABSTRACT

ServiceCast ist eine Architektur zur dienstorientierten Kommunikation in selbstorganisierenden Sensor-Aktor-Netzen (SSAN). Ziel ist es Methoden der Dienstorientierung auf niederen Protokollschichten verfügbar zu machen, um im gesamten Stack einem einheitlichen Kommunikations-Paradigma folgen zu können. Als zentrales Merkmal werden *Dienste* statt Knoten als Grundlage der Adressierung und des Routings betrachtet. Die folgenden von ServiceCast behandelten Aspekte sind für die Kommunikation in Selbstorganisierenden Sensor-Aktor-Netzen essentiell und daher von besonderem Interesse: Dienstorientierte Adressierung, Auffinden von Diensten, Verwaltung von Verbindungen sowie die verteilte Konfiguration und Initialisierung der Dienstinstanzen im Netz.

1. EINLEITUNG

Sensornetze werden üblicherweise als aus vielen kleinen, ressourcenarmen Knoten bestehend angenommen. Die Knoten erbringen dabei Dienste, welche sie anderen Knoten zur Verfügung stellen. Aufgrund der häufig beschränkten Energieressourcen, wird in vielen Arbeiten davon ausgegangen, dass Knoten temporär in Energiesparmodi wechseln, wodurch auch die vom Knoten erbrachten Dienste zeitweise nicht verfügbar sind. Daneben sind zahlreiche weitere Ursachen denkbar, welche zu Dynamik und damit zu ähnlichen Auswirkungen führen. Wir gehen davon aus, dass ein Großteil der Dienste redundant im Netz vorhanden ist, also mehrere Instanzen desselben Dienstes auf unterschiedlichen Knoten zur Verfügung stehen. Durch geeignete Abstraktion der Adressierung der Kommunikationspartner möchten wir diese Redundanz ausnutzen, um Nachrichten transparent an aktive Dienstbringer auszuliefern und damit die Verfügbarkeit der erbrachten Dienste zu erhöhen. Wir halten Dienstorientierung auf niederen Protokollschichten für einen vielversprechenden Ansatz, um eine geeignete Abstraktion zu realisieren.

Aufgrund der Annahme, dass Dienste durch mehrere Instanzen redundant vertreten sind, kann der Dienstbezeichner alleine den Kommunikationspartner nicht eindeutig identifizieren. Daher führt ServiceCast als orthogonales Merkmal Kontexte ein, welche Sensorknoten ihrer Umgebung zuordnen. Kontexte erlauben, das „wo“ in einer vom Benutzer beeinflussbaren Granularität zu definieren (z. B. „Lokation innerhalb eines Rechtecks mit den Eckpunkten (\vec{x}, \vec{y}) “), während der Dienstbezeichner das „was“ definiert (z. B.

„Temperaturdienst“).

Anstatt Knoten und Dienstinstanzen direkt zu adressieren, werden Nachrichten mit einem Ziel-Kontextbereich und einem Ziel-Dienstbezeichner versehen. Aufsuchen des Kontextes und Auswahl der Dienstinstanz werden vom Netz übernommen. So ist weder explizit das Übersetzen von Dienstbezeichner auf die ID des dienstbringenden Knotens notwendig, noch muss auf temporäre Nicht-Erreichbarkeit eines Dienstes Rücksicht genommen werden.

Architekturen zur Dienstfindung wie [3, 4, 5, 9] stützen sich auf Dienstverzeichnisse. Das Dienstverzeichnis bildet Dienstbezeichner auf die Adressen der Wirtsknoten ab, welche als Grundlage der späteren Kommunikationsbeziehung dienen. ServiceCast kommt ohne ein solches Verzeichnis aus. D. h. Dienstbezeichner müssen nicht in eine netzweit gültige Knotenadresse umgesetzt werden.

Im Folgenden wird der Ansatz anhand eines kurzen Beispiels erläutert. Auf Kontexte, Dienstfindung und Instanzauswahl wird anschließend eingegangen. Aspekte des Routings bezüglich der Kontexte sollen hier nicht weiter vertieft werden, da für viele mögliche Kontexträume bereits geeignete, spezielle Ansätze existieren [1, 7, 8]. Das Anycast-ähnliche Adressierungsschema erfordert besonderes Augenmerk bei der Zustellung von Nachrichtenfolgen, wenn mehrere, zueinander in Beziehung stehende Nachrichten an dieselbe Dienstinstanz übermittelt werden müssen. Auf diesen Punkt wird in Abschnitt 2.4 kurz eingegangen. Abschließend werden Aspekte der verteilten Konfiguration, der Dynamik und der Sicherheit kurz angerissen. Eine umfassende Behandlung derselben ist für zukünftige Arbeiten vorgesehen.

2. KONZEPT

ServiceCast adressiert Kommunikationspartner durch die Kombination von Ziel-Kontextbereich und Ziel-Dienstbezeichner. So werden durch dieselbe Adresse potentiell alle gleichwertigen Dienstinstanzen innerhalb des angegebenen Kontextbereiches angesprochen. Die Instanzauswahl wird transparent vom Netz vorgenommen, was die effektive Verfügbarkeit des Dienstes verbessert. ServiceCast setzt keine netzweit eindeutigen Knoten-Identifikatoren voraus; einzig lokal eindeutige Identifikatoren sind notwendig, um Nachbarknoten voneinander unterscheiden zu können.

Nachrichten werden in zwei Schritten zur Ziel-Dienstinstanz

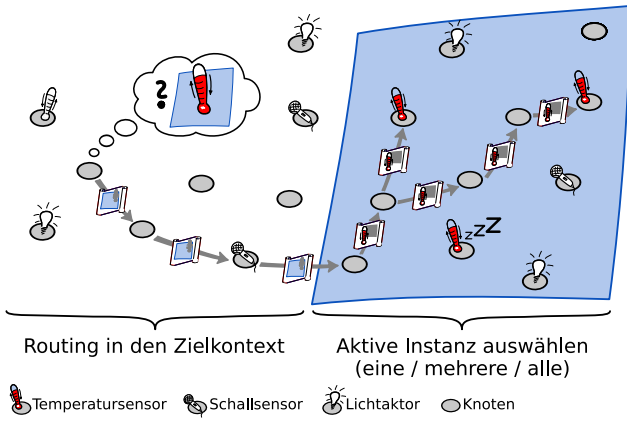


Figure 1: Adressierung und Routing mit ServiceCast

geleitet: (1) Routing der Nachricht in den Ziel-Kontext, (2) Auswahl der Ziel-Dienstinstanz. Abbildung 1 zeigt ein Beispielszenario: Sensorknoten sind durch Ovale angedeutet, Dienste durch die Piktogramme auf den Knoten. Das Beispiel-Netz bietet Temperatur- und Schallsensoren, sowie Lampen als Aktoren. Sensoren und Aktoren werden jeweils durch die zugehörigen Dienste gekapselt. Als Kontextrraum soll hier beispielhaft die Lokation der Knoten in einem zwei-dimensionalen Koordinatensystem dienen.

Möchte beispielsweise ein Knoten die Temperatursensoren innerhalb eines bestimmten Bereiches abfragen, versendet er eine Nachricht, in welcher der gewünschte Bereich als Ziel-Kontextbereich codiert ist. In der Abbildung ist dieser durch das graue Feld rechts gekennzeichnet. Als Ziel-Dienstbezeichner trägt der Knoten „Temperaturdienst“ in die Nachricht ein. Die Nachricht wird zunächst — ohne Berücksichtigung des Zieldienstes — in den „grauen“ Kontextbereich geroutet (Abbildung 1, links). Erst innerhalb des Ziel-Kontextbereiches wird der Ziel-Dienstbezeichner „Temperaturdienst“ ausgewertet und die Nachricht an die aktiven Instanzen des Temperaturdienstes geleitet (Abbildung 1, rechts). So kann z. B. die temporäre Nicht-Verfügbarkeit eines Dienstes, verursacht durch einen schlafenden Knoten, kompensiert werden. Ein Parameter in der Nachricht gibt an, wieviele der Instanzen die Nachricht erhalten sollen. Derzeit sind „genau eine“, „mehrere“ und „alle“ als mögliche Werte vorgesehen.

2.1 Kontexte

Sensoren beobachten Phänomene ihrer Umwelt. Gemessene Sensorwerte sind an die Lokation ihrer Messung gebunden und verlieren ohne diese Information an Aussagekraft. Ein Kontext ordnet einen Sensorknoten seiner Umgebung zu und soll den Begriff der Lokation eines Sensorknotens verallgemeinern bzw. von ihm abstrahieren. Kontexte können symbolisch, diskreter Natur sein (z. B. die Nummer eines Raumes innerhalb eines Gebäudes oder die symbolische Identität eines Patienten) oder auch kontinuierlich (z. B. eine geographische Koordinate). Kontexte sollen als abstraktes Konstrukt gesehen werden und sind nicht auf die hier angeführten Beispiele beschränkt. Anforderungen an Kontexträume sind im diskreten Fall zwei Kontexte auf Gleichheit prüfen zu können, im kontinuierlichen Fall der Vergleich bezüglich

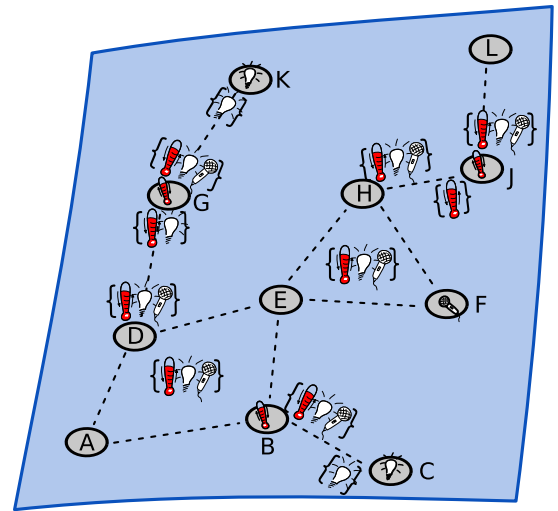


Figure 2: Trans Service Sets

einer zum Kontextrraum gehörenden Ordnungsrelation. Kontexte erhöhen einerseits die Spezifität einer Anfrage indem sie die Menge der adressierten Knoten einschränken. Andererseits erlauben sie die Kompensation der Auswirkungen von Dynamik, z. B. im Falle von schlafenden Knoten.

2.2 Auffinden von Diensten

Da ServiceCast Dienste als Grundlage des Routings verwendet, wird kein explizites Verzeichnis benötigt, welches Dienstbezeichner auf die IDs der dienstbringenden Knoten abbildet. Stattdessen werden Dienstinstanzen anhand von Dienstverfügbarkeitsinformationen gefunden. Diese werden periodisch zwischen benachbarten Knoten ausgetauscht. Während ein Knoten initial nur seine lokal angebotenen Dienste, das „Local Service Set“ (LSS) kennt, lernt er mit der Zeit welche Dienstmengen über seine Nachbarknoten direkt und indirekt verfügbar sind. Eine solche Dienstmenge wird als „Trans Service Set“ (TSS) bezeichnet. TSSs lassen sich als Erreichbarkeitsrelation wie folgt definieren: „Das $TSS(x, y)$ ist die Menge aller über einen Knoten x erreichbaren Dienste aus Sicht des Nachbarknotens y .“ TSSs sind also immer für ein Paar von benachbarten Knoten definiert. Genauer:

$$TSS(x, y) := \left(\bigcup_{\substack{n \in \mathcal{N}(x) \\ n \neq y}} TSS(n, x) \right) \cup LSS(x),$$

wobei $\mathcal{N}(x)$ die Menge aller zu x benachbarten Knoten ist.

Abbildung 2 verdeutlicht dies an einem Beispiel. Die Kanten zwischen den Knoten deuten die Nachbarschaftsbeziehungen an. Die Dienstsymbole in geschweiften Klammern repräsentieren das jeweilige TSS, also die Menge der Dienste, welche der Knoten über die entsprechende Verbindung seinem Nachbar verfügbar macht. In diesem Beispiel wurden bidirektionale Kanten angenommen. Ferner wurden die Knoten durch Buchstaben mit Namen versehen. Es sei hier noch einmal ausdrücklich erwähnt, dass die eindeutige Unterscheidbarkeit der Knoten keine für ServiceCast notwendige Voraussetzung ist, sondern hier nur der einfacheren

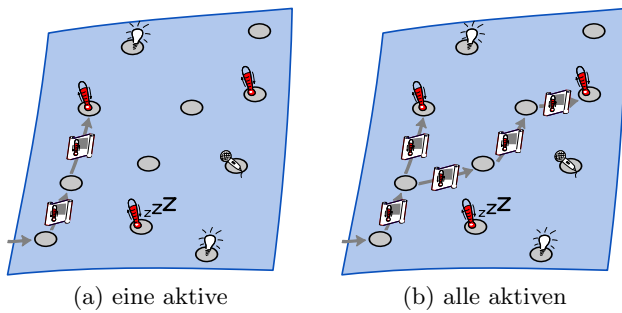


Figure 3: Auswahl (a) einer bzw. (b) aller aktiven Instanzen des Temperaturdienstes

Beschreibung dient. ServiceCast kommt mit lokal eindeutigen Knoten-Identifikatoren aus. Innerhalb von Zyklen (z. B. ABDE oder EFH) steht an jeder Kante das selbe TSS. Vereinfachend ist dieses nur einmal in der Mitte des entsprechenden Zyklus eingetragen. Anders verhalten sich die TSSs am Rand des Netzes. Knoten K ist beispielsweise über Knoten G mit dem restlichen Netz verbunden. So enthält hier das TSS(G, K) alle im Netz verfügbaren Dienste, da K jeden einzelnen über G erreichen kann. Umgekehrt enthält TSS(G, D) nur die von G und K angebotenen Dienste.

Das Beispiel macht deutlich, dass in stark vermaschten Netzen TSSs alleine nur wenig Information zur Routingentscheidung beitragen können. Denn TSSs spiegeln die Erreichbarkeit von Diensten wider und codieren inhärent die Menge aller alternativen Pfade zu jedem der Dienste. TSSs repräsentieren also die eingangs erwähnte Redundanz der Dienste. Um sinnvolle Entscheidungen bezüglich des Routings treffen zu können, wird das Einbeziehen zusätzlicher Informationen wie z. B. die „Entfernung zur nächsten Dienst-Instanz“, Informationen über „Multiplizitäten“ oder über die Energiereserven des Wirtsknoten notwendig. Untersuchung und Evaluation der Einsatzfähigkeit und des Nutzens solcher Zusatzinformationen, sowie die Integration in ServiceCast ist für weiterführende Arbeiten geplant.

TSSs lassen sich z. B. mittels Bloomfilter [2] kompakt, und mit konstantem Speicherbedarf verwalten. Da für jeden Nachbarknoten exakt ein TSS gepflegt werden muss, wächst der gesamte Speicherbedarf linear mit der Anzahl der direkten Nachbarn pro Knoten (bzw. mit der Dichte der Knoten) und mit der maximalen Anzahl der Dienste. Wir erwarten, diese Anzahl für viele Applikationen einheitlich nach oben abschätzen zu können.

2.3 Dienstinstanz Auswahl

Innerhalb des Ziel-Kontextbereiches sind gleichwertige Dienstinstanzen austauschbar, falls (1) die Kommunikation mit einem Dienst aus einer einzigen Nachricht besteht oder (2) während einer längeren Verbindung zu einem Dienst kein Zustand innerhalb des Dienstes aufgebaut wird oder (3) der Zustand vollständig auf andere Dienstinstanzen repliziert werden kann. Die Behandlung von Nachrichtenfolgen, in welchen die einzelnen Nachrichten zueinander in Beziehung stehen, wird im nächsten Abschnitt besprochen. Ein Parameter in der Nachricht gibt an, wieviele der aktiven

Instanzen die Nachricht erhalten sollen. Derzeit vorgesehen sind die Werte: „genau eine“ ($=1$), „mehrere“ (≥ 0) und „alle“ (all). Der Wert „genau eine“ realisiert einen dienst- und kontextspezifischen Anycast, d. h. es wird eine beliebige, aktive Instanz des Ziel-Dienstes angesprochen, welche sich im Ziel-Kontextbereich befindet (siehe Abbildung 3(a)). Da hier implizit eine Garantie („genau eine“) ausgesprochen wurde, ist es notwendig für diese Art der Zustellung in den weiterleitenden Knoten temporär Zustand zu halten. Der Wert „mehrere“ spricht beliebige aktive Instanzen des Ziel-Dienstes innerhalb des Ziel-Kontextbereiches an. Undefiniert viele weitere Instanzen im Zielkontext können die Nachricht ebenfalls erhalten. Ein dienst- und kontextspezifischer Broadcast wird durch den Wert „alle“ bereit gestellt, welcher die Nachricht an alle aktiven Instanzen des Ziel-Dienstes innerhalb des Kontextbereiches übermitteln lässt (siehe Abbildung 3(b)).

2.4 Verbindungsorientierte Kommunikation

Obwohl angenommen werden kann, dass ein großer Teil des Datenaufkommens in Sensornetzen in oben beschriebener Weise behandelbar ist, sind Fälle denkbar in denen es notwendig sein wird den Kommunikationspartner temporär zu fixieren, also eine länger dauernde Kommunikationsverbindung mit derselben Instanz eines Dienstes (oder Gruppe von Instanzen) zu ermöglichen. Insbesondere im Umgang mit Aktoren kann dies wichtig sein. Für diesen Fall sollen — ähnlich der in [6] vorgeschlagenen Architektur — sogenannte Pseudo-Dienste eingeführt werden (in [6] „Transaction Identifier“ genannt). Pseudo-Dienste tragen probabilistisch und temporär netzweit eindeutige Dienstbezeichner. Sie werden bei Bedarf dynamisch instantiiert und markieren den Pfad, welchen die Nachricht durch das Netz genommen hat. Folgenachrichten enthalten den Pseudo-Dienst als Zieldienst, wodurch sie zu den selben Ziel-Dienstinstanzen wie die initiale Nachricht gelangen. So kann ein einzelner oder eine Gruppe von Kommunikationspartnern temporär fixiert werden. Die Pseudo-Dienste etablieren einen Softstate in den Zwischenknoten, welcher nach Nichtbenutzung der Verbindung wieder abgebaut wird.

2.5 Verteilte Konfiguration

Knoten sollen zunächst in die Zielregion ausgebracht und erst anschließend konfiguriert bzw. programmiert werden. Aus energietechnischen Gründen kann es sinnvoll sein, Sensorknoten von schon initial mit den applikationsspezifischen, zur Hardware passenden Diensten auszustatten, aber erst während der Konfigurationsphase zu entscheiden, welche von ihnen gestartet werden. Während der Konfigurationsphase soll durch lokale Kommunikation benachbarter Knoten entschieden werden, welche Dienste auf einem Knoten zur Verfügung gestellt werden. Dabei sollen Parameter wie die Anzahl der auf (möglicherweise indirekt) benachbarten Knoten verfügbaren Instanzen eines Dienstes oder die Lage des Knotens innerhalb des Kontextraumes eine Rolle spielen. Denkbar wäre z. B. eine gleichmäßige Verteilung von Dienstinstanzen bei einer vorgegebenen mittleren Dichte. Genau so sind andersartige Verteilungen (z. B. Gauss, etc.) als Vorgabe vorstellbar.

2.6 Dynamik

Neben Knoten-Mobilität sind Reprogrammierung oder Rekonfiguration von Knoten, Ausbringen neuer Knoten

bzw. deren Wegfall, Kommunikationsstörungen und temporäre Unerreichbarkeit aufgrund von Energiesparmaßnahmen wichtige Ursachen für Dynamik. Allen Ursachen gemeinsam ist eine Veränderung der Topologie des Netzes, welche Auswirkungen auf Routing, das Auffinden von Diensten und andere Funktionalitäten haben kann. Durch die mit ServiceCast eingeführte abstrakte, topologiefreie Adressierung durch Dienstbezeichner und Kontexte, lassen sich Auswirkungen von Dynamik transparent kompensieren. Denn anstatt eine spezielle Dienstinstanz (oder einen Knoten) als Kommunikationspartner zu wählen, wird durch den Dienstbezeichner nur die Art des Dienstes spezifiziert, die Auswahl der Instanz(en) bleibt dem Netz überlassen.

2.7 Sicherheit

Zur Realisierung von Sicherheitszielen wie Integrität, Vertraulichkeit, Authentizität, Autorisation, etc. sind Sicherheitsprimitive wie Hashing, Verschlüsselung und ein Vertrauensanker notwendig. Kritisch ist in diesem Falle der Vertrauensanker zu betrachten. Ist er klassisch fest einem Knoten zugeordnet, muss für ServiceCast ein neuer Ansatz gefunden werden, da Knoten hier als austauschbar, anonym und frei von jeglichen fixen IDs angenommen werden. Darüber hinaus sollen Dienste unabhängig vom Wirtsknoten, also verlagerbar sein, was ebenfalls gegen einen an den Knoten gebundenen Vertrauensanker spricht.

Der Weg, welcher mit ServiceCast beschritten werden soll, ist Sicherheit als dienstbasiertes Konzept zu realisieren. Anstatt z.B. einen Knoten zu authentifizieren, gilt es die Integrität eines Dienstes zu überprüfen. Andere Ziele sind analog auf Dienste zu übertragen.

3. ZUSAMMENFASSUNG

Mit ServiceCast wurde ein Ansatz vorgestellt, welcher Methoden der Dienstorientierung auf niederen Protokollschichten verfügbar macht. Die Adressierung über Dienste, anstatt durch knotenspezifische Adressen, erlaubt eine Abstraktion der Kommunikation von der Topologie des Netzes. Dienstinstanzen werden innerhalb des Zielkontextbereiches lokal vom Netz ausgewählt, wodurch Dynamik-Auswirkungen für den Benutzer transparent kompensiert werden. Dadurch erwarten wir ein verbessertes Verhalten bezüglich der Verfügbarkeit von Diensten in dynamischen Netzen. ServiceCast kommt ohne explizites Dienstverzeichnis aus, was einen im Vergleich zu anderen Architekturen geringeren Protokoll-overhead erwarten lässt. Darüber hinaus verlangt ServiceCast keine fixen, netzweit eindeutigen Knotenidentifikatoren, wodurch Knoten austauschbar und anonym werden.

4. DANKSAGUNG

Dank gilt es meinem geduldigen Kollegen Hans-Joachim Hof zu sagen für viele fruchtbare Diskussionen.

Diese Arbeit wurde im Rahmen des DFG-Graduiertenkollegs 1194 „Selbstorganisierende Sensor-Aktor-Netzwerke“ durch die Deutsche Forschungsgemeinschaft gefördert.

5. REFERENCES

- [1] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international*

- conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.
- [2] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.
- [3] K. Edwards and T. Rodden. *Jini Example by Example*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2001.
- [4] Hans-Joachim Hof and Martina Zitterbart. SCAN: A secure service directory for service-centric wireless sensor networks. *Computer Communications*, 28(13):1517–1522, Aug. 2005. ISSN 0140-3664.
- [5] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM Press.
- [6] Jeremy Elson and Deborah Estrin. Random, Ephemeral Transaction Identifiers in Dynamic Sensor Networks. In *Proceedings of the Twenty First International Conference on Distributed Computing Systems (ICDCS-21)*, pages 459 – 468, Phoenix, Arizona, April 2001.
- [7] Matthias Gauger, Pedro José Marrón, Marcus Handte, and Kurt Rothermel. Routing in Sensor Networks based on Symbolic Coordinates. In P. Marrón, editor, *5. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze"*, Technischer Bericht 2006/07, pages 81–86. Stuttgart: Universität Stuttgart, Juli 2006.
- [8] M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, Nov.-Dec. 2001.
- [9] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM Press.