

# Dienstplatzierung in Ad-hoc-Netzen

Georg Wittenburg, M.Sc.  
Freie Universität Berlin

- Ablauf- und Planungsforschung
  - Einführende Beispiele
  - Facility Location
  - Dienstplatzierung in Ad-hoc-Netzen
- Das SPi-Dienstplatzierungssystem
  - Rahmenwerk und Platzierungsalgorithmen
  - Ergebnisse der experimentellen Evaluation
  - Bewertung und Zusammenfassung

# Der richtige Ort



Photo by: Exothermic on flicker



© Copyright Social Media Group



© Christian und Andreas Bergel



© 2006-2008, Patterson Warehouse, Inc.



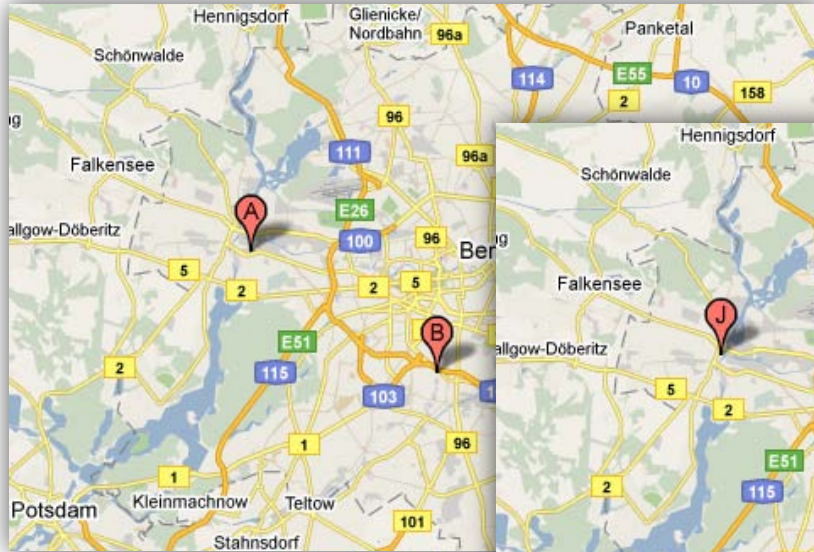
© 2010 MKU.de Neue Medien



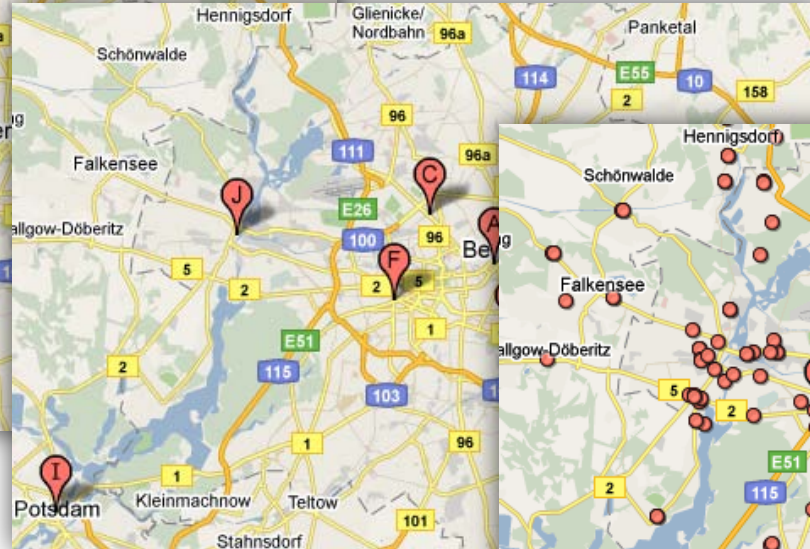
© 1999 - 2010 Inter IKEA Systems B.V.

➤ Wo ist der geeignetste Ort für ein neues Logistikzentrum, eine neue Fabrik oder ein neues Warenhaus?

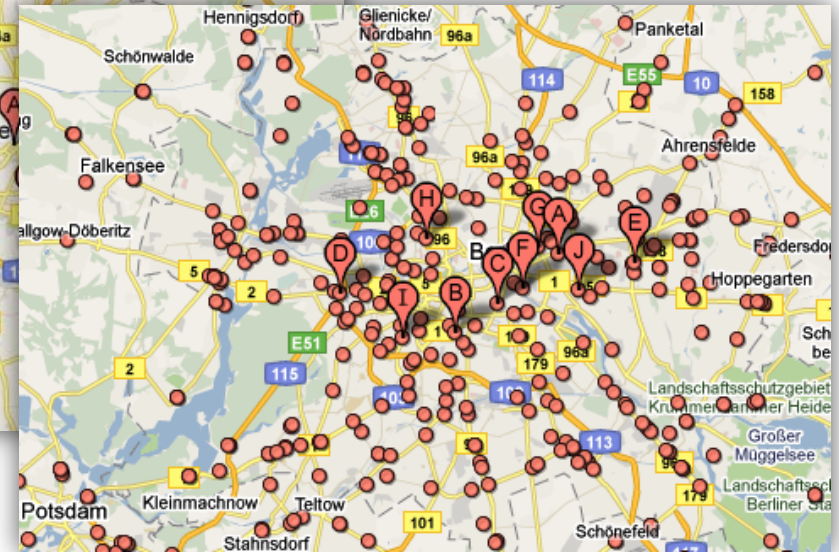
# Die richtige Anzahl



IKEA in Berlin



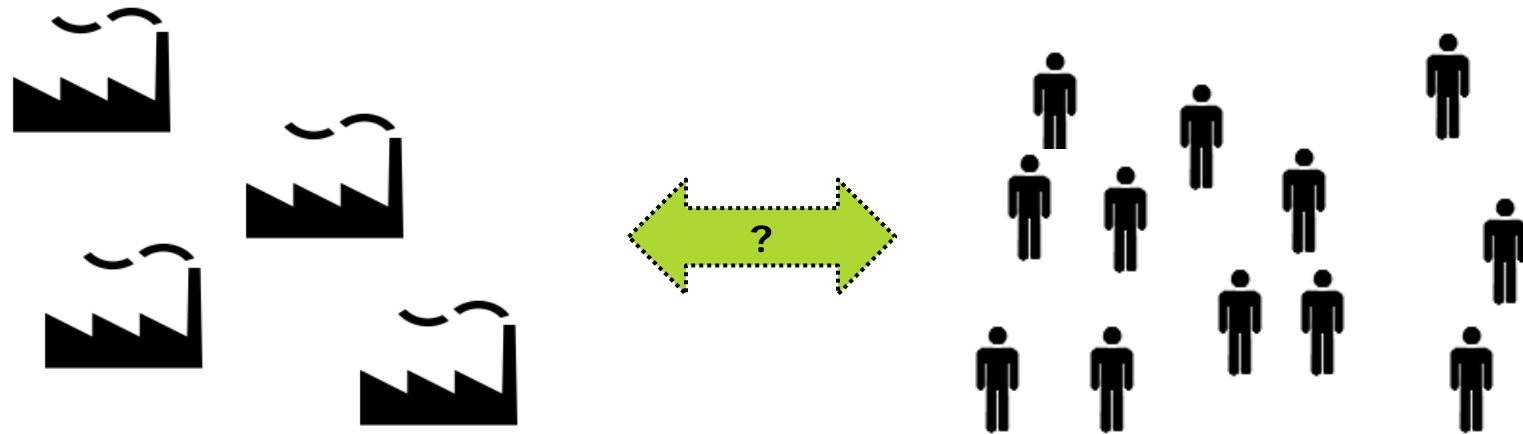
Karstadt / Kaufhof  
in Berlin



Tankstellen in Berlin

- Wie viele Filialen sollten in einer Region eröffnet werden?
- In Anbetracht der Nachfrage, der Logistik, der Konkurrenz, ...

- Ablauf- und Planungsforschung (engl. Operations Research)
  - Projektplanung (z.B. Methode des kritischen Pfades)
  - Floorplanning (z.B. Chipentwurf)
  - Netzoptimierung (z.B. in Telekommunikationsnetzen)
- Facility Location Theory
  - $p$ -Median Problem – Menge der  $p$  Knoten eines Graphen, die Distanz zu anderen Knoten minimiert
  - (Uncapacitated) Facility Location Problem – Auswahl aus Menge von möglichen Versorgungsstandorten, die Summe aus Transport- und Standortkosten minimiert



- Gegeben:
  - Menge potentieller Standorte  $F$
  - Menge von Kunden  $\mathcal{C}$ , deren Nachfrage zu decken ist
  - Kosten für Befriedigung der Nachfrage  $c_{ij}$  von Kunde  $i \in \mathcal{C}$  durch Standort  $j \in F$
  - Einmalige Kosten  $f_j$  für Eröffnung von Standort  $j \in F$
- Gesucht:
  - Menge der zu eröffnenden Standorte  $F' \subseteq F$
  - Zuordnung der Kunden auf geöffnete Standorte  $\phi : \mathcal{C} \rightarrow F'$  unter Minimierung der Gesamtkosten



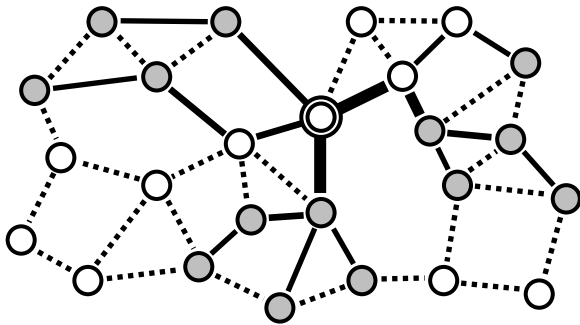
- Ad-hoc-Netze setzen Kooperationsbereitschaft zwischen teilnehmenden Knoten voraus



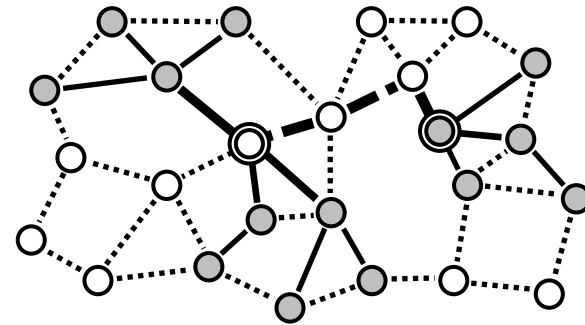
- Verteilte Dienstleistung – Aufteilung der Serverkomponente eines Dienstes in mehrere *Dienstinstanzen*



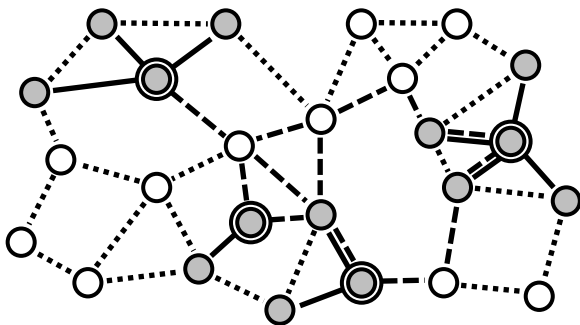




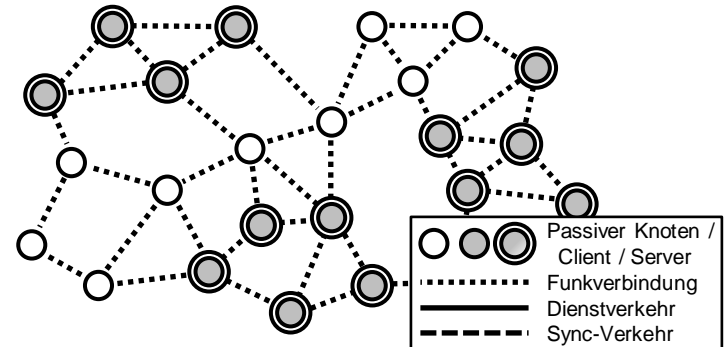
- 1. Beispiel: Transaktionale Datenbank
- Eine einzelne Dienstinstanz



- 2. Beispiel: Verzeichnisdienst
- Mehrere Dienstinstanzen



- 3. Beispiel: Web-Server
- Viele Instanzen pro Nachfragecluster



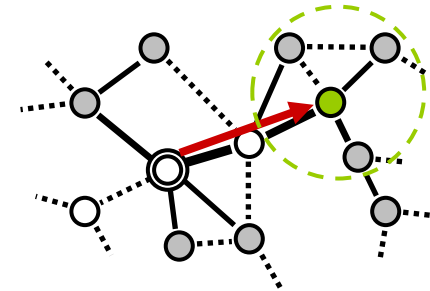
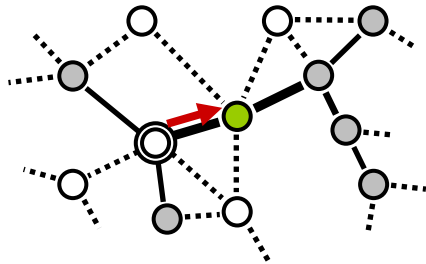
- 4. Beispiel: Taschenrechner
- Eine Dienstinstanz pro Knoten

➤ Berücksichtigung des Synchronisationsbedarfs zwischen Dienstinstanzen bei Berechnung der Dienstkonfiguration erforderlich

- ***Dienstplatzierung*** – Auswahl einer optimalen Menge von dienst anbietenden Knoten angesichts
  - 1) derzeitige Nachfrage nach dem Dienst
  - 2) aktuelle Netztopologie
- Unterprobleme:
  - *Wo*, d.h. auf welchen Knoten, sollen Dienstinstanzen angeboten werden?
  - *Wie viele* Dienstinstanzen sind für die optimale Funktion des Gesamtnetzes erforderlich?
  - *Wann* sollte eine Dienstkonfiguration angepasst werden?

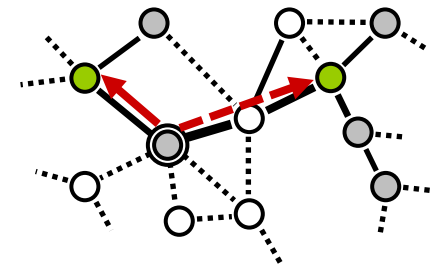
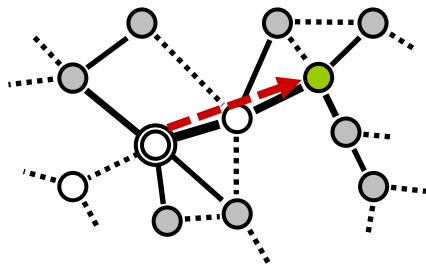
- Für zentralisierte Dienste mit einer einzigen Dienstinstanz:

- Migration zum Nachbarknoten, der mehr als die Hälfte aller Anfragen weiterleitet [1]
- Migration zum Cluster mit dem höchsten aggregierten Anfragevolumen [2]



- Für verteilte Dienste mit einer variablen Anzahl von Dienstinstanzen:

- Replikation der Instanz unter Maximierung der Anzahl der neuen Dienstnehmer [3]
- Migration/Replikation zu Knoten, die größtes Anfragevolumen weiterleiten/verursachen [4]



[1] K. Oikonomou and I. Stavrakakis. Scalable Service Migration: The Tree Topology Case. In *5th Annual Mediterranean Ad Hoc Networking Workshop*, Lipari, Italy, June 2006.

[2] H. Liu, T. Roeder, K. Walsh, R. Barr, and E. G. Sirer. Design and Implementation of a Single System Image Operating System for Ad Hoc Networks. In *3rd Intl. Conf. on Mobile Systems, Applications, and Services*, Seattle, USA, June 2005.

[3] F. Sailhan and V. Issarny. Scalable Service Discovery for MANET. In *3rd IEEE Intl. Conf. on Pervasive Computing and Communications*, Kauai, USA, March 2005.

[4] Klaus Herrmann. Self-Organizing Infrastructures for Ambient Services. PhD thesis, Berlin University of Technology, Berlin, Germany, July 2006.

## Fest vorgegebener Koordinator

- Zentraler, fest vorgegebener Knoten steuert Dienstkonfiguration
- + Optimale Dienstkonfiguration
- Mittlerer Signalaufwand
- Erfordert Infrastruktur, folglich bedingt geeignet für Ad-hoc-Netze

## Dynamischer Koordinator

- Dynamisch ausgewählter Knoten steuert globale Dienstkonfiguration
- + Optimale Dienstkonfiguration
- Mittlerer Signalaufwand

## Verteilte Konsensfindung

- Dienstbietende Knoten koordinieren verteilt die Dienstkonfiguration
- + Optimale Dienstkonfiguration
- Hoher Signalaufwand

## Lokale Entscheidungen

- Dienstbietende Knoten treffen lokal Platzierungsentscheidungen
- + Geringer Signalaufwand
- Suboptimale Dienstkonfiguration

zentralisiert



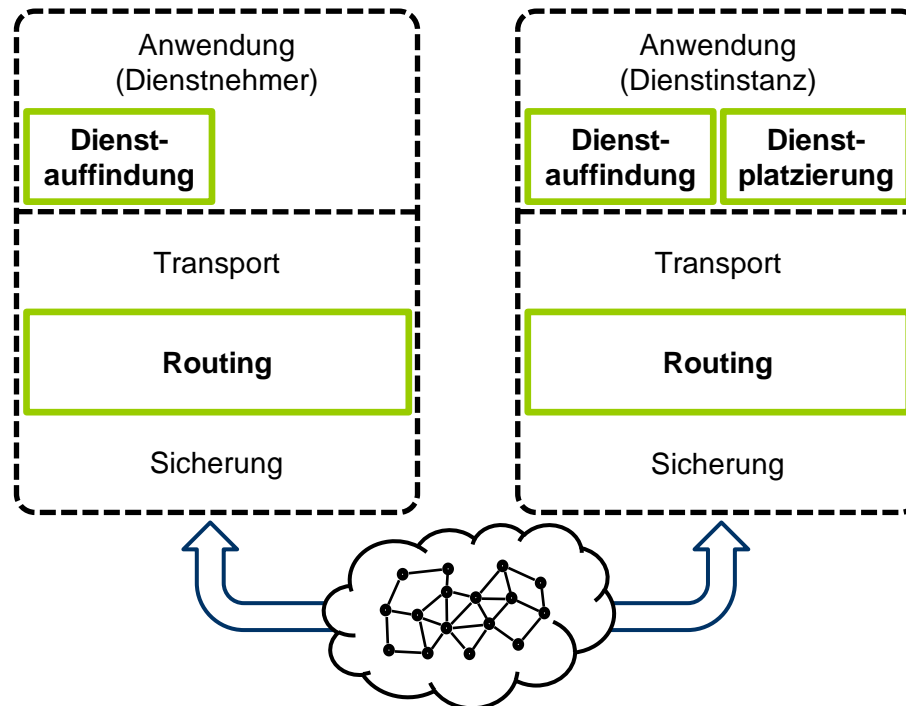
verteilt

- Zusammenfassung aktueller Ansätze in: G. Wittenburg and J. Schiller. A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks. In *Proc. of PerCom '08*, Hong Kong, China, March 2008.

- Ablauf- und Planungsforschung
  - Einführende Beispiele
  - Facility Location
  - Dienstplatzierung in Ad-hoc-Netzen
- Das SPi-Dienstplatzierungssystem
  - Rahmenwerk und Platzierungsalgorithmen
  - Ergebnisse der experimentellen Evaluation
  - Bewertung und Zusammenfassung

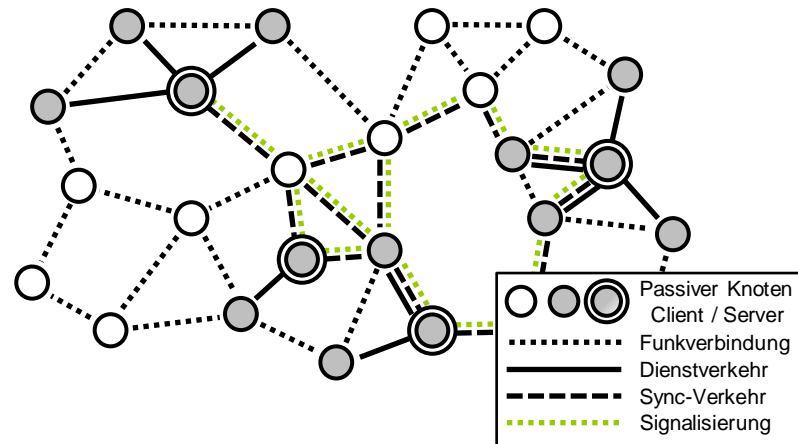
- **SPi-Rahmenwerk zur Dienstplatzierung**
  - Basisfunktionalität für Dienstplatzierung
  - Ermöglicht Vergleich von Dienstplatzierungsalgorithmen
- **Graph Cost / Single Instance (GCSI)** für zentralisierte Dienste (mit einer Dienstinstanz)
- **Graph Cost / Multiple Instances (GCMI)** für verteilte Dienste (mit variabler Anzahl von Instanzen)
  - Optimierung der Dienstkonfiguration basierend auf Daten bezüglich Dienstonutzung und Netztopologie
  - Berücksichtigung von Kommunikation zwischen Dienstinstanzen
  - Abwägung der Adaptationskosten gegen Qualitätssteigerung der neuen Dienstkonfiguration





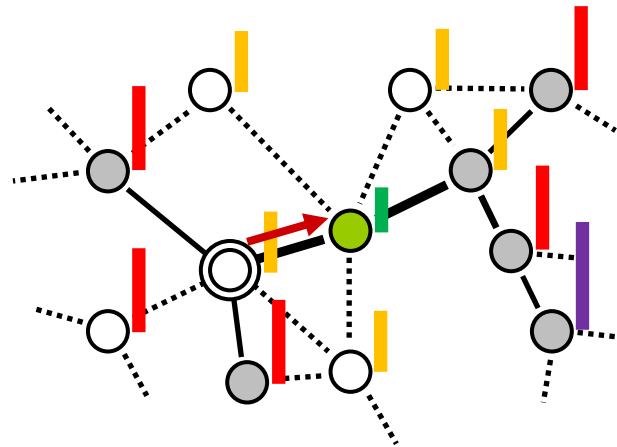
- **Routing-Komponente:**
  - Konnektivität auf Vermittlungsschicht
  - Sammlung von Topologiedaten
- **Dienstauffindungskomponente:**
  - Dienstauffindungsprotokoll
  - Auswahl geeigneter Dienstinstanz
- **Dienstplatzierungsmiddleware:**
  - Erstellung der Nutzungsstatistik
  - Implementierung von Basisfunktionalität, z.B. Migration und Replikation von Dienstinstanzen
  - Generisches Interface für Dienstplatzierungsalgorithmen

- 1) Aggregation von Topologie- und Nutzungsinformation auf dienstspezifischem Koordinatorknoten
- 2) Berechnung der optimalen Dienstkonfiguration
  - Minimum der Diensterbringungskosten  $p(s, H_s)$  für Dienst  $s$  bei Dienstkonfiguration  $H_s$
  - Diensterbringungskosten entsprechen der dienstbedingten Netzlast



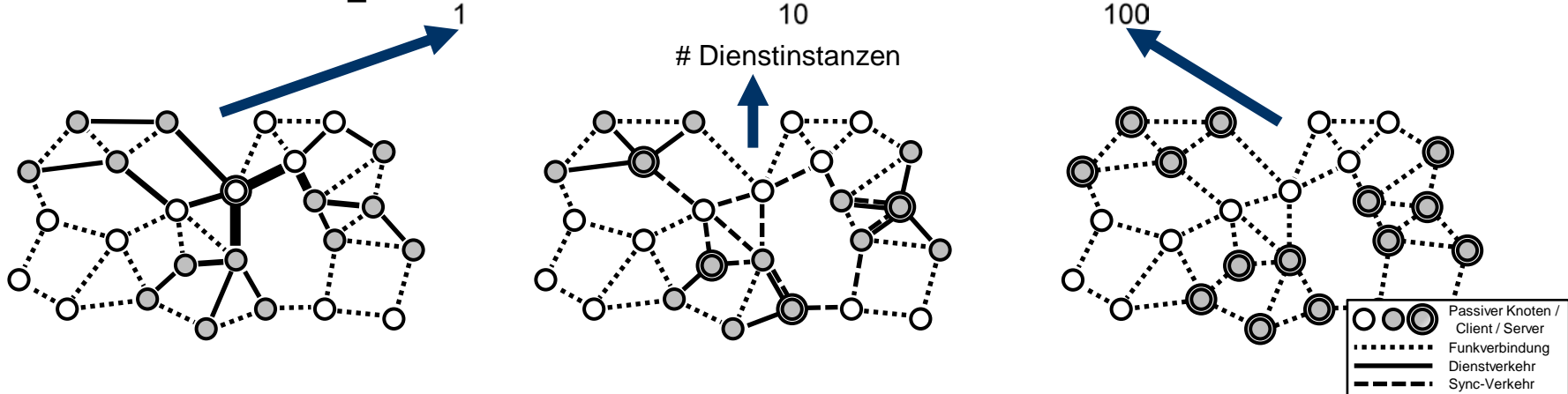
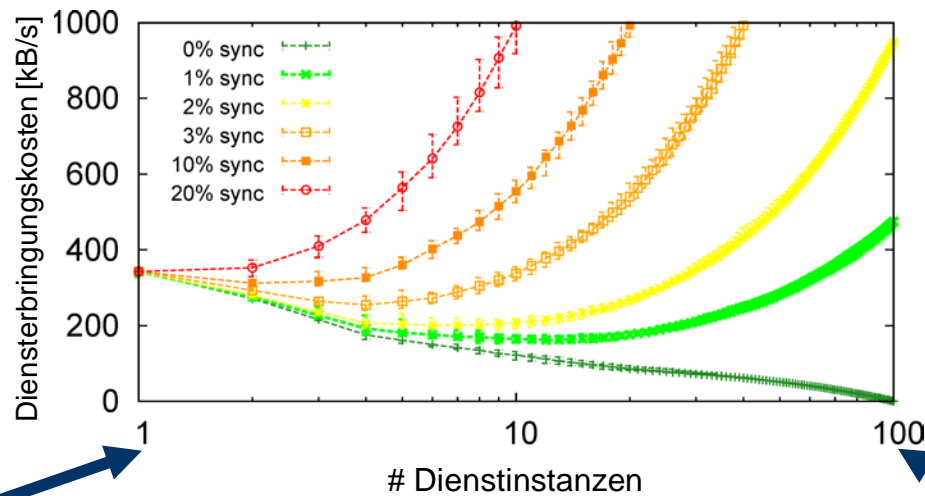
- 3) Entscheidung, die aktuelle Dienstkonfiguration anzupassen
- 4) Ggf. Erteilung von Instruktionen zur Adaptation der Dienstkonfiguration an aktuelle Dienstinstanzen

- Berechnung der Kostenfunktion  $p(s, H_s)$  für jeden Knoten als einziges Element der Dienstkonfiguration
- Knoten mit geringsten Diensterbringungskosten ist potentiell Migrationziel

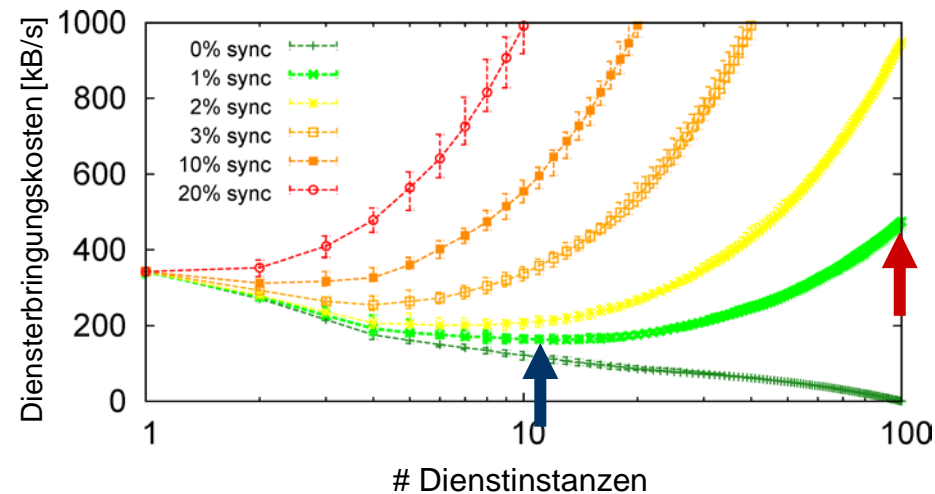
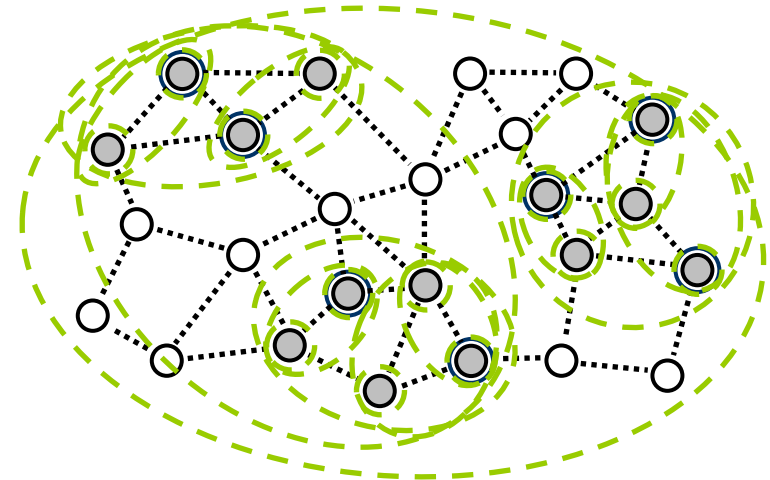


- Sehr einfach, da über *SPi*-Middleware auf Topologie- und Nutzungsinformation zurückgegriffen werden kann

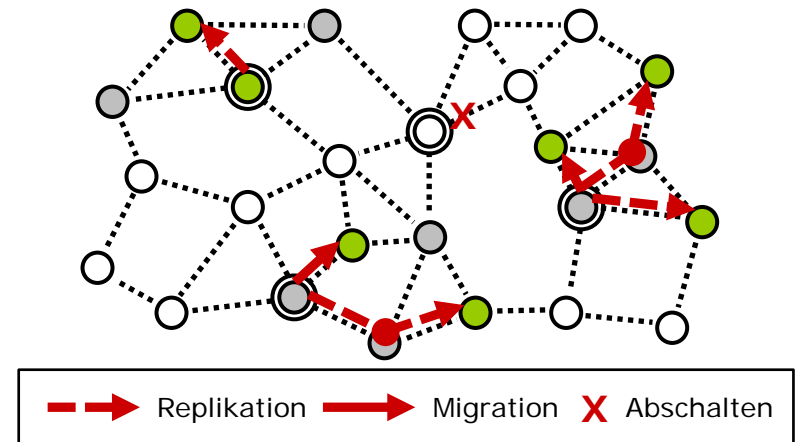
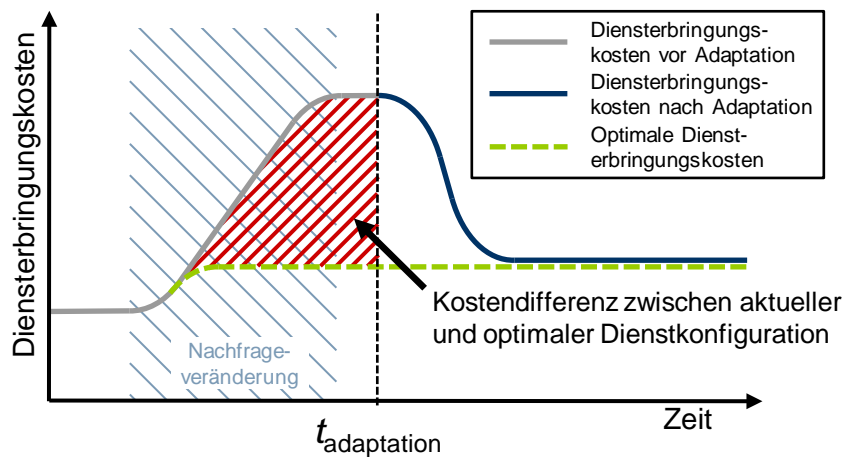
- Anzahl der Instanzen beeinflusst maßgeblich Dienstleistungskosten
- Optimale Anzahl ergibt sich aus Synchronisationsbedarf zwischen Dienstinstanzen



- Beobachtung:
  - Dienstnehmende Knoten verwenden nächstliegende Instanz
  - Clusterbildung um Dienstinstanzen
- Ansatz:
  - Initialisierung der Dienstkonfiguration  $H_s$  mit einer Instanz pro dienstnehmenden Knoten
  - Iteratives Verschmelzen des Clusterpaares mit geringster Dienstanfrage
  - Berechnung von  $p(s, H_s)$  für Dienstkonfiguration aus jeweiligen Clusterzentren



- Berechnung der potentiellen Kostenreduktion
  - Differenz zwischen Kosten der aktuellen und der optimalen Dienstkonfiguration



➤ Anpassung der Dienstkonfiguration, sobald Reduktion der Dienstbringungskosten die Adaptationskosten übertrifft



## Plattformen:

- Netzsimulator *ns-2*
  - Skalierbarkeit (Netzgröße und -last)
  - Anpassungsfähigkeit (zeitlich variable / regionale Nachfrage, Synchronisationsbedarf)
  - Vergleich mit anderen Ansätzen
- IEEE 802.3 LAN
  - Skalierbarkeit (Netzlast)
  - Einfluss von Paketverlusten
- IEEE 802.11 Testbed
  - Skalierbarkeit (Netzlast)
  - Überprüfung der Ergebnisse unter realen Bedingungen

## Metriken:

- Bzgl. der Platzierung:
  - Anzahl der Dienstinstanzen
  - Abstand zwischen Dienstinstanz und dienstnehmenden Knoten
  - Anzahl der Replikationen (und Migrationen)
- Bzgl. des Dienstes:
  - Rücklaufquote (engl. recall)
  - Datendurchsatz des Dienstes (engl. goodput)
  - Verzögerungszeit (engl. round-trip time)
  - Netzlast (engl. network load)

## Plattformen:

- Netzsimulator *ns-2*
  - Skalierbarkeit (Netzgröße und -last)
  - Anpassungsfähigkeit (zeitlich variable / regionale Nachfrage, Synchronisationsbedarf)
  - Vergleich mit anderen Ansätzen
- IEEE 802.3 LAN
  - Skalierbarkeit (Netzlast)
  - Einfluss von Paketverlusten
- IEEE 802.11 Testbed
  - Skalierbarkeit (Netzlast)
  - Überprüfung der Ergebnisse unter realen Bedingungen

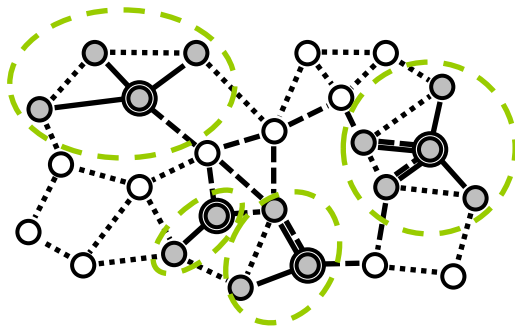
## Metriken:

- Bzgl. der Platzierung:
  - Anzahl der Dienstinstanzen
  - Abstand zwischen Dienstinstanz und dienstnehmenden Knoten
  - Anzahl der Replikationen (und Migrationen)
- Bzgl. des Dienstes:
  - Rücklaufquote (engl. recall)
  - Datendurchsatz des Dienstes (engl. goodput)
  - Verzögerungszeit (engl. round-trip time)
  - Netzlast (engl. network load)

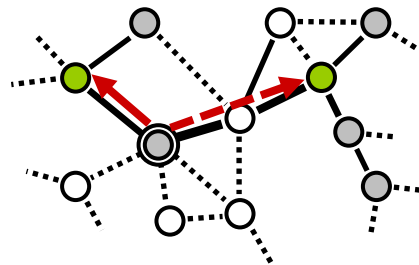
- Netzsimulator *ns-2* v2.33 mit 802.11b, DYMO Routing
- Variable Netzgröße: [16, 25, ... 100] Knoten
  - Zufällige Anordnung der Knoten
  - Annähernd konstanter Knotengrad durch Anpassung des Simulationsareals

- Platzierungsalgorithmen:

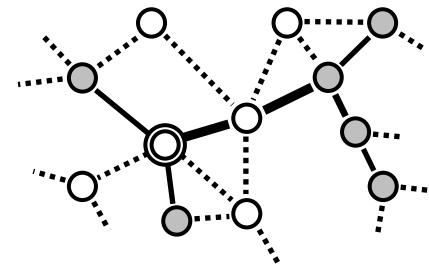
SPi/GCMI

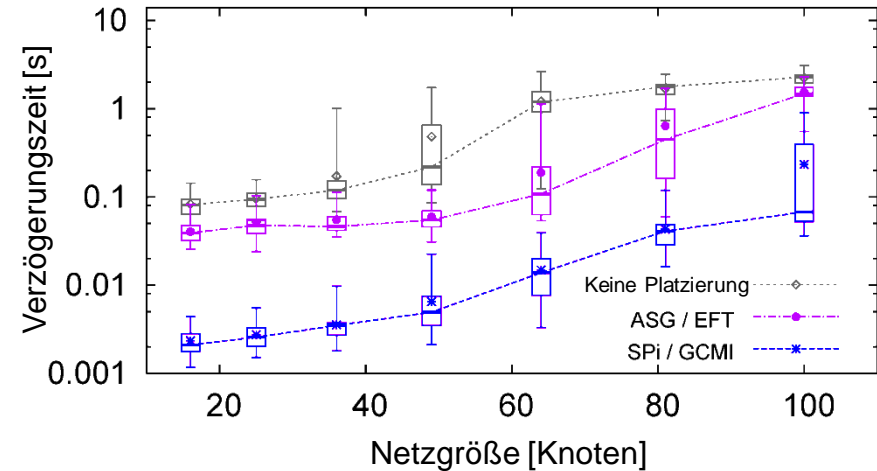
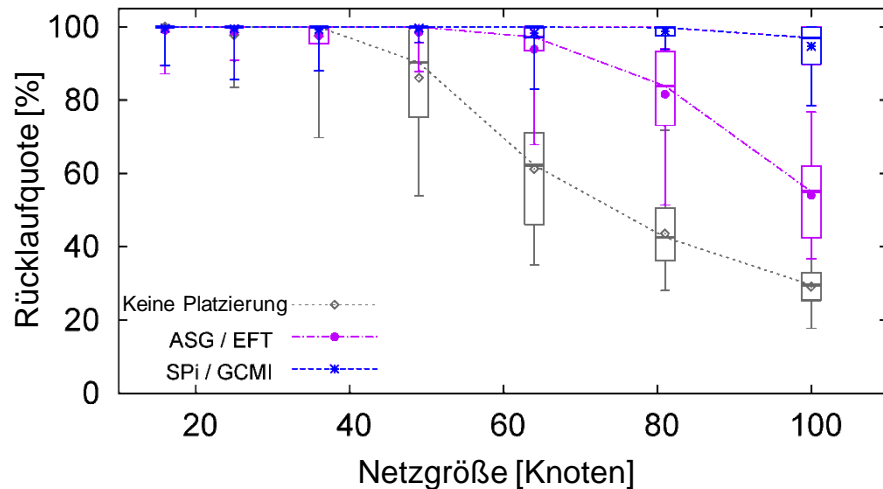


ASG/EFT

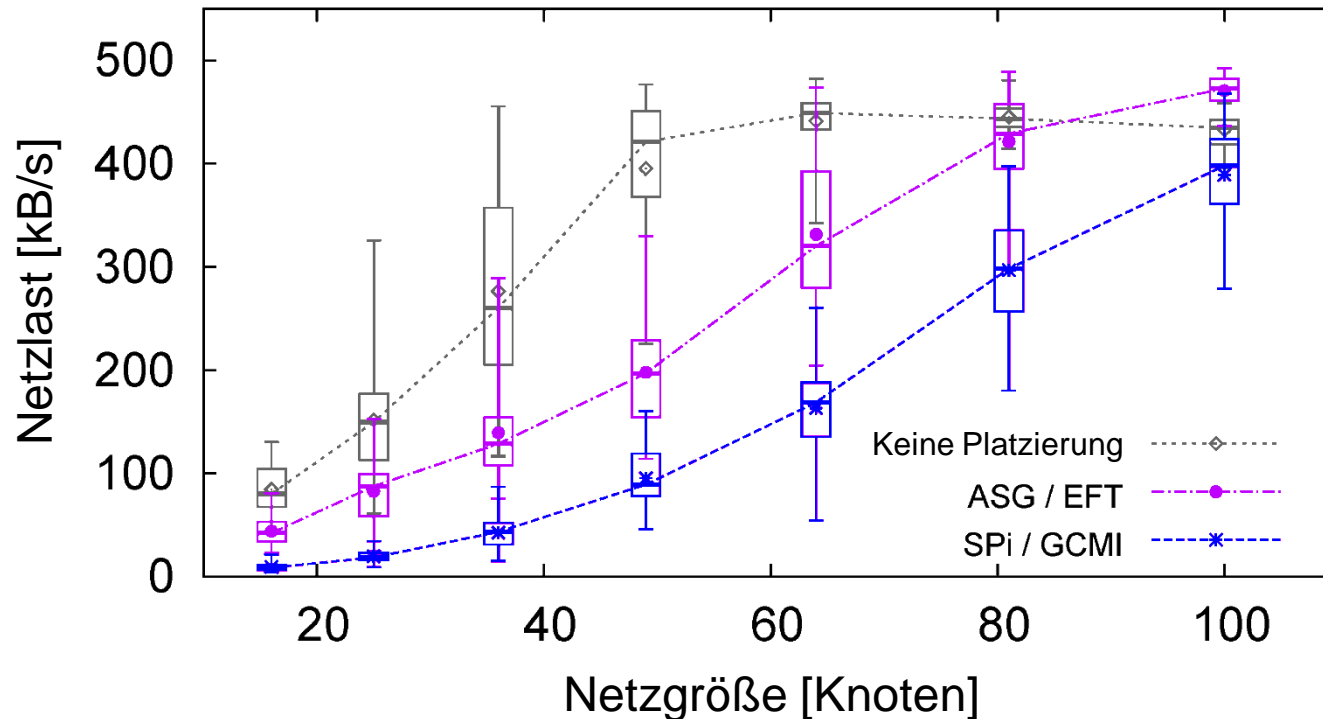


Keine Platzierung





- SPi/GCMI ermöglicht annähernd vollständige Bearbeitung von Dienstanfragen
- SPi/GCMI verringert Verzögerungszeit um mehr als eine Größenordnung



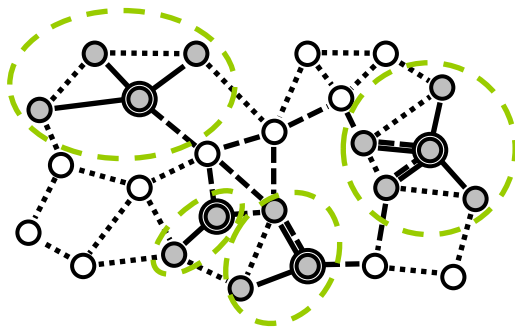
- *SPi/GCMI* vermindert die zur Dienstleistung erforderlichen Netzlast

# Versuchsaufbau – Testbed

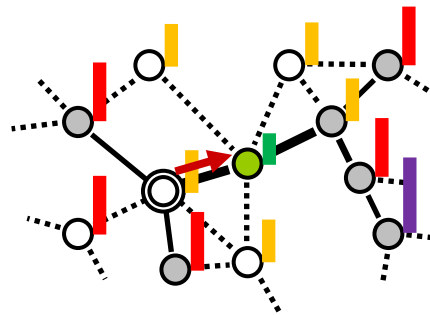
- DES-Testbed: Linux, IEEE 802.11b, DYMO Routing
- Variable Netzlast: [0,1; 0,2; 1,0; 2,0; 5,0] Anfragen/s
- Platzierungsalgorithmen:



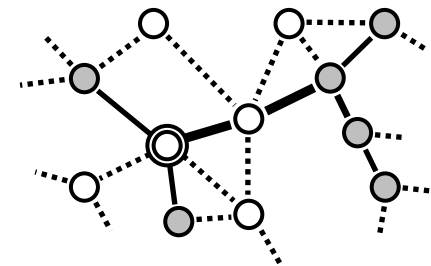
SPi/GCMI



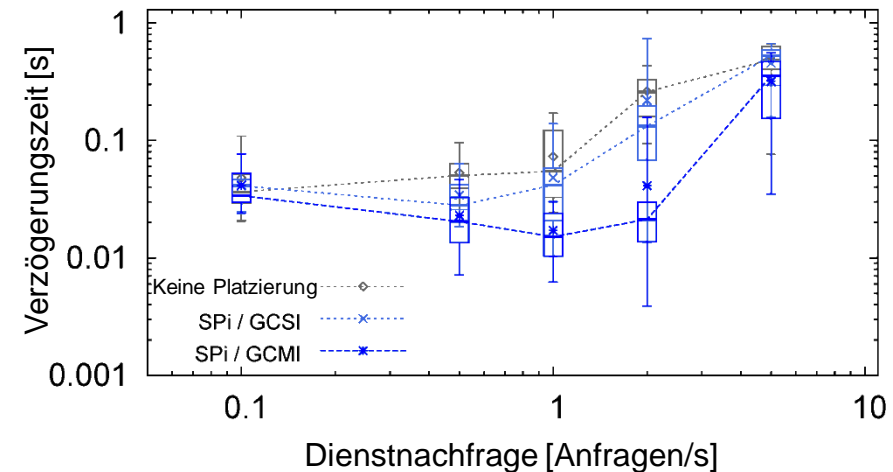
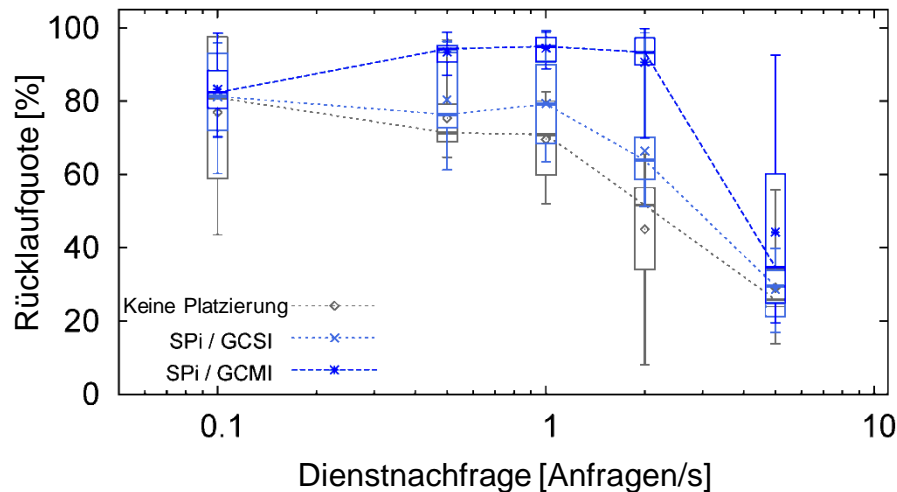
SPi/GCSI



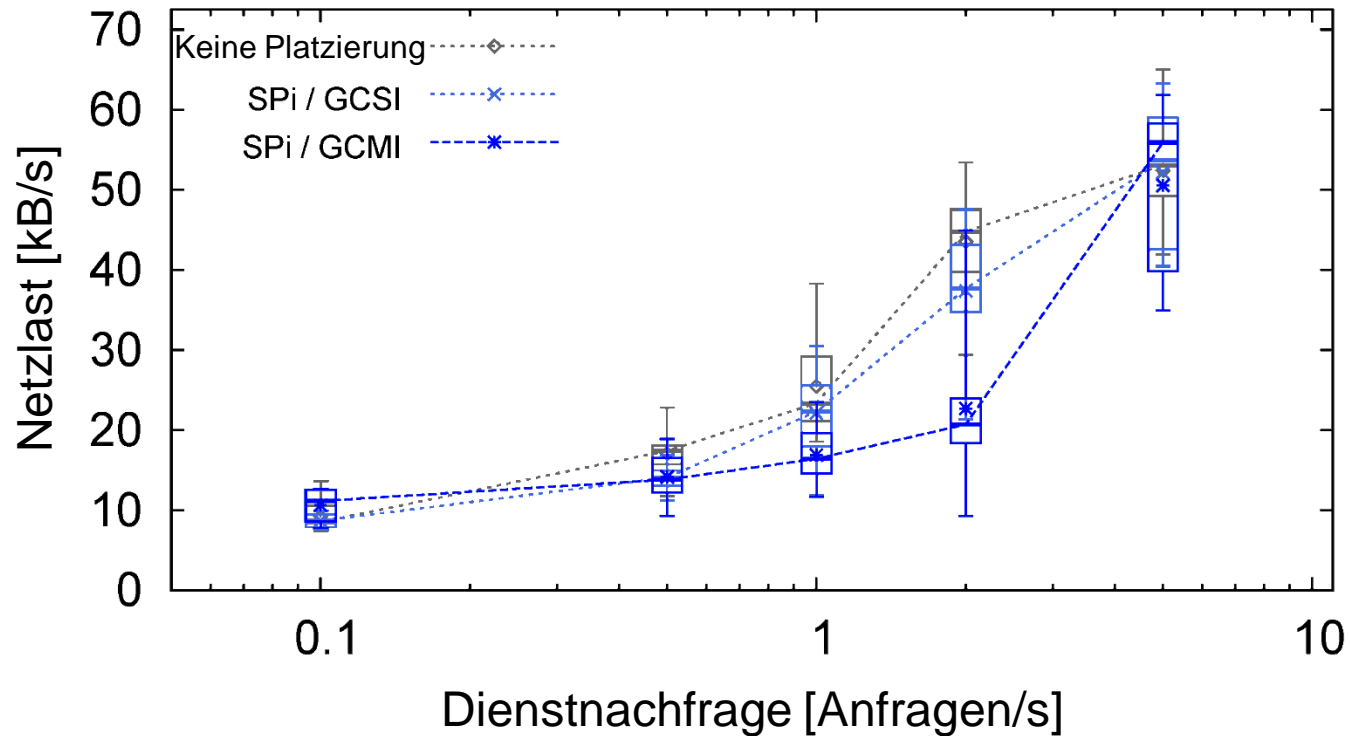
Keine Platzierung







- SPi/GCMI verbessert Rücklaufquote und Verzögerungszeit
- SPi/GCSI hat ähnlichen, aber weniger ausgeprägten Effekt



- *SPi/GCMI* vermindert erneut die zur Diensterbringung erforderlichen Netzlast

	Keine Platzierung	SPi/GCSI		SPi/GCMI	
		absolut	relativ	absolut	relativ
<b>Rücklaufquote [%]</b>	59,21	67,20	<b>13,49%</b>	81,23	<b>37,20%</b>
<b>Verzögerungszeit [s]</b>	0,1840	0,1596	<b>-13,29%</b>	0,0881	<b>-52,13%</b>
<b>Netzlast [kB/s]</b>	29,53	26,87	<b>-9,02%</b>	22,98	<b>-22,19%</b>

Mittelwerte von 50 Messungen für alle Lastszenarien

- Vorteile der Dienstplatzierung mit SPi:
    - Reduziert Abstand (in Hops) zwischen dienst anbietenden und -nehmenden Knoten
    - Verbessert Dienstqualität bei gleichzeitiger Verringerung der zur Dienstleistung erforderlichen Netzlast
  - Weitere Vorteile (nicht im Vortrag):
    - Anpassung der Dienstkonfiguration an veränderliche regionale Dienstmuster
    - Verminderung negativer Auswirkungen von unzuverlässigen Funkverbindungen
- Verteilte Dienstleistung mit aktiver Dienstplatzierung übertrifft die Leistungsfähigkeit einer Client/Server-Architektur

- *SPi*-Rahmenwerk zur Dienstplatzierung
  - Möglichkeit zum quantitativen Vergleich von Platzierungsalgorithmen durch Implementierung der Basisfunktionalität
- Graph Cost / Single Instance (GCSI) und Graph Cost / Multiple Instances (GCMI) Dienstplatzierungsalgorithmen
  - Dienstplatzierung unter Betrachtung von Synchronisationsanforderungen und Adaptationskosten
- Methode zur plattformübergreifenden Evaluation von Kommunikationsprotokollen
  - Leichtgewichtiger Wechsel zwischen Evaluationswerkzeugen (Simulationen, Emulationen und Experimenten auf einem Testbed)
- Quantitative Evaluation von Dienstplatzierungsalgorithmen
  - Direkter Vergleich von acht Dienstplatzierungsalgorithmen

- Georg Kunz, Olaf Landsiedel, and Georg Wittenburg. From Simulations to Deployments. In Klaus Wehrle, Mesut Günes, and James Gross, editors, *Modeling and Tools for Network Simulation*. Springer, September 2010.
- Georg Wittenburg and Jochen Schiller. Service Placement in Ad Hoc Networks. In *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 33(1):21-25, January 2010.
- Georg Wittenburg and Jochen Schiller. Towards Self-Organizing, Integrated Service Placement in Ad Hoc Networks. In *IFIP Fourth International Workshop on Self-Organizing Systems (Poster Session)*, Zurich, Switzerland, December 2009.
- Georg Wittenburg and Jochen Schiller. A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks. In *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08, Middleware Support for Pervasive Computing Workshop)*, Hong Kong, China, March 2008.
- Georg Wittenburg and Jochen Schiller. Running Real-World Software on Simulated Wireless Sensor Nodes. In *Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks (REALWSN '06)*, pages 7-11, Uppsala, Sweden, June 2006.