



Telecommunication
Networks Group

Praktikum Rechnernetze und Verteilte Systeme

Block 5

— Lösungsansatz für Aufgabe 5 —

Termin: 10.–14.12.2018 & 17.12.2018–21.12.2018

Die in der Vorlesung vorgestellte Variante von *Chord* geht davon aus dass IDs immer explizit durch Lookups erfragt werden können. Dadurch kann so auch die Adresse eines Eintrags in der Fingertable immer aktiv erfragt werden. Allerdings sieht das im Praktikum vorgegebene Nachrichtenformat keinen solchen Lookup einer ID vor.¹ Daher möchten wir Ihnen hier einen Ansatz geben wie ein Knoten seine Fingertable auch *ohne zusätzliche Nachrichten* kontinuierlich aufbauen kann.

¹wenn man davon absieht die Hash-Funktion des Keys umzukehren

1 Forwarding

Bevor Sie die Fingertable umsetzen sollten sie sich zunächst Gedanken machen wie ein Knoten Anfragen auf dem Ring weiterleiten kann. Ein Blick in die Vorlesungsfolien verrät Ihnen dabei, dass zum Weiterleiten stets der Finger ausgewählt wird, dessen Startpunkt der nächste Vorgänger (!) zur angefragten ID ist². Betrachten Sie dazu folgenden Ring mit 8-bit IDs in Abbildung 1. Dazu ist in Tabelle 1 die Fingertable von Knoten 0 gegeben. Wenn in diesem Beispiel

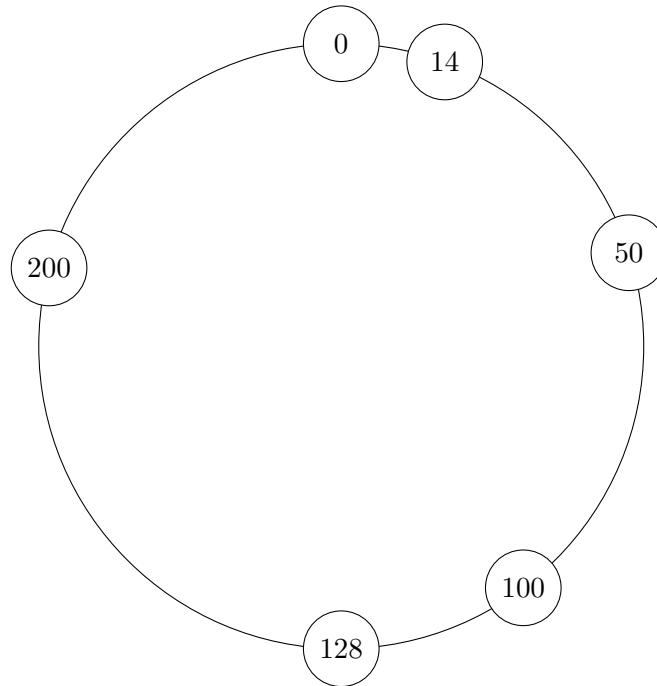


Abbildung 1: Beispierring mit 8-bit IDs.

Tabelle 1: Fingertable für Knoten 0

i	Start	Knoten
0	1	14
1	2	14
2	4	14
3	8	14
4	16	50
5	32	50
6	64	100
7	128	128

Knoten 0 eine Anfrage für die ID 58 bekommt, dann wird er diese an seinen Finger 5 weiterleiten, also an Knoten 50, obwohl Knoten 100 für diese ID zuständig ist. Da aber ein Knoten in einem verteilten System nie die vollständige Netzwerktopologie kennt, kann sich Knoten 0 nicht sicher sein, dass kein Knoten mit der ID 60 existiert. Wäre dass der Fall, würde das Paket einmal um den gesamten Ring wandern wenn Knoten 0 an Knoten 100 weiterleiten würde, da in Chord stets vorwärtsgerichtet weitergeleitet wird.

²Unit 05, S. 19

2 Lösungsansatz

Da die Regel zur Weiterleitung von Paketen unabhängig von der Aktualisierung ist, ist es sinnvoll diese beiden Funktionen zu trennen. Sie könnten also damit beginnen zunächst nur Forwarding von Paketen zu implementieren. Initialisieren Sie dafür alle Einträge in der Fingertable mit der ID des Nachfolgeknotens, denn im Zweifel kann an direkt Nachfolger immer weitergeleitet werden. Sobald Sie sicher sind, dass ihr Forwarding korrekt funktioniert, können Sie mit dem nächsten Schritt fortfahren.

Jetzt gilt es die Fingertable mit sinnvollen Einträgen zu aktualisieren. Eine Aktualisierung ist nur dann sinnvoll, wenn der Knoten sich sicher sein kann dass der Eintrag näher am Zielbereich ist, ohne Gefahr zu laufen den zuständigen Knoten zu überspringen. Der Schlüssel dies zu garantieren liegt in den ACKs. Antwortet ein Knoten N auf eine Anfrage für Key K , kann man sich sicher sein, dass er *mindestens* zuständig ist für das Intervall $[hash(K), N]$. Fällt der Startpunkt eines Fingers in diesen Bereich, kann man den Eintrag auf die ID des Knoten N setzen. Zusätzlich kann für alle nachfolgenden Finger überprüft werden ob deren eingetragene Knoten ID vor N liegt. Ist das der Fall können diese Finger ebenfalls aktualisiert werden, da N näher am Ziel liegt. Wenn ein Knoten genügend Anfragen für möglichst verteilte Keys bekommt, sodass er eine Antwort von allen Knoten aus dem Bereich seiner Finger erhält, kann eine vollständige Fingertable aufgebaut werden. Zusammengefasst können Sie also wie folgt vorgehen:

- Initialisieren Sie alle Finger der Fingertable mit der ID des Nachfolgers.
- Speichern Sie für jede Anfrage auf Key K , welche Sie an andere Knoten weiterleiten, den zugehörigen ID Wert ($hash(K)$) zur Transaction ID mit.
- Sobald Sie ein ACK für den Key K erhalten von Knoten N für eine Anfrage, überprüfen Sie für jeden Finger i folgende Bedingungen:
 - Liegt *start* von $FT[i]$ hinter oder genau auf $hash(K)$?
 - Liegt *start* von $FT[i]$ vor oder genau auf N ?
- Falls beide Bedingungen zutreffen: Setzen Sie $FT[i]$ auf Knoten N
- Falls nicht: Überprüfen Sie
 - ob *start* von $FT[i]$ hinter N liegt.
 - und ob der Eintrag in $FT[i]$ vor N liegt.
- Falls ja: Setzen Sie $FT[i]$ auf Knoten N

3 Beispiel

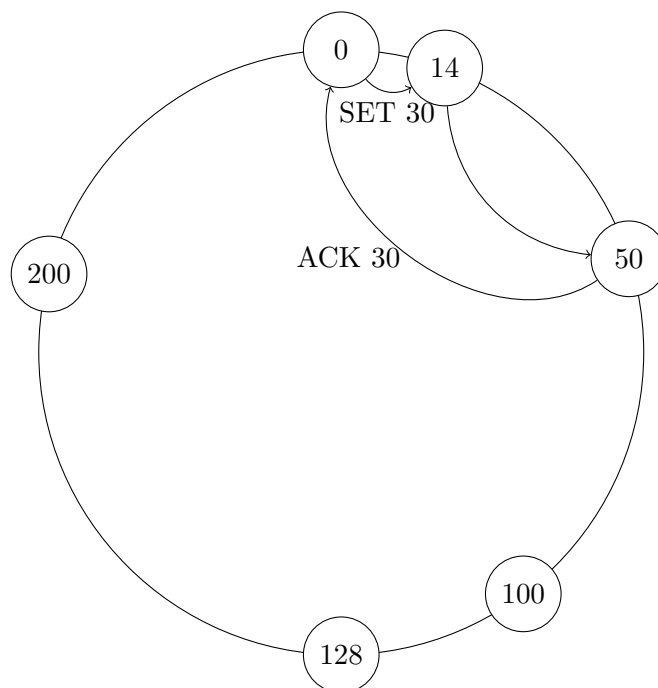


Abbildung 2: Beispiel für Anfrage auf ID 30 von Knoten 0

Tabelle 2: Fingertable für Knoten 0 - Vorher und Nachher

i	Start	Knoten		i	Start	Knoten
0	1	14	→	0	1	14
1	2	14		1	2	14
2	4	14		2	4	14
3	8	14		3	8	14
4	16	14		4	16	14
5	32	14		5	32	14 50
6	64	14		6	64	14 50
7	128	128		7	128	128