

Wintersemester 2018/2019

## Peer-to-Peer und Cloud Computing

### Lösungsvorschläge zu Aufgabenblatt 5

#### Strukturierte P2P-Systeme: CAN (22 Punkte)

Gegeben sei ein leeres CAN. Eine Zone teilt sich immer in der Mitte ihrer längeren Seite; sind die Seiten gleich lang, teilt sie sich in der ersten Dimension (also *vertikal*). Der neue Knoten bekommt *immer* die rechte (bei vertikaler Teilung) beziehungsweise die obere Zone (bei horizontaler Teilung) zugewiesen – unabhängig von der von ihm gewählten Koordinate zur Identifizierung seines Einstiegsknotens.

1. In dieses CAN ordnen sich nun nacheinander die folgenden neun Knoten ein:

a)  $v_1 = (0,70; 0,60)$

d)  $v_4 = (0,40; 0,90)$

g)  $v_7 = (0,40; 0,40)$

b)  $v_2 = (0,20; 0,20)$

e)  $v_5 = (0,90; 0,90)$

h)  $v_8 = (0,40; 0,60)$

c)  $v_3 = (0,70; 0,20)$

f)  $v_6 = (0,90; 0,40)$

i)  $v_9 = (0,20; 0,60)$

Zeichnen Sie die zweidimensionale Struktur des Netzwerkes nach jedem neu hinzugefügten Knoten (inklusive Knotenbeschriftungen, siehe Abbildung 1). Am Ende sollten Sie *neun* Netzwerke gezeichnet haben! (4 Punkte)

#### Lösung

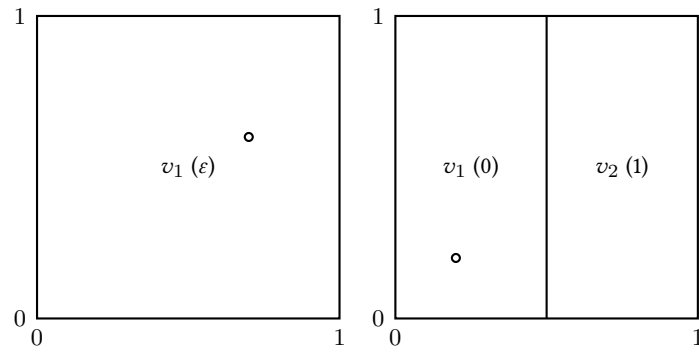
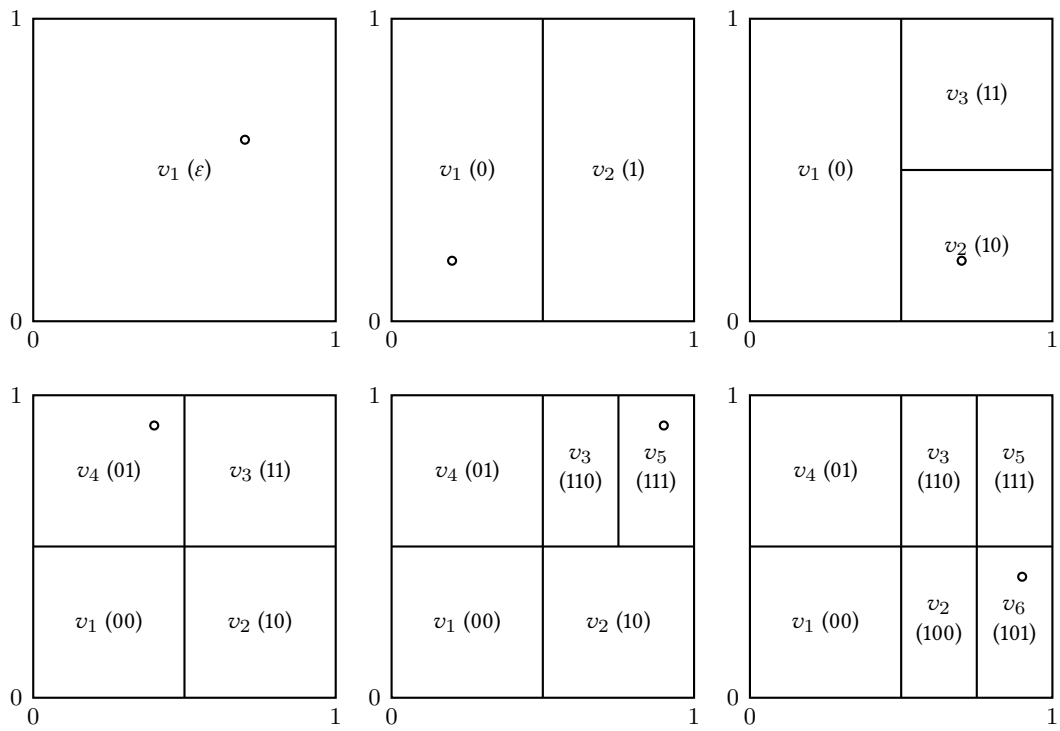
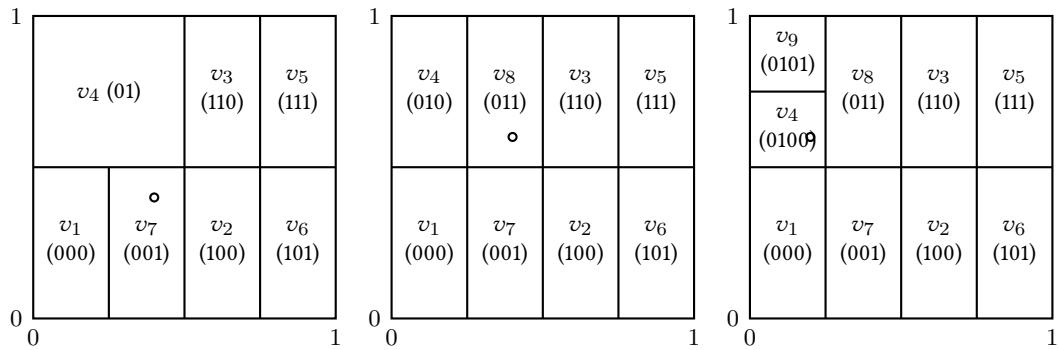


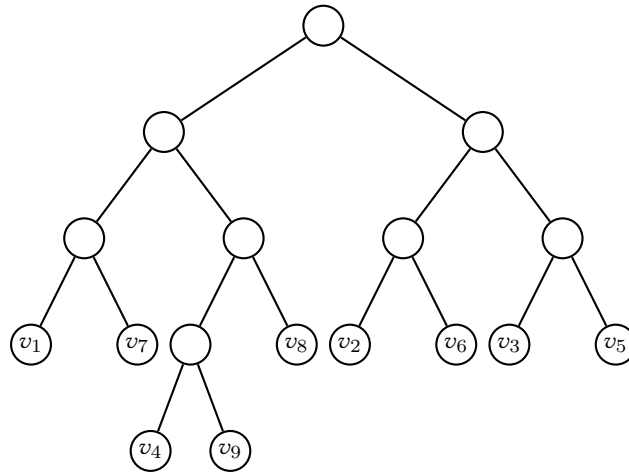
Abbildung 1: Die Struktur des Netzwerks in den ersten zwei Schritten von Teilaufgabe 1 (bereits mit den VIDs aus Teilaufgabe 3,  $\varepsilon$  steht hierbei für das leere Wort).





2. Zeichnen Sie den Partitionsbaum des endgültigen Netzwerks (linke Teilbäume stehen dabei für die linke bzw. untere Partition – also diejenige, die die kleineren Paare aus  $[0; 1] \times [0; 1]$  enthält –, rechte Teilbäume für die rechte bzw. obere). (4 Punkte)

### Lösung



3. Zeichnen Sie die *virtual IDs* (VIDs) in alle Zwischenschritte ein. (3 Punkte)

### Lösung

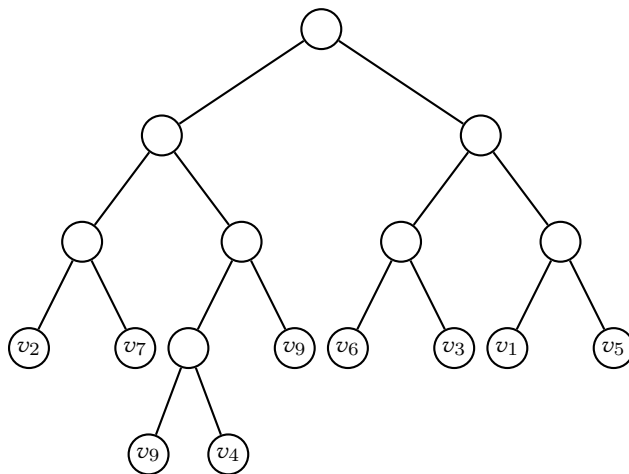
Siehe Lösung zu Teilaufgabe 1.

4. Schließlich meldet Knoten  $v_8$  seinen Austritt aus dem Netzwerk.
- a) Beschreiben Sie, auf welche Weise sich die Struktur des Netzwerks verändern *sollte*! Argumentieren Sie dabei mithilfe des Partitionsbaums und zeichnen Sie beide Alternativen für die neue Netzwerkstruktur sowie die zugehörigen neuen Partitionsbäume! (5 Punkte)

### Lösung

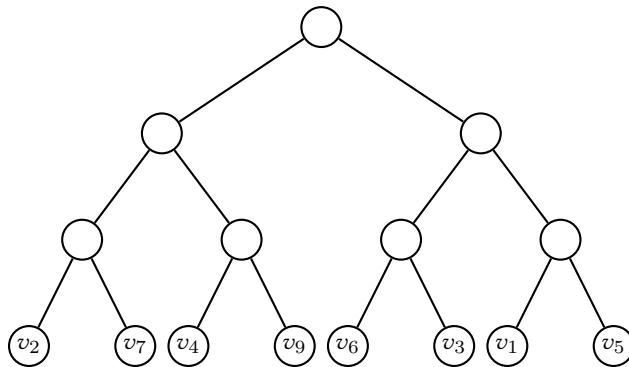
- Geschwisterteil  $v$  von  $v_8$  ist kein Blatt
- somit Tiefensuche im Teilbaum  $t_v \setminus \{v_8\}$  (bis ein Blatt gefunden wird)
- $v_9$  wird als Blatt gefunden (oder  $v_4$ )  
 $\Rightarrow v_9$  wird Takeover-Knoten
- Zusammenführen der Zonen nicht möglich  
 $\Rightarrow$  zwei neue Netzwerkstrukturen möglich
  - $v_9$  managt Zone von  $v_8$  bis neuer Knoten in  $v_9$  beitrifft und übergibt diese an ihn

1	$v_9$ (0101)	$v_9$ (011)	$v_3$ (110)	$v_5$ (111)
	$v_4$ (0100)			
0	$v_1$ (000)	$v_7$ (001)	$v_2$ (100)	$v_6$ (101)
0				1



- die Zone von  $v_9$ 's Geschwisterteil  $v_4$  wird mit der von  $v_9$  zusammengelegt und nun von  $v_4$  verwaltet;  $v_9$  übernimmt Zone von  $v_8$

1	$v_4$ (010)	$v_9$ (011)	$v_3$ (110)	$v_5$ (111)
0	$v_1$ (000)	$v_7$ (001)	$v_2$ (100)	$v_6$ (101)
	0			1

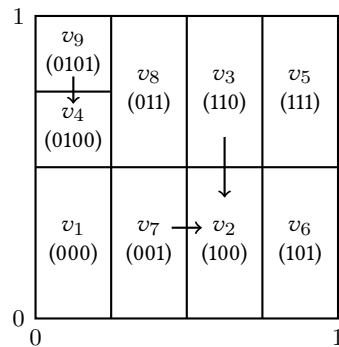


- b) In Echtwelt-Netzwerken kennen die Knoten den Partitionsbaum nicht, sondern nur die VIDs ihrer Nachbarn. Beschreiben Sie den Ablauf des Austritts von  $v_8$ , wenn für die Recovery-Nachrichten *Greedy Forwarding* benutzt wird! (2 Punkte)

### Lösung

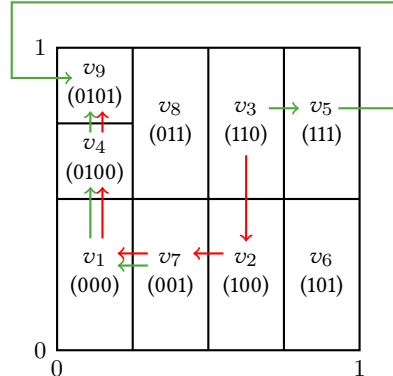
Zwei mögliche Antworten.

- i. Fall:  $v_8$  verlässt das Netzwerk *gracefully*.
  - kann  $v_9$  (oder  $v_4$ ) vorher noch direkt ansprechen
  - $v_9$  (oder  $v_4$ ) übernimmt wie in a) beschrieben die Zone von  $v_8$
  - $v_8$  kann auch noch seine anderen Nachbarn über seine Wahl des Takeover-Knoten benachrichtigen
- ii. Fall:  $v_8$  versagt.
  - Nachbarn ( $v_3, v_4, v_7, v_9$ ) erkennen Versagen  
 $\Rightarrow$  senden Recovery-Nachrichten an ihren Nachbarn, dessen VID der von  $v_8$  am nächsten ist
  - numerische Nachbarauswahl (Interpretation des VID-Bitstring als natürliche Zahl)



⇒ als Takeover-Knoten werden identifiziert:  $v_2$  und  $v_4$

- präfixbasierte Nachbarsauswahl
  - längste Präfixübereinstimmung
  - Mehrere gleichgeeignete? Zufällige Auswahl!
  - Keiner geeigneter aber jemand genauso geeignet wie aktueller Knoten? Zufällige Auswahl eines Nachbarn!
  - Keiner geeigneter oder genauso geeignet wie aktueller Knoten? Aktueller Knoten ist Takeover-Knoten!
  - Recovery-Nachricht nie zurückschicken



⇒ als Takeover-Knoten wird identifiziert (unabhängig von Zufall):  $v_9$

- c) Welches Problem tritt dabei auf? (Wenn in Ihrer Lösung kein Problem auftritt: Welches Problem könnte im Allgemeinen beim Austritt eines Knoten unter Benutzung von *Greedy Forwarding* auftreten?) (1 Punkt)

**Lösung**

Sowohl  $v_2$  und  $v_4$  werden als Takeover-Knoten ausgewählt.

Außerdem: Wenn mehrere Knoten gleichzeitig ausfallen, kann es auch dazu kommen, dass einzelne Recovery-Nachrichten nicht den eigentlichen Takeover-Knoten erreichen. Dies könnte zu Inkonsistenzen durch Fehlentscheidungen aufgrund von unvollständigen Informationen führen (Netzwerkstruktur wird nicht mehr aufrechterhalten).

- d) Welche Lösung gibt es für das Problem aus Teilaufgabe c)? Wie funktioniert sie? Zeichnen Sie sie auch in das letzte CAN-Netzwerk aus Teilaufgabe 1 (also noch mit Knoten  $v_8$ ) ein! (3 Punkte)

### Lösung

Aufbauen eines Chord-Rings (aka Linked-List) durch alle Peers, Benutzen von Chord-Routing statt *Greedy Forwarding*: Hierzu wird eine Ordnung auf den VIDs der Peers definiert. Sie „interpretiert“ die VID-Bitstrings der Peers als natürliche Zahl und erlaubt Vergleiche, obwohl die Bitstrings mitunter von verschiedener Länge sind. Gibt es im Netzwerk VID-Bitstrings unterschiedlicher Länge, so füllt die Ordnung quasi die VIDs, die weniger Zeichen, als die Peers mit dem längsten Bitstring haben, auf deren Länge mit nachfolgenden Nullen auf. Danach können die Peers sortiert, und gemäß der Ordnung, in den Chord-Ring eingefügt werden. Da es sich um einen Ring handelt, muss natürlich am Ende der Peer, mit der numerisch höchsten VID, mit dem Peer, der die niedrigste VID hat, verbunden werden.

Somit ergibt sich folgender Chord-Ring:

