

ration unberührt. Diese Eigenschaft wird *Konsistenz* genannt. Daher verwendet man für verteilte Hash-Tabellen mitunter auch den Begriff *Consistent Hashing*.

Wenn die Hash-Funktion die Daten gleichmäßig auf das Quadrat verteilt, muss nur noch gewährleistet werden, dass den Peers gleich große Bereiche zugewiesen werden, um die Daten gleichmäßig auf die Peers zu verteilen.

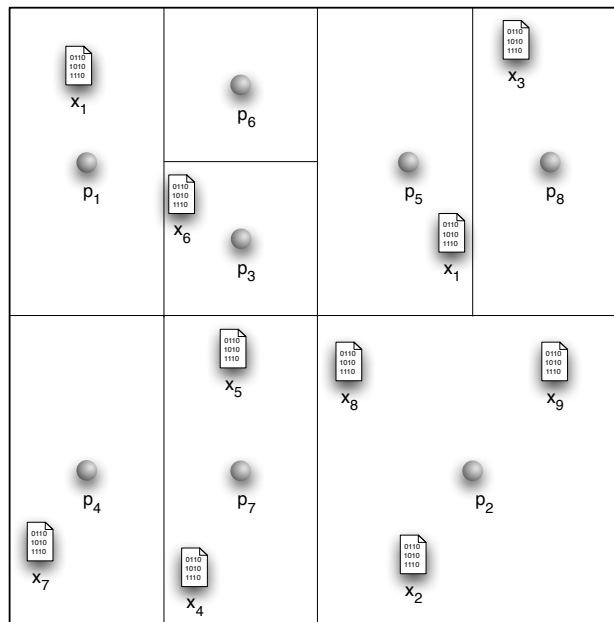


Abb. 4.2. Zuordnung von Daten zu Peers im CAN.

## 4.2 Einfügen von Peers

Zu Beginn besteht ein CAN aus nur einem Peer, der das gesamte Quadrat  $Q$  verwaltet. Wir beschränken uns zunächst auf den Fall, dass nur neue Peers hinzukommen und keine Peers das Netzwerk verlassen.

Jeder neue Peer wählt zuerst einen zufälligen Punkt im Quadrat  $Q$ . Dann kontaktiert der neue Peer den für diesen Punkt  $z$  zuständigen Peer. Das ist der Besitzer des  $z$  umgebenden Rechtecks. Diese Kontaktaufnahme geschieht durch die Suchfunktion, die wir später erläutern. Nun wird dieses Rechteck halbiert. Wenn es sich um ein Quadrat handelt, dann zuerst vertikal entlang der  $y$ -Achse und bei einem Rechteck entlang der  $x$ -Achse. In Abbildung 4.3 wird dargestellt, wie dieser Prozess das Quadrat immer weiter in kleinere Rechtecke unterteilt.

Nach der Unterteilung wird auch die Netzwerkstruktur entsprechend der neuen Nachbarschaft angepasst. Wir wenden uns zunächst der Frage zu, wie gleichmäßig die Daten durch diese Einfügeoperation auf die Peers verteilt werden.

Der Anteil der Daten eines Peers ist erwartungsgemäß proportional zur Fläche des von ihm verwalteten Rechtecks. Zwar ist jeder Eintrittspunkt gleichwahrscheinlich, so dass große Rechtecke beim Einfügen weiterer Peers mit höherer Wahrscheinlichkeit geteilt werden als kleine, eine vollkommen gleichmäßige Aufteilung ist jedoch sehr unwahrscheinlich.

Wir betrachten einen Peer  $p$  mit Rechteck  $R(p)$  in CAN. Sei  $A(p)$  die Fläche dieses Rechtecks. Das folgende Lemma liefert eine obere Schranke für die Wahrscheinlichkeit, dass das Rechteck  $R(p)$  nicht weiter unterteilt wird, wenn  $n$  weitere Peers eingefügt werden.

**Lemma 4.1.** *Sei  $P_{R,n}$  die Wahrscheinlichkeit, dass ein Rechteck  $R$  mit Fläche  $A(R)$  nach dem Einfügen von  $n$  Peers ungeteilt bleibt. Dann gilt*

$$P_{R,n} \leq e^{-nA(R)}.$$

*Beweis.* Wir betrachten ein Rechteck  $R$  mit der Fläche  $q = A(R)$ . Die Wahrscheinlichkeit, dass ein bestimmter Peer diese Fläche im Zuge des Einfügens nicht weiter unterteilt, also zufällig keinen Punkt innerhalb von  $R$  wählt, ist  $1 - q$ . Das geschieht unabhängig vom Verhalten anderer Peers. Somit ist die Wahrscheinlichkeit, dass  $n$  Peers keinen Punkt aus  $R$  wählen, das Produkt aller  $n$  Einzelwahrscheinlichkeiten  $1 - q$  und damit  $(1 - q)^n$ . Nun gilt für alle  $m > 0$ :

$$\left(1 - \frac{1}{m}\right)^m \leq \frac{1}{e}.$$

Daraus folgt

$$P_{R,n} = (1 - q)^n = \left((1 - q)^{\frac{1}{q}}\right)^{nq} \leq e^{-nq} = e^{-nA(R)}$$

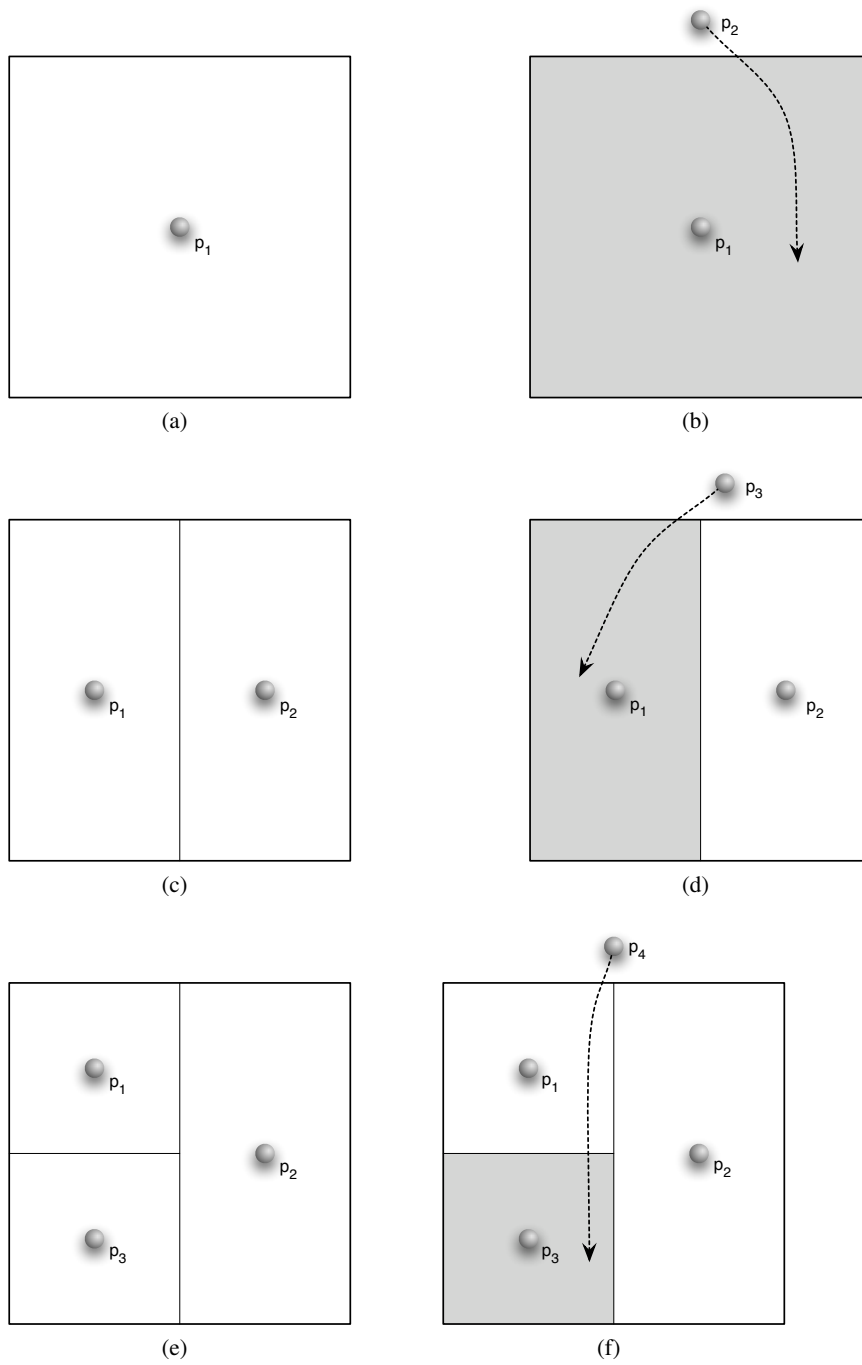
und somit das Lemma.  $\square$

So ist es zum Beispiel extrem unwahrscheinlich, dass eine Fläche der Größe  $\frac{1}{2}$  nach dem Eintreffen von  $n$  weiteren Peers ungeteilt bleibt. Die Wahrscheinlichkeit ist nach Lemma 4.1 höchstens  $e^{-n/2}$  (tatsächlich sogar höchstens  $2^{-n}$ ).

Mit Hilfe von Lemma 4.1 können wir nun eine Aussage darüber treffen, wie groß die Rechtecke in einem CAN höchstens sein werden.

**Theorem 4.2.** *In einem CAN wird es nach dem Einfügen von  $n$  Peers mit hoher Wahrscheinlichkeit, d.h. mit Wahrscheinlichkeit  $\geq 1 - n^{-c}$  für eine Konstante  $c > 0$ , kein Rechteck  $R$  mit Fläche  $A(R) \geq 2c \frac{\ln n}{n}$  geben.*

*Beweis.* Bezeichne  $R_i$  ein Rechteck mit Fläche  $A(R_i) = 2^{-i}$ . Mit Hilfe von Lemma 4.1 können wir die Wahrscheinlichkeit  $P_{R_i, c2^i \ln n}$ , dass ein Rechteck der Größe  $R_i$  von  $c2^i \ln n$  Peers nicht geteilt wird, abschätzen:



**Abb. 4.3.** Einfügen von Peers in das CAN.

$$\begin{aligned}
P_{R_i, c2^i \ln n} &\leq e^{-A(R_i)c2^i \ln n} \\
&= e^{-c \ln n} \\
&= n^{-c}.
\end{aligned}$$

Es genügen also  $c 2^i \ln n$  Peers, um  $R_i$  mit Wahrscheinlichkeit  $1 - n^{-c}$  zu teilen. Damit ein Rechteck  $R_{i+1}$  überhaupt entsteht, muss das entsprechende Rechteck  $R_i$  geteilt werden. Wir betrachten das Einfügen weiterer Peers nun staffelweise für  $i = 1, 2, \dots, \log \frac{n}{2c \ln n}$ . In Staffel  $i$  werden  $c2^i \ln n$  neue Peers eingefügt. Diese teilen das umschließende Rechteck  $R_i$  eines Peers mit hoher Wahrscheinlichkeit.

So wird nach  $\log \frac{n}{2c \ln n}$  solcher Staffeln mit hoher Wahrscheinlichkeit auch das Rechteck der Größe  $2c \frac{\ln n}{n}$  geteilt. Die Summe der Peers in den Staffeln ist nach folgender Rechnung höchstens  $n$ :

$$\begin{aligned}
\sum_{i=1}^{\log \frac{n}{2c \ln n}} c2^i \ln n &= c(\ln n) \sum_{i=1}^{\log \frac{n}{2c \ln n}} 2^i \\
&\leq c(\ln n) 2 \frac{n}{2c \ln n} \\
&= n.
\end{aligned}$$

Damit wird ein bestimmtes Rechteck der Größe  $2c \frac{\ln n}{n}$  mit Wahrscheinlichkeit  $n^{-c} \log n$  von  $n$  Peers nicht geteilt. Von diesen Rechtecken gibt es höchstens  $n$  Stück. Damit ist die Wahrscheinlichkeit, dass eines dieser Rechtecke ungeteilt bleibt, höchstens  $n \cdot n^{-c} \log n \leq n^{-c+2}$ . Wählt man also  $c$  groß genug, so kann jede *polynomiell kleine Fehlerwahrscheinlichkeit* erreicht werden.  $\square$

Wie schon erwähnt, ist die erwartete Anzahl von Daten, die ein Peer verwalten muss, proportional zur Fläche seines Rechtecks. Im Durchschnitt hat diese Fläche die Größe  $1/n$ . Theorem 4.2 zeigt, dass die Fläche eines Peers mit hoher Wahrscheinlichkeit kleiner als  $(2c \ln n)/n$  ist. Anders formuliert bedeutet dies, dass ein Peer mit hoher Wahrscheinlichkeit höchstens  $2c(\ln n)$  mal mehr Daten als der Durchschnitt verwalten muss.

### 4.3 Netzwerkstruktur und Routing

Die Netzwerkstruktur von CAN ergibt sich aus der Lage der Rechtecke der Peers. Ein Peer unterhält Verbindungen zu allen Peers, die für ein orthogonal benachbartes Rechteck verantwortlich sind. Beim Einfügen kann jeder Peer diese Information von dem Peer erfragen, dessen Rechteck hierbei geteilt wird. Natürlich müssen die Nachbar-Peers diese Information dann auch anpassen. Sind alle Rechtecke gleich groß und jeder Peer nur für ein Rechteck verantwortlich, dann ist der Grad des Netzwerk-Graphen konstant groß (siehe Abbildung 4.1). Tatsächlich ist es jedoch unwahrscheinlich, dass alle Rechtecke gleich groß sind. Ein Grund hierfür ist der