Fehlertolerantes Routing auf dem n-dim Würfel

Elmar Schömer

Diplomarbeit nach einem Thema von Professor Dr. Günter Hotz im Fachbereich 10, Angewandte Mathematik und Informatik, der Universität des Saarlandes.

Hiermit versichere ich an Eides Statt, daß ich diese Arbeit nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Saarbrücken, im Januar 1988

Inhaltsverzeichnis

Einleitung

Ein weltweites Telefonnetz ermöglicht vielen Menschen, auch über große Entfernungen hinweg miteinander zu kommunizieren. Auch die Kommunikation zwischen Rechnern gewinnt zunehmend an Bedeutung. Die Installation von Rechnernetzen dient der schnellen Übermittlung digitalisierter Daten jeglichen Ursprungs, aber auch der gemeinsamen Nutzung von Rechnerkapazität. Während das Telefonnetz nach dem Prinzip der Leitungsvermittlung arbeitet, benutzt man bei Rechnernetzen in der Praxis oft die sogenannte Paketvermittlung. Nachrichtenpakete sind wie bei der Post mit Adressen versehen und werden im Netz von Punkt zu Punkt weitergereicht. Diese Arbeitsweise erspart die Reservierung von Leitungen für die gesamte Dauer eines Gespräches, was zu einer besseren Auslastung der Leitungen führt.
Eine Vernetzung von vielen gleichartigen Rechnern betreibt man auch zur Entwicklung von leistungsstarken und universellen Parallelrechnern. Die eher abstrakten parallelen Rechnermodelle wie PRAMs simuliert man auf Rechnernetzen mit geeigneter Netzwerktopologie. Die Simulation eines read-write-Zyklus einer PRAM erfordert dabei eine beträchtliche Kommunikation zwischen den einzelnen Netzwerkrechnern. Im Modell der Paketvermittlung führt dies zu dem Problem des simultanen Routings von Paketen im gegebenen Netz. Mit Hilfe probabilistischer Routingverfahren läßt sich der Verzögerungsfaktor der Simulation jedoch klein halten.
Neben dem Aspekt der Geschwindigkeit sollte man bei immer größer werdenden Kommunikationsnetzen dem Faktor Zuverlässigkeit eine ebenso große Bedeutung beimessen. Der Ausfall einiger weniger Netzwerkkomponenten darf nicht gleich das gesamte System lahmlegen. Deshalb sollte man über Routingstrategien verfügen, die so viele Defekte "wie möglich" tolerieren, ohne große Geschwindigkeitsverluste hinnehmen zu müssen.
Das Ziel dieser Arbeit ist es, einen ersten Schritt in diese Richtung zu tun. Sie gliedert sich wie folgt:

Die **Einführung** gibt einen kurzen Einblick in die untersuchte Thematik und erläutert die grundlegenden Begriffe und Notationen. Sie enthält außerdem die Definition der betrachteten Kommunikationsnetzwerke.

Kapitel I beschäftigt sich mit der Problematik des Routings einer *Permutationsanforderung* auf einem intakten n-dim Würfel. Neben einer deterministischen Routingstrategie (BITONIC-SORT) wird das Laufzeitverhalten einer probabilistischen Routingstrategie untersucht, die von L.G. Valiant vorgeschlagen wurde (siehe **[VB]**,**[Va]**). Die Laufzeitanalyse verwendet eine verallgemeinerte Version des *Prinzips der kritischen Verzögerungsfolge*. Unter Verwendung dieser Beweistechnik, die auf E. Upfal zurückgeht (siehe

[**Upf**]), kann man zeigen, daß sich jede beliebige Permutationsanforderung mit hoher Wahrscheinlichkeit in Zeit $c \cdot n$ routen läßt. Dabei ist zur internen Verwaltung der Pakete *nur eine* Warteschlange notwendig, die als *priority queue* organisiert ist.

In den beiden folgenden Kapiteln werden zwei Techniken diskutiert, die es erlauben, Permutationsanforderungen auch auf einem n-dim Würfel mit nnicht allzu vielen" *Leitungsdefekten* schnell zu routen.

In **Kapitel II** wird folgender Weg eingeschlagen: Will man die Sicherheit der Nachrichtenübertragung zwischen zwei beliebigen Prozessoren steigern, so kann man im n-dim Würfel n Pakete mit identischer Nachricht auf disjunkten Wegen vom Sender zum Empfänger transportieren. Obwohl diese Methode das Kommunikationsaufkommen um das n-fache erhöht, kann man einen Geschwindigkeitsverlust beim Routen einer Permutationsanforderung vermeiden, wenn man die Prozessoren so ausrüstet, daß sie über alle Leitungen gleichzeitig senden und empfangen können, wie dies bei Valiant schon vorgesehen war.

In **Kapitel III** wird eine weitere Methode zum fehlertoleranten Routing auf dem n-dim Würfel vorgestellt: Die Route eines Paketes, das über eine defekte Leitung transportiert werden soll, wird so abgewandelt, daß die schadhafte Stelle *lokal* umgangen wird. Die Existenz defekter Leitungen verursacht eine zusätzliche Belastung auf den Leitungen der entsprechenden Umwege. Um diese Belastung möglichst gleichmäßig zu verteilen, muß eine Koordination der Umwege stattfinden. Dies kann mit Hilfe eines *heuristischen parallelen Matching*-Algorithmus auf dem defekten n-dim Würfel selbst geschehen. Die Erfolgsquote dieses Algorithmus ist auch bei einer *konstanten* Leitungsausfallwahrscheinlichkeit sehr hoch.
Für beliebige Kommunikationsnetze läßt sich allgemein der Begriff des γ–*konfluenten Umwegesystems* definieren. Verwendet man ein solches Umwegesystem, so kann man Kommunikationsanforderungen auf dem defekten Netzwerk mit Verzögerungsfaktor γ durchführen. Die Größe γ hängt im wesentlichen von der Länge des längsten Umweges und von der Zahl der Umwege ab, die über eine intakte Leitung geführt werden müssen.
Im Fall des defekten n-dim Würfels kann man bei Permutationsanforderungen zum Beispiel den Verzögerungsfaktor 3 erreichen, wenn man ein Umwegesystem mit Hilfe der Matching-Heuristik findet.

An dieser Stelle möchte ich mich besonders bei Herrn Prof. Dr. Günter Hotz, Herrn Dr. Hans Ulrich Simon, Herrn Dr. Bernd Becker und meinen Studienkollegen bedanken, die mir stets mit Rat und Tat zur Seite standen.

Kapitel I

Deterministisches Routing versus Probabilistisches Routing

1. Routing mit BITONIC-SORT

Gegeben sei folgende Aufgabenstellung:
Jeder Prozessor $v \in V$ im Kommunikationsnetz möchte ein Paket π_v an den Zielprozessor ziel(π_v) schicken. Wenn $V = \{\text{ziel}(\pi_v) | v \in V\}$ gilt, so spricht man von einer *Permutationsanforderung*. D.h. jeder Prozessor verschickt genau ein Paket, und jeder Prozessor erhält auch genau ein Paket.

Die Aufgabe, die Pakete bei einer Permutationsanforderung auf einem Kommunikationsnetz zu routen, entspricht einem parallelen Sortierproblem. Betrachtet man als Kommunikationsgraph den n-dim Würfel, so bietet sich z.B. das *Bitonische Sortieren* an, weil es auf dieser Netzstruktur sehr leicht zu parallelisieren ist. (Das *Bitonische Sortieren* wurde von Batcher [**Bat**] entwickelt.)

Zunächst der sequentielle Algorithmus:

Gegeben sei eine Folge $Z = z_0, z_1, \ldots, z_{2^n-1}$, die aufwärts sortiert werden soll. Zum bitonischen Sortieren benutzt man die rekursive Unterprozedur *bitonsort*, die bitonische Folgen sortieren kann. Die Hauptprozedur *sort* erzeugt aus Z eine bitonische Folge, indem die linke Hälfte aufwärts und die rechte Hälfte von Z abwärts sortiert wird. Ein Aufruf von *sort*$(Z, n, 0)$ sortiert eine beliebige Folge Z der Länge 2^n aufwärts und *sort*$(Z, n, 1)$ abwärts.

Bitonisches Sortieren

procedure bitonsort (Z, l, σ)
begin
$/*\ |Z| = 2^l$, $Z = z_0, z_1, \ldots, z_{2^l-1}$ biton

$$\sigma = \begin{cases} 0 & Z \text{ aufwärts sortieren} \\ 1 & Z \text{ abwärts sortieren} \end{cases} \quad */$$

$$\left.\begin{array}{l} X = x_0, x_1, \ldots, x_{2^{l-1}-1} \quad \text{mit} \quad x_i := z_{2i} \\ Y = y_0, y_1, \ldots, y_{2^{l-1}-1} \quad \text{mit} \quad y_i := z_{2i+1} \end{array}\right\} \text{für } 0 \leq i < 2^{l-1};$$

if $l > 1$
then bitonsort $(X, l-1, \sigma)$;
bitonsort $(Y, l-1, \sigma)$

fi;
if $\sigma = 0$
then $\left.\begin{array}{ll} z_{2i} & := \min\{x_i, y_i\} \\ z_{2i+1} & := \max\{x_i, y_i\} \end{array}\right\}$ für $0 \leq i < 2^{l-1}$
else $\left.\begin{array}{ll} z_{2i} & := \max\{x_i, y_i\} \\ z_{2i+1} & := \min\{x_i, y_i\} \end{array}\right\}$ für $0 \leq i < 2^{l-1}$
fi
end;

procedure sort (Z, l, σ)
begin
/* $|Z| = 2^l$, $Z = z_0, z_1, \ldots, z_{2^l-1}$ beliebig
$\sigma = \begin{cases} 0 & Z \text{ aufwärts sortieren} \\ 1 & Z \text{ abwärts sortieren} \end{cases}$ */
$\left.\begin{array}{l} X = x_0, x_1, \ldots, x_{2^{l-1}-1} \quad \text{mit} \quad x_i := z_i \\ Y = y_0, y_1, \ldots, y_{2^{l-1}-1} \quad \text{mit} \quad y_i := z_{i+2^{l-1}} \end{array}\right\}$ für $0 \leq i < 2^{l-1}$;
if $l > 1$
then sort $(X, l-1, \sigma)$;
sort $(Y, l-1, \overline{\sigma})$;
fi;
$Z := X \cdot Y$ mit $z_i := \begin{cases} x_i & \text{für} \quad 0 \leq i < 2^{l-1} \\ y_{i-2^{l-1}} & \text{für} \quad 2^{l-1} \leq i < 2^l \end{cases}$;
bitonsort (Z, l, σ)
end

Die Korrektheit des Bitonischen Sortierens kann mit Hilfe des *0-1–Prinzips* gezeigt werden. Den exakten Beweis kann man z.B. bei Knuth [**Knu**] nachlesen.

Setzt man $Z = ziel(\pi_0), ziel(\pi_1), \ldots, ziel(\pi_{2^n-1})$ als zu sortierende Folge an, und verwandelt man das rekursive sequentielle Programm in ein iteratives paralleles Programm, das in jedem Prozessor des n-dim Würfels in synchronisierter Art und Weise abgearbeitet wird, so erhält man folgenden Algorithmus, dessen Korrektheit sich unmittelbar aus dem sequentiellen Algorithmus ergibt, wenn man die Einzelprozessoren im Zusammenspiel betrachtet.

Zu Beginn des Sortiervorgangs besitzen alle Prozessoren also genau ein Paket und sie starten gleichzeitig mit der Abarbeitung ihrer Prozedur *parallelsort*. π_v bezeichne das Paket, das sich *momentan* am Knoten v aufhält. Die Variable σ eines jeden Knotens v steuert, ob v sich momentan in einer Teilfolge befindet, die auf- bzw. abwärts zu sortieren ist.

procedure parallelsort /* für Prozessor v */
begin
    $\sigma := \bigoplus_{i=1}^{n} v(i)$; /* Summe der Adressbits modulo 2 */
    for k **from** 1 **to** n
    do $\sigma := \sigma \oplus v(k)$;
        for d **from** k **downto** 1
        do /* $v \overset{d}{\longleftrightarrow} w$ */
            sende eine Kopie des Paketes π_v an den Knoten w über die ausgehende Kante der Dimension d und empfange *zur gleichen Zeit* vom Knoten w eine Kopie des Paketes π_w über die eingehende Kante der Dimension d;
            if $\sigma = v(d)$
              then if ziel$(\pi_v) >$ ziel(π_w) **then** $\pi_v := \pi_w$ **fi**
              else if ziel$(\pi_v) <$ ziel(π_w) **then** $\pi_v := \pi_w$ **fi**
            fi;
            lösche die Kopie von π_w
        od
    od
end

Laufzeit:

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} = O(n^2) \quad \text{Zeittakte}$$

Der Vorteil dieser deterministischen Routingstrategie für Permutationsanforderungen auf dem n-dim Würfel liegt in ihrer Einfachheit und der Tatsache, daß jeder Prozessor nur einen Puffer der Größe 2· Paketlänge besitzen muß. Des weiteren versendet und empfängt er pro Zeittakt nur über eine, genau festgelegte Leitung ein Paket.
Man beachte, daß zu jedem beliebigen Zeittakt alle Prozessoren in uniformer Weise nur Kanten ein und derselben Dimension zur Kommunikation benutzen. Die Reihenfolge, in der die Dimensionen gewählt werden, wird durch die doppelte **for**–Schleife gesteuert.
Der entscheidende Nachteil ist jedoch die Laufzeit von $\frac{1}{2}n \cdot (n+1)$ Zeiteinheiten. L.G. Valiant [**Val**] hat erstmalig einen probabilistischen Routingalgorithmus für Permutationsanforderungen auf dem n-dim Würfel untersucht. Seine Laufzeitanalyse zeigt, daß man eine mittlere Laufzeit von $c \cdot n$ erreichen kann und daß die Streuung der Laufzeiten extrem gering ist.
Er geht dabei allerdings davon aus, daß die Prozessoren des Netzwerkes gleichzeitig über alle eingehenden Leitungen Pakete empfangen und über alle ausgehenden Leitungen gleichzeitig je ein Paket verschicken können.

In einer Folgearbeit verallgemeinert Eli Upfal [**Upf**] die Methoden Valiants derart, daß er für eine ganze Klasse von Kommunikationsnetzen, die Balancierten Kommunikationsschemata, ein gutes Laufzeitverhalten bei der Anwendung probabilistischer Routingverfahren nachweisen kann. Die betrachteten Kommunikationsnetze müssen jedoch einen *konstanten Grad* haben, und in Upfals Modell können die Prozessoren des Netzes zwar über jede eingehende Leitung gleichzeitig Pakete empfangen, aber sie brauchen pro Zeiteinheit nur ein Paket über eine der ausgehenden Leitungen verschicken zu können.
Dieses Modell soll im folgenden auch benutzt werden. Die Prozessoren verfügen über eine interne Warteschlange, in die alle eintreffenden Pakete – dies können ja mehrere sein – eingeordnet werden. Pro Zeittakt soll jeweils das erste Paket der Warteschlange weiterverschickt werden. Dazu muß eine geeignete Warteschlangendisziplin gewählt werden.

2. Die Prioritätenregel und das Prinzip der kritischen Verzögerungsfolge

Es stellt sich die Frage, wie man das Laufzeitverhalten von randomisiertem Routing analysieren kann. Daß die Laufzeit im allgemeinen ungleich der Routenlänge sein wird, ist dadurch bedingt, daß oft mehrere Pakete zur gleichen Zeit einen Prozessor verlassen möchten. Da pro Zeiteinheit aber immer nur ein Paket einen Prozessor verlassen kann, verzögert sich die Ankunft der dort zurückgestellten Pakete an ihrem Zielknoten um mindestens eine Zeiteinheit. Im Konfliktfall, daß sich zu einem Zeitpunkt mehrere Pakete darum streiten als erstes einen Prozessor zu verlassen, erscheint es angebracht solche Pakete zu bevorzugen, die von ihrem Ziel noch am weitesten entfernt sind. Diese Strategie beschleunigt die Pakete, die noch einen weiten Weg vor sich haben, auf Kosten derer, die von ihrem Ziel nicht mehr so weit entfernt sind. Zu dieser Art der Konfliktauflösung benutzt man Prioritätszahlen, die von den Paketen mitgeführt werden und von den Prozessoren verändert werden können.

Bezeichne L die größte vorkommende Prioritätszahl und $p_\pi(u) \in \mathbf{N}$ die Priorität des Paketes π am Knoten u. Bewegt sich ein Paket π auf der Route

$$R(\pi): \; u_0 \to u_1 \to \ldots \to u_{i-1} \to u_i \ldots \to u_l \; ,$$

so soll die *Prioritätenregel* gelten:
Wenn zu einem Zeitpunkt mehrere Pakete an einem Knoten u sind, so wird ein Paket mit der *kleinsten* Prioritätszahl verschickt, und die Prioritätszahlen

nehmen auf der Reise des Paketes streng monoton zu:

$$p_\pi(u_{i-1}) < p_\pi(u_i) \quad \text{für} \quad i = 1, \ldots, l-1 \,.$$

Die Prioritätenregel erweist sich nun auch als wesentlicher Faktor für die Laufzeitanalyse.
Ein Paket π erreiche den Knoten u, dann wird es mit Priorität $p_\pi(u)$ in die Warteschlange eingereiht. Im schlimmsten Fall muß dieses Paket nun so lange auf seine nächste Übertragung warten, bis alle Pakete der Priorität $\leq p_\pi(u)$, die jemals den Knoten u passieren, weitergeschickt sind. Von besonderem Interesse sind also die folgenden Zahlen:

$f(p, u) :=$ Anzahl der Pakete, die den Knoten u mit Priorität p passieren.

Die Wartezeit eines Paketes π am Knoten u ist dann sicherlich beschränkt durch

$$\sum_{p \leq p_\pi(u)} f(p, u) \,.$$

Ein erster Ansatz zur Laufzeitanalyse könnte vielleicht zu folgender Abschätzung führen:
Wenn T die Laufzeit des Routings ist, dann exisitiert ein Paket π, das zum Zeitpunkt T seinen Zielknoten u_l erreicht.
Sei $R(\pi) : \; u_0 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_l$ die Route dieses Paketes, dann gilt:

$$T \leq \sum_{j=0}^{l-1} \sum_{p \leq p_\pi(u_j)} f(p, u_j) \,.$$

Der entscheidende Nachteil dieser Methode besteht darin, daß folgender Sachverhalt unberücksichtigt bleibt:
Ein Paket π treffe zu einem relativ späten Zeitpunkt mit großer Priorität am Knoten u ein, dann braucht es in der Regel nicht auf Pakete kleiner Priorität zu warten, weil diese ja schon vor dem Eintreffen des Paketes π bearbeitet wurden.
Das nun folgende Analyseverfahren, das auf Eli Upfal [**Upf**] zurückgeht, nutzt eben diese Eigenschaft aus.

Wenn ein Knoten u ein Paket π erst sehr spät übermitteln kann, so kann es dafür zwei Erklärungsmöglichkeiten geben:

1. Das Paket π ist sehr spät am Knoten u eingetroffen.
2. Bevor der Knoten u das Paket π bearbeiten konnte, mußte er sich zuerst um eine Menge anderer Pakete mit Priorität $\leq p_\pi(u)$ kümmern.

Um beide Gesichtspunkte gleichermaßen in die Analyse einfließen zu lassen, definiert man nun induktiv die sogenannte *kritische Verzögerungsfolge*

$$D = ((v_1, p_1), (v_2, p_2), \ldots, (v_l, p_l)) \ .$$

Wenn T die Laufzeit des Routings ist, dann gibt es einen Knoten v im Netz, der zum Zeitpunkt $T-1$ noch ein Paket π verschickt. Setze

$$v_l := v \ \ , \ \ p_l := p_\pi(v) \ \ , \ \ t_l := T - 1 \ .$$

Angenommen v_i, p_i und t_i sind bereits so definiert, daß gilt:
v_i verschickt zum Zeitpunkt t_i das letzte Paket mit Priorität $= p_i$, das ihn passiert.

Dann definiere, falls möglich, v_{i-1}, p_{i-1} und t_{i-1} wie folgt:
v sei ein Knoten unter den Vorgängern von v_i, der als einer der letzten ein Paket der Priorität $< p_i$ *an* v_i verschickt. Wenn dieses Paket den Knoten v mit Priorität p verläßt, dann sei t der Zeitpunkt, zu dem v das letzte Paket der Priorität p verschickt, das ihn passiert. v_i selbst verschicke zum Zeitpunkt t' sein letztes Paket mit Priorität $< p_i$. Dieses habe Priorität p'.

$$\begin{aligned} t_{i-1} &:= \max\{t, t'\} \\ (v_{i-1}, p_{i-1}) &:= \begin{cases} (v, p) & \text{falls} \quad t' \leq t \\ (v_i, p') & \text{falls} \quad t' > t \end{cases} \end{aligned}$$

Diese induktive Definition erhält offenbar folgende Invariante aufrecht:

Beh.:

Zum Zeitpunkt $t_{i-1} + 1$ gilt:
(1) Der Knoten v_i hat alle Pakete erhalten, die er mit Priorität $\leq p_i$ jemals verschicken muß.
(2) Alle Pakete der Priorität $< p_i$, die v_i passieren, sind schon verschickt.

Beweis:

(1) Spätestens ab dem Zeitpunkt $t_{i-1} + 1$ verschicken die Vorgänger von v_i nur noch Pakete mit Priorität $\geq p_i$ an v_i, denn dann hat auch der letzte unter ihnen seine Pakete mit Priorität $< p_i$ an v_i verschickt, die er an v_i zu schicken hat. Die Pakete, die einen Vorgänger von v_i mit Priorität $\geq p_i$ passieren, verschickt der Knoten v_i mit Priorität $> p_i$, wegen der Monotonieeigenschaft der Prioritätenregel. Deshalb hat v_i bereits zum Zeitpunkt $t_{i-1} + 1$ alle Pakete der Priorität $\leq p_i$ erhalten, die er jemals verschicken muß.

(2) 1.Fall: $v_{i-1} \neq v_i$

Angenommen v_i müßte zum Zeitpunkt $t_{i-1}+1$ oder später noch ein Paket der Priorität $< p_i$ verschicken, so würde dies zum Widerspruch $v_{i-1} = v_i$ führen.

Dieser Fall deckt die Möglichkeit ab, daß eine Verzögerung dadurch zustande kommt, daß v_i erst sehr spät von seinem Vorgänger v_{i-1} ein Paket zur Weitervermittlung bekommen kann.

2.Fall: $v_{i-1} = v_i$

Nach Definition des Zeitpunktes t_{i-1} hat v_i zum Zeitpunkt $t_{i-1}+1$ alle Pakete der Priorität $< p_i$ verschickt. Pakete der Priorität p_i können am Knoten v_i eine Verzögerung erfahren, weil sie auf eine Reihe von Paketen der Priorität $< p_i$ warten müssen.

■

Ist $t_{i-1} + 1 \leq t_i$, so verschickt der Prozessor v_i während des Zeitraumes $[t_{i-1}+1 : t_i]$ nur Pakete der Priorität p_i. Denn ab dem Zeitpunkt $t_{i-1}+1$ muß er nur noch Pakete der Priorität $\geq p_i$ verschicken. Da aber alle Pakete der Priorität $= p_i$ bereits angekommen sind, werden diese zuerst weitergeschickt. Deshalb gilt:

$$t_i - t_{i-1} \leq f(p_i, v_i) \ .$$

Ist $t_{i-1} + 1 > t_i$, so gilt trivialerweise:

$$t_i - t_{i-1} \leq 0 \leq f(p_i, v_i) \ .$$

Wenn die induktive Definition der kritischen Verzögerungsfolge D bei v_1 abbricht, so bedeutet das, daß v_1 niemals ein Paket der Priorität $< p_1$ verschicken muß. Deshalb kann v_1 bereits vom Zeipunkt 0 an Pakete der Priorität p_1 verschicken. Folglich gilt:

$$t_1 \leq f(p_1, v_1) - 1 \ .$$

Insgesamt ergibt sich:

$$\sum_{i=2}^{l} f(p_i, v_i) \geq \sum_{i=2}^{l} t_i - t_{i-1} = t_l - t_1 \geq T - 1 - (f(p_1, v_1) - 1)$$

und

$$T \leq \sum_{i=1}^{l} f(p_i, v_i) \qquad \text{für} \quad D = ((v_1, p_1), (v_2, p_2), \ldots, (v_l, p_l)) \ .$$

Das folgende Lemma faßt das Ergebnis zusammen.

Lemma I.1:

Gegeben sei ein Kommunikationsgraph $G = (V, E)$. Die Knoten des Graphen sollen Prozessoren repräsentieren, die pro Zeiteinheit über alle Eingangsleitungen Pakete empfangen und über *eine* der Ausgangsleitungen ein Paket verschikken können.
Erfordert das Routing von Paketen, das der *Prioritätenregel* gehorcht, auf diesem Netzwerk die Zeit T, dann folgt daraus die Existenz einer *kritischen Verzögerungsfolge*

$$D = ((v_1, p_1), (v_2, p_2), \ldots, (v_l, p_l))$$

mit $1 \leq p_{i-1} < p_i \leq L$
und $v_{i-1} \in \{v_i\} \cup \{v | (v, v_i) \in E, v \text{ schickt Paket der Priorität } p_{i-1} \text{ an } v_i\}$,
so daß

$$T \leq \sum_{i=1}^{l} f(p_i, v_i) \ ,$$

wobei L die größte vorkommende Priorität ist und $f(p_i, v_i)$ die Zahl der Pakete, die den Knoten v_i mit Priorität p_i passieren.
Die hier verwendete Definition der kritischen Verzögerungsfolge weicht von der *ursprünglichen* Definition Upfal's etwas ab. Mit Hilfe dieser Definition wird es möglich, eine probabilistische Routingstrategie auf dem *n-dim Würfel* zu analysieren, wie der folgende Abschnitt zeigen wird.
Die Existenz einer kritischen Verzögerungsfolge im Sinne Upfal's mit der entsprechenden Laufzeitschranke gewinnt man aber auch leicht aus aus Lemma I.1. Diese speziellere Form wird im Kapitel II benutzt werden.

Lemma I.2:

Erfordert das Routing von Paketen, das der *Prioritätenregel* gehorcht, auf dem Kommunikationsgraph $G = (V, E)$ die Zeit T, dann folgt daraus die Existenz einer *kritischen Verzögerungsfolge*

$$\begin{aligned} D &= (w_1, w_2, \ldots, w_L) \text{ mit } w_{p-1} \in \{w_p\} \cup \{w \in V | (w, w_p) \in E\} \ , \\ T &\leq \sum_{p=1}^{L} f(p, w_p) \ , \end{aligned}$$

wobei L die größte vorkommende Priorität ist und $f(p, w_p)$ die Zahl der Pakete, die den Knoten w_p mit Priorität p passieren.

Beweis:
Setze

$$w_p := \begin{cases} v_1 & \text{für} \quad 1 \leq p \leq p_1 \\ v_i & \text{für} \quad p_{i-1} < p \leq p_i \\ v_l & \text{für} \quad p_l \leq p \leq L \end{cases},$$

dann gilt wegen Lemma I.1:

$$T \leq \sum_{i=1}^{l} f(p_i, v_i) = \sum_{i=1}^{l} f(p_i, w_{p_i}) \leq \sum_{p=1}^{L} f(p, w_p) .$$

■

3. Laufzeitanalyse des randomisierten Routings auf dem n-dim Würfel

Der Kommunikationsgraph $G = (V, E)$ sei der n-dim Würfel, und die Knoten repräsentieren Prozessoren, die zwar mehrere Pakete gleichzeitig empfangen können, aber pro Zeiteinheit nur *ein* Paket verschicken. Es soll im folgenden gezeigt werden, daß trotz dieser "drastischen" Einschränkung das Valiant'sche Ergebnis aufrecht erhalten werden kann, welches besagt, daß Permutationsanforderungen auf dem n-dim Würfel mit hoher Wahrscheinlichkeit in $O(n)$ Zeiteinheiten geroutet werden können. Unter Verwendung der Beweisidee Upfals für randomisiertes Routing auf Balancierten Kommunikationsschemata wird sich auch eine Vereinfachung des Laufzeitbeweises für den n-dim Würfel ergeben.

Valiants 2-Phasen-Methode für randomisiertes Routing:
Wenn ein Prozessor $s \in V$ ein Paket π_s mit $\text{start}(\pi_s) = s$ und $\text{ziel}(\pi_s) = z$ verschicken möchte, so könnte man für jedes Paar $(s, z) \in V \times V$ eine fixe Route für das Paket π_s vorschreiben. Betrachtet man zum Beispiel die Gesamtheit aller Permutationsanforderungen auf dem vorgegebenen Kommunikationsnetz mit fixer Routenwahl, so ist die Laufzeit natürlich von der speziellen Permutation abhängig. Und es ist nicht auszuschließen, daß es Permutationen gibt, die eine außergewöhnlich schlechte Laufzeit bewirken.
In der Hoffnung, daß dies nur sehr wenige sind und mit dem Anspruch ein möglichst gleichverteiltes Laufzeitverhalten für alle Permutationsanforderungen zu gewinnen, hat Valiant eine in 2 Phasen gegliederte Strategie vorgeschlagen.

Für jedes Paket π_s mit $(\text{start}(\pi_s), \text{ziel}(\pi_s)) = (s, z)$ wird ein Zwischenziel $y \in V$ zufällig ermittelt, so daß gilt: $\forall\, v \in V : \Pr(y = v) = 1/|V|$.

1.Phase: Beförderung des Paketes π_s von s nach y
$R^1(\pi_s): \ s = u_0 \to u_1 \to \ldots \to u_k = y$
2.Phase: Beförderung des Paketes π_s von y nach z
$R^2(\pi_s): \ y = u_k \to u_{k+1} \to \ldots \to u_l = z$

Hierbei wird der 1.Teil $R^1(\pi_s)$ und der 2.Teil $R^2(\pi_s)$ der Route des Paketes π_s jeweils nach einer festen Vorschrift bestimmt.

Für die Bestimmung der Route zwischen den Knoten v und w in einer der beiden Phasen benutzt man auf dem n-dim Würfel folgende Methode:

$$v \xrightarrow{\alpha} w \ \text{ mit } \ \text{dist}(v,w) = l$$

Wenn $\{d_1, d_2, \ldots, d_l\} = \{j | v(j) \neq w(j)\}$ mit $d_1 < d_2 < \ldots < d_l$ ist, dann sei $\alpha = (d_1, d_2, \ldots, d_l)$.
Dies bewirkt, daß ein Paket auf seiner Reise in jeder der beiden Phasen die Dimensionen in aufsteigender Reihenfolge durchläuft.

Ein Paket π_s starte vom Knoten s und laufe in der 1.Phase zum Zwischenknoten y und von dort aus in der 2.Phase zum Zielknoten z. Die Vergabe der Prioritätszahlen für das Paket π_s auf seiner Route $R(\pi_s)$ geschieht wie folgt:

$$R(\pi_s): \ s = u_0 \xrightarrow{d_1} u_1 \xrightarrow{d_2} \ldots \xrightarrow{d_k} u_k = y = u_k \xrightarrow{d_{k+1}} u_{k+1} \xrightarrow{d_{k+2}} \ldots \xrightarrow{d_l} u_l = z$$

$$p_{\pi_s}(u_{i-1}) := \begin{cases} d_i & \text{für} \quad 1 \leq i \leq k \quad \text{1.Phase} \\ d_i + n & \text{für} \quad k+1 \leq i \leq l \quad \text{2.Phase} \end{cases}$$

Da die Dimensionen in jeder Phase in aufsteigender Reihenfolge durchlaufen werden, ergibt sich daraus die Monotonieeigenschaft der Prioritätenregel.

Beweisidee zur Laufzeitanalyse:

Den Grundstein zur Laufzeitanalyse liefert das Lemma des vorherigen Abschnitts. Wenn T die Laufzeit des Routings ist, so existiert eine kritische Verzögerungsfolge $D = ((v_1, p_1), \ldots, (v_l, p_l))$ mit $T \leq \sum_{i=1}^{l} f(p_i, v_i)$. Mit Hilfe dieser Aussage kann man die Wahrscheinlichkeit dafür abschätzen, daß $T \geq 2\gamma cn$ ist. (γ ist eine geeignet zu wählende Konstante > 1.)

$$\begin{aligned} & \Pr(T \geq 2\gamma cn) \\ \leq \ & \Pr\Big(\exists\, D = ((v_1,p_1),\ldots,(v_l,p_l)) \ \text{ mit } \ \sum_{i=1}^{l} f(p_i,v_i) \geq 2\gamma cn\Big) \\ \leq \ & \sum_{D \in DS} \Pr\Big(\sum_{i=1}^{l} f(p_i,v_i) \geq 2\gamma cn \ \text{ für } \ D = ((v_1,p_1),\ldots,(v_l,p_l))\Big)\,, \end{aligned}$$

wobei DS = Menge der möglichen kritischen Verzögerungsfolgen ist.
Um hier weiterzukommen, zählt man die möglichen kritischen Verzögerungsfolgen und schätzt die Wahrscheinlichkeit dafür ab, daß für eine beliebige aber fest vorgegebene kritische Verzögerungsfolge D die Summe $\sum_{i=1}^{l} f(p_i, v_i) \geq 2\gamma cn$ ist.
Sei also $D = ((v_1, p_1), \ldots, (v_l, p_l))$ gegeben, dann gilt es

$$\Pr\Big(\sum_{i=1}^{l} f(p_i, v_i) \geq 2\gamma cn\Big)$$

abzuschätzen. Die Zahlen $f(p_i, v_i)$ lassen sich wegen der probabilistischen Arbeitsweise nicht direkt angeben, aber man kann eine Aussage über die Mittelwerte machen. Betrachtet man die Größen $f(p_i, v_i)$ als Zufallsvariable, so muß man leider feststellen, daß diese voneinander abhängig sein können. Da die Anwendung von wahrscheinlichkeitstheoretischen Sätzen zumeist die Unabhängigkeit der betrachteten Ereignisse voraussetzt, versucht man das Ereignis

$$E = \Big(\sum_{i=1}^{l} f(p_i, v_i) \geq 2\gamma cn\Big)$$

durch Ereignisse mit unabhängigen Zufallsvariablen zu beschreiben. Ausgehend von der Unabhängigkeit der Routen der einzelnen Pakete definiert man:

$$\begin{aligned} h(\pi) &:= |\{i | \pi \text{ verläßt } v_i \text{ mit Priorität } p_i\}| \\ x(\pi) &:= \begin{cases} 1 & \text{falls } h(\pi) \geq 1 \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

als unabhängige Zufallsgrößen mit der Eigenschaft

$$\sum_{i=1}^{l} f(p_i, v_i) = \sum_{s=0}^{2^n-1} h(\pi_s) \ ,$$

denn die Variablen $h(\pi)$ sagen, wie oft ein Paket π auf die kritische Verzögerungsfolge D trifft, und die Variablen $x(\pi)$ geben an, ob ein Paket π überhaupt auf D trifft. Deshalb bezeichnet $\sum_{s=0}^{2^n-1} x(\pi_s)$ die Anzahl der Pakete, die auf D treffen und somit einen Beitrag zu $\sum_{s=0}^{2^n-1} h(\pi_s)$ leisten.

$$\begin{aligned} E &:= \Big(\sum_{s=0}^{2^n-1} h(\pi_s) \geq 2\gamma cn\Big) \\ E_1 &:= \Big(\sum_{s=0}^{2^n-1} x(\pi_s) \geq cn\Big) \\ E_2 &:= \Big(\sum_{s=0}^{2^n-1} h(\pi_s) \geq 2\gamma cn \ \text{ und } \ \sum_{s=0}^{2^n-1} x(\pi_s) < cn\Big) \end{aligned}$$

$$E \Rightarrow E_1 \vee E_2 \quad \text{also} \quad \Pr(E) \leq \Pr(E_1) + \Pr(E_2)$$

E_1 beschreibt das Ereignis, daß sehr viele Pakete auf D treffen und daß dadurch eine große Verzögerung verursacht wird. Bei E_2 hingegen geht man davon aus, daß die Verzögerung groß ist, obwohl nur wenige Pakete auf D treffen. E_2 ist nur möglich, wenn die wenigen Pakete, die auf D treffen, sich sehr lange auf D mitbewegen. Es bleibt also zu zeigen, daß die Wahrscheinlichkeit der beiden Ereignisse E_1 und E_2 verschwindend klein ist.

Bezeichnungen und Sätze aus der Wahrscheinlichkeitsrechnung:

$$B(m, N, q) = \sum_{i=m}^{N} \binom{N}{i} \cdot q^i \cdot (1-q)^{N-i}$$

bezeichnet die Wahrscheinlichkeit von mindestens m Erfolgen in einer Reihe von N *unabhängigen* Bernoulliversuchen mit Einzelwahrscheinlichkeit q.

Satz von Chernoff: [Che]

$$B(m, N, q) \leq \left(\frac{Nq}{m}\right)^m \cdot e^m - Nq \quad \text{für} \quad m \geq Nq \; .$$

Satz von Hoeffding: [Hoe]

Sei Z die Zahl der Erfolge in N unabhängigen Poissonversuchen mit Einzelwahrscheinlichkeiten $q_1, \ldots, q_N$ und $\overline{q} = \frac{1}{N} \cdot \sum_{i=1}^{N} q_i$ die *mittlere* Erfolgswahrscheinlichkeit, dann gilt:

$$\Pr(Z \geq m) \leq B(m, N, \overline{q}) \quad \text{falls} \quad m \geq N\overline{q} + 1 \; .$$

Eine Abschätzung für Binomialkoeffizienten:

$$\binom{N}{m} \leq \left(\frac{eN}{m}\right)^m$$

Ein Ergebnis aus der Kombinatorik:

$$\begin{aligned} \mid \{(a_1, \ldots, a_m) \mid \sum_{i=1}^{m} a_i = b\} \mid \; &= \; \binom{b+m-1}{m-1} \\ a_1, \ldots, a_m, b \; &\in \; \mathbf{N}_0 \end{aligned}$$

Die nun folgenden Lemmata ergänzen die Details zur Laufzeitanalyse:
Sei $G = (V, E)$ der Kommunikationsgraph des n-dim Würfels.

Lemma I.3:

Die Zahl der möglichen kritischen Verzögerungsfolgen $D = ((v_1, p_1), \ldots, (v_l, p_l))$ ist beschränkt durch:

$$|DS| \leq \frac{1}{2} \cdot 18^n \ .$$

Beweis:

Es gilt: $1 \leq p_{i-1} < p_i \leq 2n$ für $i = 2, \ldots, l$ wegen der Monotonieeigenschaft der Prioritätenregel, und weil die größte vorkommende Priorität $= 2n$ ist.
Die Zahl der verschiedenen l-Tupel $(p_1, \ldots, p_l)$ mit $1 \leq p_{i-1} < p_i \leq 2n$ ist gleich der Zahl der l-elementigen Teilmengen aus $\{1, \ldots, 2n\}$.

$$\begin{array}{lcll} v_{i-1} & = & v_i & \text{oder} \\ v_{i-1} & \xrightarrow{d_i} & v_i & \text{mit} \quad d_i = \begin{cases} p_{i-1} & \text{falls} \quad p_{i-1} \leq n \\ p_{i-1} - n & \text{falls} \quad p_{i-1} > n \end{cases} \end{array}$$

Bei gegebenem v_i gibt es also genau 2 Möglichkeiten für v_{i-1}. Für v_l können alle 2^n Knoten des Netzes in Frage kommen. Folglich gibt es $2^n \cdot 2^{l-1}$ verschiedene Möglichkeiten für $(v_1, \ldots, v_l)$ bei gegebenem $(p_1, \ldots, p_l)$.
Deshalb gilt:

$$|DS| \leq 2^{n-1} \cdot \sum_{l=0}^{2n} \binom{2n}{l} \cdot 2^l = \frac{1}{2} \cdot 2^n \cdot 3^{2n} = \frac{1}{2} \cdot 18^n \ .$$

■

Lemma I.4:

$\overline{f(p,v)}$ bezeichne die *mittlere* Anzahl der Pakete, die den Knoten v mit Priorität p verlassen, dann gilt:

$$\forall p : 1 \leq p \leq 2n \ , \ \forall v \in V : \overline{f(p,v)} = \frac{1}{2} \ ,$$

wenn eine *Permutationsanforderung* vorliegt.

Beweis:

Sei $v \in V$ beliebig.
Phase 1: $1 \leq p \leq n$

$$W = \{w \in V | \{j | w(j) \neq v(j)\} \subseteq \{1, \ldots, p-1\}\} \ \ , \ \ |W| = 2^{p-1}$$

ist die Menge der Knoten im n-dim Würfel, die ein Paket π abschicken können, das den Knoten v mit Priorität $p_\pi(v) = p$ verlassen kann.
Ein Paket, das von einem Knoten $w \in W$ startet, verläßt den Knoten v mit Priorität p mit Wahrscheilichkeit 2^{-p}. Also gilt:

$$\overline{f(p,v)} = 2^{p-1} \cdot 2^{-p} = \frac{1}{2} .$$

Phase 2: $n+1 \leq p \leq 2n$
symmetrisch zu Phase 1.

■

Lemma I.5:

$$\Pr(E_1) = \Pr\Big(\sum_{s=0}^{2^n-1} x(\pi_s) \geq cn\Big) \leq e^{-(c\ln c - c + 1)\cdot n}$$

Beweis:
$D = ((v_1,p_1),\ldots,(v_l,p_l))$ sei die kritische Verzögerungsfolge. Wegen der Unabhängigkeit der Routen darf man das Auftreffen der Pakete π_s auf D als unabhängige Poissonversuche auffassen. Die Einzelwahrscheinlichkeiten $q_s = \Pr(x(\pi_s) = 1)$ sind zwar unbekannt, aber man kann etwas über die mittlere Trefferwahrscheinlichkeit $\overline{q}$ aussagen. Dies eröffnet die Möglichkeit zur Anwendung des **Satzes von Hoeffding**.

$$\begin{aligned}
\Pr(x(\pi_s) = 1)) &\leq \sum_{i=1}^{l} \Pr(\pi_s \text{ verläßt } v_i \text{ mit Priorität } p_i) \\
\sum_{s=0}^{2^n-1} \Pr(x(\pi_s) = 1) &\leq \sum_{i=1}^{l} \sum_{s=0}^{2^n-1} \Pr(\pi_s \text{ verläßt } v_i \text{ mit Priorität } p_i) \\
&\leq \sum_{i=1}^{l} \overline{f(p_i,v_i)} \\
&\leq 2n \cdot \frac{1}{2} = n
\end{aligned}$$

$$\begin{aligned}
\Pr\Big(\sum_{s=0}^{2^n-1} x(\pi_s) \geq cn\Big) &\overset{\text{Hoeffding}}{\leq} B(cn, 2^n, \frac{n}{2^n}) \\
&\overset{\text{Chernoff}}{\leq} \left(\frac{2^n \cdot \frac{n}{2^n}}{cn}\right)^{cn} \cdot e^{cn-n} \\
&\leq e^{-(c\ln c - c + 1)\cdot n}
\end{aligned}$$

■

Lemma I.6:

$$\begin{aligned}\Pr(E_2) &= \Pr\Big(\sum_{s=0}^{2^n-1} h(\pi_s) \geq 2\gamma cn \text{ und } \sum_{s=0}^{2^n-1} x(\pi_s) < cn\Big) \\ &\leq e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn}\end{aligned}$$

Beweis:

$$\begin{aligned}h^{Ph}(\pi) &:= |\{i|\pi \text{ verläßt } v_i \text{ mit Priorität } p_i \text{ in Phase } Ph\}| \quad Ph = 1,2 \\ h(\pi) &= h^1(\pi) + h^2(\pi)\end{aligned}$$

Beh: $\forall \pi \ , \ Ph = 1,2 \ : \ \Pr(h^{Ph}(\pi) \geq a) \leq \left(\frac{1}{2}\right)^{a-1}$

Bew: π habe v_{i-1} mit Priorität p_{i-1} verlassen. π bleibt nur dann einen weiteren Schritt auf D, wenn es den Knoten $v_i \neq v_{i-1}$ mit Priorität p_i verläßt. Die Wahrscheinlichkeit hierfür beträgt $(1/2)^{p_i - p_{i-1}} \leq 1/2$. Wenn das Paket π die Verzögerungsfolge D einmal verlassen hat, so kann es frühestens in der nächsten Phase wieder auf D treffen.

$$\Pr(h^{Ph}(\pi) \geq a) \leq \left(\frac{1}{2}\right)^{a-1}$$

Der Beweis des Lemmas I.6 folgt in seinen Grundzügen ähnlichen Beweisen aus **[Meh1]** bzw. **[Sta]**, denen eine Überarbeitung (und Korrektur) der Arbeit Upfals zu verdanken ist.
Sei $\sum_{s=0}^{2^n-1} x(\pi_s) < cn$, und $s_1, \ldots, s_{cn-1}$ seien die Indizes der Pakete, die eventuell auf die Verzögerungsfolge D treffen. D.h:

$$\forall s \notin \{s_1, \ldots, s_{cn-1}\} : \quad x(\pi_s) = h^1(\pi_s) = h^2(\pi_s) = 0$$

$$\begin{aligned}\Pr(E_2) &= \Pr\Big(\sum_{s=0}^{2^n-1} \geq 2\gamma cn \text{ und } \sum_{s=0}^{2^n-1} x(\pi_s) < cn\Big) \\ &\leq \Pr\Big(\sum_{s=0}^{2^n-1} h(\pi_s) \geq 2\gamma cn \;\Big|\; \sum_{s=0}^{2^n-1} x(\pi_s) < cn\Big) \\ &\leq \Pr\Big(\sum_{j=1}^{cn-1} (h^1(\pi_{s_j}) + h^2(\pi_{s_j})) \geq 2\gamma cn\Big) \\ &\leq \Pr\Big(\sum_{j=1}^{cn-1} h^1(\pi_{s_j}) \geq \gamma cn\Big) + \Pr\Big(\sum_{j=1}^{cn-1} h^2(\pi_{s_j}) \geq \gamma cn\Big)\end{aligned}$$

$$\Big(\sum_{j=1}^{cn-1} h^{Ph}(\pi_{s_j}) \geq \gamma cn \Big)$$

$$\Longrightarrow \Big(\exists a_1, \ldots, a_{cn-1} \text{ mit } \sum_{j=1}^{cn-1} a_j = \gamma cn \text{ und } h^{Ph}(\pi_{s_j}) \geq a_j \geq 0 \Big)$$

$$\begin{aligned}
\Pr\Big(\sum_{j=1}^{cn-1} h^{Ph}(\pi_{s_j}) \geq \gamma cn \Big) &\leq \sum_{\substack{a_1,\ldots,a_{cn-1} \\ \sum_{j=1}^{cn-1} a_j = \gamma cn}} \prod_{j=1}^{cn-1} \Pr(h^{Ph}(\pi_{s_j}) \geq a_j) \\
&\leq \binom{cn + \gamma cn - 2}{cn-2} \cdot \left(\frac{1}{2}\right)^{\sum_{j=1}^{cn-1}(a_j - 1)} \\
&\leq \binom{cn + \gamma cn}{cn} \cdot \left(\frac{1}{2}\right)^{\gamma cn - cn + 1} \\
&\leq \frac{1}{2} \cdot \left(\frac{ecn(1+\gamma)}{cn}\right)^{cn} \cdot e^{-\ln 2 \cdot (\gamma cn - cn)} \\
&\leq \frac{1}{2} \cdot e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn}
\end{aligned}$$

$$\Pr(E_2) \leq e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn}$$

■

Satz I.1:
Das Kommunikationsnetzwerk des n-dim Würfels sei mit Prozessoren ausgerüstet, die pro Zeiteinheit über alle Eingangsleitungen gleichzeitig Pakete empfangen, aber *nur ein* Paket abschicken können. Dann gibt es eine probabilistische Routingstrategie für Permutationsanforderungen, deren Laufzeit T mit großer Wahrscheinlichkeit nur $O(n)$ beträgt. Genauer:

$$\forall\, S\ \exists\, R:\ \Pr(T \geq R \cdot n) \leq e^{-S \cdot n}\ .$$

Beweis: $\forall D = ((v_1, p_1), \ldots, (v_l, p_l)) \in DS$:

$$\begin{aligned}
&\Pr\Big(\sum_{i=1}^{l} f(p_i, v_i) \geq 2\gamma cn \Big) \\
\leq\ & \Pr(E_1) + \Pr(E_2) \\
\leq\ & e^{-(c \ln c - c + 1) \cdot n} + e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn}
\end{aligned}$$

Setze $\gamma := 6$.

$$\Pr\Big(\sum_{i=1}^{l} f(p_i, v_i) \geq 12cn\Big) \leq e^{-(c\ln c - c + 1)\cdot n} + e^{-1/2cn}$$

$$\begin{aligned} & \Pr(T \geq 12cn) \\ \leq & \sum_{D \in DS} \Pr\Big(\sum_{i=1}^{l} f(p_i, v_i) \geq 12cn \text{ für } D = ((v_1, p_1), \ldots, (v_l, p_l))\Big) \\ \leq & \frac{1}{2} \cdot 18^n \cdot 2 \cdot e^{-\min\{c/2,\, c\ln c - c + 1\}\cdot n} \end{aligned}$$

Bei gegebenem S wähle man R so, daß

$$S \leq \min\{R/24, R/12\ln(R/12) - R/12 + 1\} - 3\ ,$$

dann gilt:

$$\Pr(T \geq R \cdot n) \leq e^{-S\cdot n}\ .$$

■

Bem 1:

Der ursprüngliche Beweis von Valiant für das Laufzeitverhalten des n-dim Würfels bei Permutationsanforderungen ist von der benutzten Warteschlangendisziplin *unabhängig*. Aus dem obigen Beweis läßt sich dieses Resultat auch gewinnen, wenn man wie Valiant den Prozessoren erlaubt pro Zeiteinheit über alle ausgehenden Leitungen gleichzeitig Pakete zu verschicken.
Ordnet man allen ausgehenden Leitungen der Dimensionen 1 bis n je eine Warteschlange zu, so entspricht das bisherige Einsortieren der Pakete gemäß ihrer Priorität in die einzige Warteschlange nun einfach einem Sortieren durch Fachverteilung auf die n Warteschlangen, weil die Prioritätszahlen ja zu den durchlaufenen Dimensionen korrespondieren. Die bisherige Arbeitsweise läßt sich jetzt dadurch simulieren, daß jeweils die nicht-leere Warteschlange der kleinsten Dimension bedient wird. Darf man alle Warteschlangen gleichzeitig bedienen, so stellt sich sicherlich keine Verschlechterung der Laufzeit ein, und man erhält genau die Arbeitsweise von Valiant.

Bem 2:

Eli Upfal spricht in seiner Arbeit über die Balancierten Kommunikationsschemata von $\mu-$begrenzten Kommunikationsanforderungen, wenn jeder Prozessor Sender und Empfänger von höchstens μ Paketen ist. Da eine

solche Kommunikationsanforderung sich auf μ Permutationsanforderungen zurückführen läßt, kann man in analoger Weise zeigen, daß für die Laufzeit bei $\mu-$begrenzten Kommunikationsanforderungen auf dem n-dim Würfel gilt:

$$\forall\, S \;\; \exists\, R: \;\; \Pr(T \geq R \cdot \mu \cdot n) \leq e^{-S \cdot \mu \cdot n} \; .$$

Kapitel II

Fehlertolerantes Routing mit disjunkten Wegen

Die einfachste Methode, um die Übertragungssicherheit einer Nachricht zwischen Sender und Empfänger in einem Kommunikationsnetzwerk zu erhöhen, besteht wohl darin, mehrere Pakete mit dieser Nachricht vom Sender zum Empfänger zu schicken. Wenn man von nicht lokalisierbaren, aber statischen Fehlern ausgeht, so hat es wenig Sinn, mehrere Pakete über dieselbe Route zu schicken. Man sollte vielmehr die Pakete auf disjunkten Wegen zu ihrem Ziel befördern, um zu verhindern, daß mehrere Pakete mit identischer Nachricht durch ein und denselben Defekt abhanden kommen.
Eine Vervielfachung der Pakete führt zu einem höheren Kommunikationsaufkommen und wird im allgemeinen das Laufzeitverhalten des Routings nachteilig beeinflussen. Das n-dim Butterflynetz aus Kapitel I, das aus $n \cdot 2^n$ Prozessoren besteht, kann eine Permutationsanforderung mit hoher Wahrscheinlichkeit in Zeit $c \cdot n$ routen (siehe [**Upf**]). Die Arbeitsweise des n-dim Butterflynetzes läßt sich aber auch in einfacher Weise 1 zu 1 durch den n-dim Würfel simulieren, wenn man den Prozessoren erlaubt, über alle Eingangsleitungen Pakete gleichzeitig zu empfangen und über alle Ausgangsleitungen gleichzeitig zu versenden. Von daher ist es plausibel, daß ein solcher n-dim Würfel sogar eine $n-$fache Permutationsanforderung in der gleichen Zeit erledigen kann wie eine einfache.

Dies legt folgende Stategie nahe:
Es liege eine *Permutationsanforderung* auf dem n-dim Würfel vor. Wenn eine Nachricht M vom Knoten s zum Knoten z übermittelt werden soll, so packt der Knoten s diese Nachricht in n Pakete $\pi_{(1,s)}, \ldots, \pi_{(n,s)}$ und schickt diese auf *knotendisjunkten* Wegen zunächst zu einem zufällig gewürfelten Zwischenknoten y, in der Hoffnung, daß trotz fehlerhafter Übertragung mindestens ein Paket mit der Nachricht M bei y ankommt. y seinerseits verfährt nun genauso und schickt n Pakete mit der Nachricht M an das eigentliche Ziel z.

Zur Erleichterung des Laufzeitbeweises wird die gewünschte Arbeitsweise des n-dim Würfels zunächst in eine korrespondierende Arbeitsweise auf dem n-dim Butterflynetzwerk übersetzt. Dabei stellt sich heraus, daß man eine spezielle Permutationsanforderung für das n-dim Butterflynetzwerk erhält. Später simuliert der n-dim Würfel diese Arbeitsweise dann in geeigneter Form.

Die Vorgehensweise wird die folgende sein:

1. Definition der knotendisjunkten Wege auf dem n-dim Würfel
2. Definition der korrespondierenden Wege im assoziierten n-dim Butterflynetz
3. Laufzeitanalyse der daraus resultierenden Routingstrategie auf dem Butterflynetz
4. Simulation dieser Routingstrategie auf dem n-dim Würfel
5. Fehlertoleranzeigenschaften dieses Verfahrens

1. Definition der knotendisjunkten Wege im n-dim Würfel:

Sei	$\mathrm{Ad}(s) = (s(n), \ldots, s(1))$	die Adresse des Startknotens	
und	$\mathrm{Ad}(z) = (z(n), \ldots, z(1))$	die Adresse des Zielknotens.	
Seien	$d_1 < d_2 < \ldots < d_k$	die Indizes mit	$s(d_i) \neq z(d_i)$
und	$d'_1 < d'_2 < \ldots < d'_{n-k}$	die Indizes mit	$s(d'_i) = z(d'_i)$.

Dann ergeben sich folgende knotendisjunkte Wege zwischen s und z: (Ein Weg im n-dim Würfel wird durch die Abfolge der zu durchlaufenden Dimensionen charakterisiert.)
$1. - k$.ter Weg:

$$\begin{array}{rcccccc} \alpha^{d_1} & = & (\, d_1, & d_2, & \ldots & d_{k-1}, & d_k \;) \\ \alpha^{d_2} & = & (\, d_2, & d_3, & \ldots & d_k, & d_1 \;) \\ \vdots & & \vdots & & & & \vdots \\ \alpha^{d_k} & = & (\, d_k, & d_1, & \ldots & d_{k-2}, & d_{k-1} \;) \end{array}$$

$(k+1). - n$.ter Weg:

$$\begin{aligned} \alpha^{d'_i} &= (d'_i, \alpha^{m_i}, d'_i) \qquad \text{für } \; i = 1, \ldots, n-k \\ m_i &= \begin{cases} \min\{d_j | d_j > d'_i\} & \text{falls existent} \\ d_1 & \text{sonst} \end{cases} \end{aligned}$$

Man zeigt leicht, daß diese n Wege bis auf Start- und Zielknoten knoten- und somit auch kantendisjunkt sind. Des weiteren gilt:

$$\forall \;\; d = 1, \ldots, n \;\; : \;\; |\alpha^d| \leq n + 1$$

Diese n knotendisjunkten Wege auf dem n-dim Würfel lassen sich in kanonischer Weise auf das Butterflynetzwerk übertragen.

2. Definition der korrespondierenden Wege im n-dim Butterflynetz:

Jeder einzelne der n Knoten des Start-Superknotens s wählt einen Weg zum Ziel-Superknoten z. Aus jeder Schicht von s führt genau ein Weg über die 1-Kante heraus, und in jede Schicht von z mündet genau ein Weg über die 1-Kante ein.
Die Route des Paketes π, das vom Knoten (d, s) startet und zum Ziel-Superknoten z will, berechnet sich wie folgt:

$$R(\pi_{(d,s)}): \quad (d,s) \xrightarrow{\beta^d} (1 + d \bmod n, z) \qquad \text{für} \quad d = 1, \ldots, n$$

$$r(i) := s(i) \oplus z(i) \qquad \text{für} \quad i = 1, \ldots, n$$

$$\beta^d = (1, r(d+1), r(d+2), \ldots, r(n), r(1), \ldots, r(d-1), \overline{r(d)}) \in \{0,1\}^{n+1}.$$

D.h:

$$\beta^d(i) = \begin{cases} 1 & \text{für} \quad i = 0 \\ r((i+d) \bmod n) & \text{für} \quad 1 \le i \le n-1 \\ \overline{r(d)} & \text{für} \quad i = n\,. \end{cases}$$

Bem: Falls $\beta^d(l) = 1, \beta^d(l+1), \ldots, \beta^d(n) = 0$, so kann man die letzten $n-l$ Nullen streichen, denn diese stehen nur für eine Bewegung des Paketes π innerhalb des Superknotens z. Außerdem gilt nach wie vor, daß in jeder Schicht von z genau ein Paket von s ankommt.

Der Weg β^d auf dem Butterflynetz korrespondiert zu dem Weg α^d auf dem n-dim Würfel:

$$\alpha^d \leftrightarrow \beta^d \qquad \text{für} \quad d = 1, \ldots, n\,.$$

Die Abb.1 illustriert die Korrespondenz der knotendisjunkten Wege im 4-dim Würfel und im 4-dim Butterflynetzwerk an einem Beispiel:

$$\begin{array}{llcll} \alpha^{\text{gelb}} & = (1,2,3,4,1) & \leftrightarrow & \beta^{\text{gelb}} & = (1,1,1,1,1) \approx (1,1,1,1,1) \\ \alpha^{\text{rot}} & = (2,3,4) & \leftrightarrow & \beta^{\text{rot}} & = (1,1,1,0,0) \approx (1,1,1) \\ \alpha^{\text{blau}} & = (3,4,2) & \leftrightarrow & \beta^{\text{blau}} & = (1,1,0,1,0) \approx (1,1,0,1) \\ \alpha^{\text{grün}} & = (4,2,3) & \leftrightarrow & \beta^{\text{grün}} & = (1,0,1,1,0) \approx (1,0,1,1) \end{array}$$

Wird der Superknoten z von s zufällig ermittelt, so sind die letzten n Bits der Route β^d des Paketes $\pi_{(d,s)}$, das vom Knoten (d, s) startet, zufällig:

$$\Pr(\beta^d(i) = 1) = \frac{1}{2} \quad \text{für} \quad i = 1, \ldots, n\,.$$

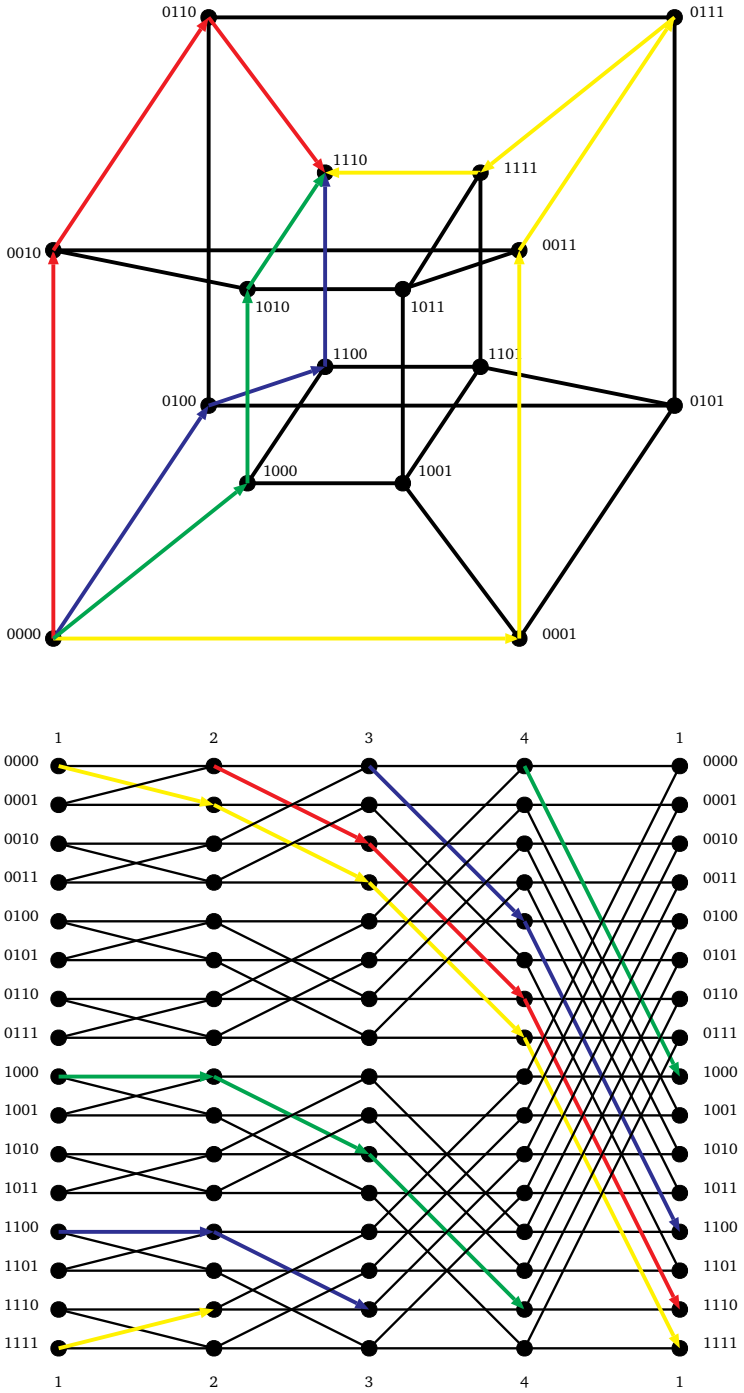

Abbildung 1: Illustration der Korrespondenz der disjunkten Wege im n-dim Würfel und im assoziierten n-dim Butterflynetzwerk.

Weiterhin gilt, daß die Routen der Pakete, die von unterschiedlichen Superknoten starten, voneinander *unabhängig* sind. Dies gilt natürlich nicht für die Pakete, die vom gleichen Superknoten starten.
Um das Routingschema zu vervollständigen, fehlt noch die Spezifikation der Prioritätenvergabe.
Sei

$$R(\pi): \quad (d,s) = u_0 \xrightarrow{1} u_1 \xrightarrow{\beta^d(1)} u_2 \xrightarrow{\beta^d(2)} \ldots \xrightarrow{\beta^d(n)} u_{n+1} = (1 + d \bmod n, z)$$

die Route eines Paketes π vom Superknoten s zum Superknoten z.

$$p_\pi(u_i) := i \quad \text{für} \;\; i = 0, \ldots, n\,.$$

D.h. die Priorität ist gleich der Länge des bereits zurückgelegten Weges.

3. Laufzeitanalyse:

In **[Upf]** wurde bereits gezeigt, daß das n-dim Butterflynetzwerk eine *Permutationsanforderung* mit großer Wahrscheinlichkeit in Zeit $O(n)$ erledigen kann. Dabei schickt jeder der $n \cdot 2^n$ Prozessoren ein Paket zu einem Zielprozessor. Im Laufzeitbeweis von Upfal wird allerdings die *Unabhängigkeit* aller Routen vorausgesetzt, was hier leider nicht gegeben ist. Trotzdem läßt sich der Beweis übertragen, wie im folgenden gezeigt wird.

Aufgrund der Symmetrie zwischen 1. und 2. Phase genügt es, sich auf die 1. Phase zu beschränken.
Die Routen der Pakete sind so geartet, daß alle Pakete im 1. Schritt eine 1-Kante benutzten. Nach Ablauf der 1. Zeiteinheit hält sich also an jedem Knoten genau ein Paket auf, das die weiteren n Schritte, wie oben definiert, durchläuft. Im folgenden wird nun nur noch dieser Teil des Routings untersucht. Sei T_{Ph1} die Laufzeit für die 1. Phase, dann ist es das Ziel, $\Pr(T_{Ph1} - 1 \geq \gamma cn)$ abzuschätzen.
Man benutzt auch hier das Beweisprinzip der *kritischen Verzögerungsfolge*.

Lemma II.1:

Die Zahl der möglichen kritischen Verzögerungsfolgen $D = (w_1, w_2, \ldots, w_n)$ ist beschränkt durch:

$$|DS| \leq \frac{1}{3} \cdot n \cdot 6^n\,.$$

Beweis:

n ist die größte vorkommende Priorität. Für w_n kommt jeder der $n \cdot 2^n$ Knoten des Netzes in Frage. Bei gegebenem w_j gibt es für w_{j-1} jeweils

3 Möglichkeiten, da jeder Knoten im Netz nur 2 Vorgänger hat. Also:

$$|DS| \leq n \cdot 2^n \cdot 3^{n-1} \, .$$

■

Lemma II.2:

$\overline{f(p,w)}$ bezeichne die *mittlere* Anzahl der Pakete, die den Knoten w mit Priorität p verlassen, dann gilt:

$$\forall p : 1 \leq p \leq n \, , \; \forall w \in V : \overline{f(p,w)} = 1 \, ,$$

wenn eine *Permutationsanforderung* vorliegt.

Beweis:

Sei $w \in V$ beliebig und $1 \leq p \leq n$.
Die Behauptung ist sicherlich richtig für $p = 1$, da genau 1 Paket jeden Knoten mit Priorität 1 verläßt. Pakete, die den Knoten w mit Priorität $p > 1$ verlassen, müssen genau $p-1$ Schichten vor w gestartet sein. Weil sie von unterschiedlichen Superknoten abstammen, sind ihre Routen *unabhängig* voneinander. Im binären Baum, der in der Wurzel w endet, gibt es 2^{p-1} Knoten, die $p-1$ Schichten vor w liegen. Ein Paket, das von einem dieser Knoten ausgeht, erreicht den Knoten w mit Wahrscheinlichkeit $(1/2)^{p-1}$.
Also gilt:

$$\overline{f(p,w)} = 2^{p-1} \cdot (1/2)^{p-1} = 1 \, .$$

■

Das nun folgende Lemma definiert *neue*, dem Problem angepaßte Zufallsvariable. Deren Eigenschaften gestatten es, mit den gleichen wahrscheinlichkeitstheoretischen Sätzen zu arbeiten, wie dies schon im Kapitel I geschehen ist.

Lemma II.3:

Sei $D = (w_1, w_2, \ldots, w_n)$ eine fest vorgegebene kritische Verzögerungsfolge, wie in Lemma I.2 definiert.

$x(s) :=$ Anzahl der Pakete, die vom Superknoten s starten, und auf D treffen, d.h. einen Knoten w_p mit Priorität p passieren.

Dann sind die $x(s)$ *unabhängige* Zufallsvariable und $x(s) \in \{0, 1\}$.

Beweis:

Die Unabhängigkeit der $x(s)$ folgt unmittelbar daraus, daß jeder Superknoten s zufällig seinen Zwischen-Superknoten z würfelt.

Sei $\pi_{(d,s)}$ das Paket, das vom Superknoten s aus Schicht d startet.

Ann: $\exists\ \pi_{(d,s)}, \pi_{(d',s)}$ mit $d \neq d'$, so daß
$\pi_{(d,s)}$ verläßt w_l mit Priorität l
$\pi_{(d',s)}$ verläßt w_k mit Priorität k

$$R(\pi_{(d,s)}): \quad (d,s) \;=\; u_0 \xrightarrow{1} u_1 \longrightarrow u_2 \longrightarrow \ldots \longrightarrow u_l \;=\; w_l \longrightarrow \ldots$$
$$R(\pi_{(d',s)}): \quad (d',s) \;=\; u'_0 \xrightarrow{1} u'_1 \longrightarrow u'_2 \longrightarrow \ldots \longrightarrow u'_k \;=\; w_k \longrightarrow \ldots$$

o.B.d.A. sei $k \leq l$
w_l liegt im binären Baum der Tiefe $n-1$, der von der Wurzel u_1 ausgeht, weil $|u_1 \longrightarrow w_l| = l - 1 \leq n - 1$.

$$\mathrm{Sch}(u_1) = 1 + d \bmod n \quad , \quad \mathrm{Ad}(u_1) = (s(n), \ldots, \overline{s(d)}, \ldots, s(1))$$

w_l liegt auch im binären Baum der Tiefe $n-1$, der von der Wurzel u'_1 ausgeht, weil $|u'_1 \longrightarrow w_k| = k - 1$ und $|w_k \longrightarrow w_l| \leq l - k$, als Teil der kritischen Verzögerungsfolge.

$$\mathrm{Sch}(u'_1) = 1 + d' \bmod n \quad , \quad \mathrm{Ad}(u'_1) = (s(n), \ldots, \overline{s(d')}, \ldots, s(1))$$

Dies ist ein Widerspruch, da die binären Bäume B_{u_1} mit Wurzel u_1 und $B_{u'_1}$ mit Wurzel u'_1 der Tiefe $n-1$ knotendisjunkt sind. Die Abb.2 verdeutlicht diesen Sachverhalt am Beispiel des 4-dim Butterflynetzes für den Fall $s = 0$ und $u_1, u'_1 \in \{(1,8),(2,1),(3,2),(4,4)\}$.

Bez:
$$1 \leq a, b \leq n: \quad \langle a : b \rangle_n = \begin{cases} [a:b] & \text{falls} \quad a \leq b \\ [1:b] \cup [a:n] & \text{falls} \quad a > b \end{cases}$$

Beh: $B_{u_1} \cap B_{u'_1} = \emptyset$

Ann: $\exists (i,v) \in B_{u_1} \cap B_{u'_1}$

$$(i,v) \in B_{u_1} \implies v(d) = \overline{s(d)}$$
$$(i,v) \in B_{u'_1} \implies v(d') = \overline{s(d')}\,,$$

da B_{u_1} und $B_{u'_1}$ nur Tiefe $n-1$ haben.

$$(i,v) \in B_{u_1} \text{ und } v(d') = \overline{s(d')} \implies i \in \langle 1 + d' \bmod n : d \rangle_n$$
$$(i,v) \in B_{u'_1} \text{ und } v(d) = \overline{s(d)} \implies i \in \langle 1 + d \bmod n : d' \rangle_n$$

Dies ist ein Widerspruch, da für $d \neq d'$ gilt:

$$\langle 1 + d' \bmod n : d \rangle_n \cap \langle 1 + d \bmod n : d' \rangle_n = \emptyset\,.$$

■

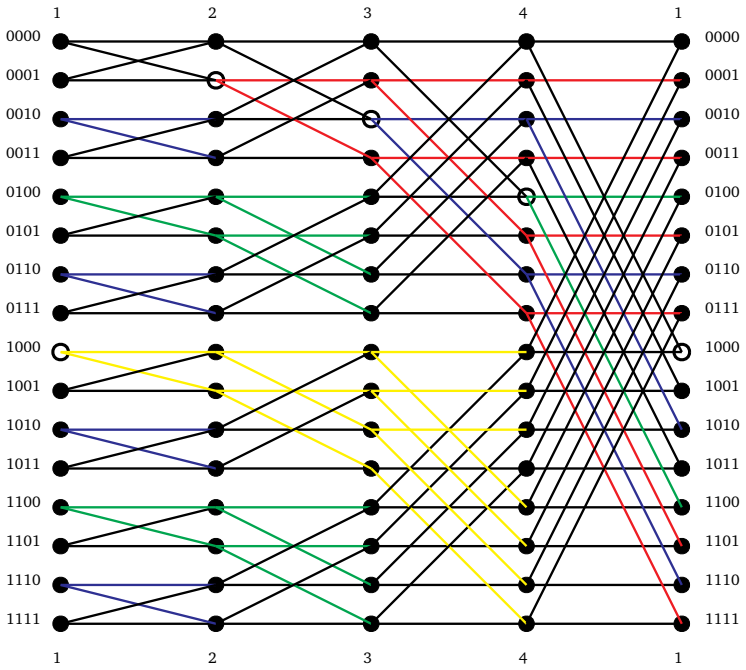

Abbildung 2: Knotendisjunkte Bäume im Butterflynetz

Lemma II.4:

$$\begin{aligned} E_1 &:= \left(\sum_{s=0}^{2^n-1} x(s) \geq cn\right) \\ \Pr(E_1) &\leq e^{-(c\ln c - c + 1)\cdot n} \end{aligned}$$

Beweis:

$\sum_{s=0}^{2^n-1} x(s)$ gibt die Anzahl der Pakete an, die auf die kritische Verzögerungsfolge treffen. Da die $x(s)$ unabhängige 0-1-wertige Zufallsvariable sind, kann man auch hier den Satz von Hoeffding über unabhängige Poissonversuche anwenden. Dazu muß man die *mittlere* Trefferwahrscheinlichkeit bestimmen:

$$\begin{aligned} x(s) = 1 &\iff \exists\, 1 \leq d, p \leq n :\ \pi_{(d,s)} \text{ verläßt } w_p \text{ mit Priorität } p \\ &\iff \exists\, 1 \leq p \leq n :\ \pi_{(d(p),s)} \text{ verläßt } w_p \text{ mit Priorität } p \\ &\quad \text{wobei} \quad d(p) = 1 + (n + \mathrm{Sch}(w_p) - p - 1) \bmod n, \end{aligned}$$

denn nur ein Paket, das p Schichten vor w_p startet, kann w_p mit Priorität p verlassen.

$$\Pr(x(s) = 1) \leq \sum_{p=1}^{n} \Pr(\pi_{(d(p),s)} \text{ verläßt } w_p \text{ mit Priorität } p)$$

Wegen Lemma II.2 gilt:

$$\sum_{s=0}^{2^n-1} \Pr(\pi_{(d(p),s)} \text{ verläßt } w_p \text{ mit Priorität } p) = 1\,.$$

Also gilt für die mittlere Trefferwahrscheinlichkeit:

$$\sum_{s=0}^{2^n-1} \Pr(x(s)=1) \leq \sum_{p=1}^{n} \sum_{s=0}^{2^n-1} \Pr(\pi_{(d(p),s)} \text{ verläßt } w_p \text{ mit Priorität } p) \leq n$$

$$\begin{aligned} \Pr\Big(\sum_{s=0}^{2^n-1} x(s) \geq cn\Big) &\overset{\text{Hoeffding}}{\leq} B(cn, 2^n, \frac{n}{2^n}) \\ &\overset{\text{Chernoff}}{\leq} \left(\frac{2^n \cdot \frac{n}{2^n}}{cn}\right)^{cn} \cdot e^{cn-n} \\ &\leq e^{-(c\ln c - c + 1)\cdot n} \end{aligned}$$

■

Lemma II.5:

$$h(\pi) := |\{p | \pi \text{ verläßt } w_p \text{ mit Priorität } p\}|$$

$$\begin{aligned} \Pr(E_2) &= \Pr\Big(\sum_{s=0}^{2^n-1}\sum_{d=1}^{n} h(\pi_{(d,s)}) \geq \gamma cn \text{ und } \sum_{s=0}^{2^n-1} x(s) < cn\Big) \\ &\leq e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma))\cdot cn} \end{aligned}$$

Beweis:

Beh: $\forall \pi \;:\; \Pr(h(\pi) \geq a) \leq \left(\frac{1}{2}\right)^{a-1}$

Bew: Die Behauptung ist sicherlich richtig für $a \leq 1$. Wenn das Paket π den Knoten w_{p-1} mit Priorität $p-1$ verläßt, dann erreicht es den Knoten w_p mit Wahrscheinlichkeit $\leq 1/2$. Hat ein Paket die kritische Verzögerungsfolge D erst einmal verlassen, so ist dies endgültig. Für die Wahrscheinlichkeit, daß ein Paket π a oder mehr Knoten der Verzögerungsfolge mit der entsprechenden Priorität passiert, gilt also:

$$\Pr(h(\pi) \geq a) \leq \left(\frac{1}{2}\right)^{a-1}\,.$$

Um die Wahrscheinlichkeit des Ereignisses E_2 abzuschätzen, kann man nun genauso wie im Lemma I.6 verfahren:
Unter der Bedingung, daß $\sum_{s=0}^{2^n-1} x(s) < cn$, ist die Zahl der Pakete, für die $h(\pi) > 0$, stark eingeschränkt. Die Pakete, die eventuell auf D treffen, mögen von den Knoten $(d_1, s_1), \ldots, (d_{cn-1}, s_{cn-1})$ gestartet sein. Die Zufallsvariablen $h(\pi_{(d_j,s_j)})$ sind *unabhängig* voneinander, da alle Pakete, die auf D treffen, von verschiedenen Superknoten gestartet sind: $s_i \neq s_j$ für $i \neq j$.

$$\begin{aligned} \Pr(E_2) &\leq \Pr(\sum_{j=1}^{cn-1} h(\pi_{(d_j,s_j)}) \geq \gamma cn) \\ &\leq \frac{1}{2} \cdot e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn} \end{aligned}$$

■

Satz II.1:
Sei T die Gesamtlaufzeit der probabilistischen 2-Phasen Routingstrategie mit knotendisjunkten Wegen auf dem Butterflynetzwerk. Dann gilt:

$$\forall\ S\ \exists\ R\ :\ \Pr(T \geq R \cdot n + 2) \leq e^{-S \cdot n}\,.$$

Beweis:

Für Phase 1 gilt:
$\forall\ D = (w_1, w_2, \ldots, w_n) \in DS\ :$

$$\sum_{s=0}^{2^n-1} \sum_{d=1}^{n} h(\pi_{(d,s)}) = \sum_{p=1}^{n} f(p, w_p)$$

$$\begin{aligned} & \Pr\Big(\sum_{p=1}^{n} f(p, w_p) \geq \gamma cn\Big) \\ \leq\ & \Pr(E_1) + \Pr(E_2) \\ \leq\ & e^{-(c \ln c - c + 1) \cdot n} + e^{-(\gamma \ln 2 - 1 - \ln 2 - \ln(1+\gamma)) \cdot cn} \end{aligned}$$

Setze $\gamma := 6$.

$$\Pr\Big(\sum_{p=1}^{n} f(p, w_p) \geq 6cn\Big) \leq e^{-(c \ln c - c + 1) \cdot n} + e^{-1/2cn}$$

$$\begin{aligned} & \Pr(T_{Ph1} \geq 6cn + 1) \\ \leq\ & \sum_{D \in DS} \Pr\Big(\sum_{p=1}^{n} f(p, w_p) \geq 6cn \text{ für } D = (w_1, w_2, \ldots, w_n)\Big) \\ \leq\ & \frac{1}{3} \cdot n \cdot 6^n \cdot 2 \cdot e^{-\min\{c/2,\, c \ln c - c + 1\} \cdot n} \end{aligned}$$

Aufgrund der Symmetrie zwischen 1. und 2. Phase gilt:

$$\Pr(T_{Ph1} \geq 6cn + 1) = \Pr(T_{Ph2} \geq 6cn + 1)$$

Die Gesamtlaufzeit $T = T_{Ph1} + T_{Ph2}$ läßt sich also wie folgt abschätzen:

$$\begin{aligned} \Pr(T \geq 12cn + 2) &\leq \Pr(T_{Ph1} \geq 6cn + 1) + \Pr(T_{Ph2} \geq 6cn + 1) \\ &\leq 2 \cdot \frac{2}{3} \cdot n \cdot 6^n \cdot e^{-\min\{c/2, c\ln c - c + 1\} \cdot n} \\ &\leq e^{-\min\{c/2, c\ln c - c + 1\} \cdot n + 3} \end{aligned}$$

Bei gegebenem S wähle man R so, daß

$$S \leq \min\{R/24, R/12\ln(R/12) - R/12 + 1\} - 3\ ,$$

dann gilt:

$$\Pr(T \geq R \cdot n + 2) \leq e^{-S \cdot n}\ .$$

■

4.Simulation der Routingstrategie auf dem n-dim Würfel:
Die Knoten $(1, v), (2, v), \ldots, (n, v)$ des Superknotens v des n-dim Butterflynetzwerkes werden durch den Knoten v im n-dim Würfel simuliert. Dazu braucht der Knoten v n interne Warteschlangen, die gleichzeitig bedient werden können. Die Verwaltung der Pakete in diesen Warteschlangen hat genau so zu geschehen, daß die Arbeitsweise des entsprechenden n-dim Butterflynetzwerkes nachgebildet wird.

5. Fehlertoleranzeigenschaften:
Ist der n-dim Würfel mit Prozessoren ausgestattet, die pro Zeiteinheit über alle einlaufenden Leitungen gleichzeitig empfangen und über alle auslaufenden Leitungen gleichzeitig senden können, so kann er mit Hilfe der randomisierten 2-Phasen Routingstrategie eine Permutationsanforderung mit großer Wahrscheinlichkeit in Zeit $O(n)$ erledigen, selbst wenn jede Nachricht in Paketform auf n disjunkten Wegen vom Sender zum Empfänger übertragen wird.
In welcher Weise wird durch diese Strategie die Fähigkeit erhöht, Fehler" zu tolerieren?

Ein einfaches Fehlermodell:
Die Übertragung einer Nachricht über eine Verbindungsleitung (Kante im Netz) kann gestört oder gar unmöglich sein. Es ist im folgenden nicht notwendig, daß ein sendender Prozessor feststellen kann, ob er von seinem Gegenüber empfangen wird. Pakete, die über eine defekte Kante geschickt

werden, sind als verloren anzusehen. Jede Kante sei mit der Ausfallwahrscheinlichkeit q behaftet. Zur Vereinfachung wird angenommen, daß es sich um statische Kantenfehler handelt. Modellhaft kann man von folgender Situation ausgehen: Vor Ausführung des Routings würfelt jede Kante, ob sie defekt ist oder nicht. Defekte Kanten bleiben während des gesamten Routings defekt, und intakte Kanten bleiben intakt. Im Mittel gibt es also $q \cdot n2^n$ defekte Kanten im Netz, die zufällig verteilt sind.
In diesem Modell läßt sich leicht ausrechnen, wie klein die Kantenausfallwahrscheinlichkeit q sein muß, damit *alle* Nachrichten mit großer Sicherheit ihr Ziel erreichen.
Die Analyse der Übertragungssicherheit der Nachrichten verläuft für beide Phasen des Routings analog:
Sei $(\pi_{(d,s)} \downarrow Ph1)$ das Ereignis, daß das d-te Paket, das vom Knoten s startet, aufgrund von Übertragungsfehlern oder Leitungsdefekten während der Phase 1 verlorengeht.

$$R(\pi_{(d,s)}) : \underbrace{s \longrightarrow z}_{\text{1.Phase}} \quad \text{für} \quad d = 1, \ldots, n$$

$(M_s \downarrow Ph1)$ bezeichne das Ereignis, daß die Nachricht des Knotens s ihr Ziel z während der 1.Phase nicht erreicht. Weil alle Pakete $\pi_{(1,s)}, \ldots, \pi_{(d,s)}$ die Nachricht M_s transportieren und ihre Routen kantendisjunkt sind, gilt:

$$\forall\, 0 \le s \le 2^n - 1 : \quad \Pr(M_s \downarrow Ph1) \le \prod_{d=1}^{n} \Pr(\pi_{(d,s)} \downarrow Ph1)$$

Sei $\mathrm{dist}(s, z) = k \ge 1$, dann gibt es k Pakete, deren Route die Länge k hat, und $n - k$ Pakete, die einen Weg der Länge $k + 2$ zurücklegen.
Die Wahrscheinlichkeit, daß ein Paket auf einem Weg der Länge l verschollen geht, ist $\le l \cdot q$, weil die Ausfallwahrscheinlichkeit für eine Kante im Netz q sein soll.

$$\begin{aligned} \Pr(M_s \downarrow Ph1) &\le (kq)^k \cdot ((k+2)q)^{n-k} \\ &\le (nq)^n \\ \Pr(M_s \downarrow) &\le \Pr(M_s \downarrow Ph1) + \Pr(M_s \downarrow Ph2) \\ &\le 2 \cdot (nq)^n \end{aligned}$$

$$\begin{aligned} \Pr(\exists 0 \le s \le 2^n - 1 :\ M_s \downarrow) &\le 2^n \cdot 2(nq)^n = 2 \cdot (2nq)^n \\ &\le 2e^{-S \cdot n} \quad \text{falls} \quad q \le \frac{1}{2ne^S} \ . \end{aligned}$$

D.h. um zu erreichen, daß mit Wahrscheinlichkeit $1-2e^{-Sn}$ alle Nachrichten ihr Ziel erreichen, muß die Ausfallwahrscheinlichkeit $q \leq 1/(2ne^S)$ sein.

Satz II.2:

Die randomisierte 2-Phasen Routingstrategie mit knotendisjunkten Wegen ist auf dem n-dim Würfel in der Lage, alle Nachrichten einer Permutationsanforderung mit Wahrscheinlichkeit $1-2e^{-Sn}$ an ihr Ziel zu befördern, wenn im Modell der statischen Kantenfehler die Ausfallwahrscheinlichkeit für eine Kante $\leq 1/(2ne^S)$ ist.

Bem: Die Laufzeit dieser Routingstrategie bleibt trotz des n-fachen Kommunikationsaufkommens linear in n (siehe Satz II.1). Dies ist insbesondere darauf zurückzuführen, daß ein Prozessor über alle ein- und ausgehenden Leitungen gleichzeitig Pakete empfangen bzw. senden kann. Die interne Belastung eines Prozessors nimmt aber schon um das n-fache zu.

Kapitel III

Fehlertolerantes Routing mit lokalen Umwegen

1. Deterministisches Routing mit lokalen Umwegen

Der im vorherigen Kapitel eingeschlagene Weg zur Erhöhung der Übertragungssicherheit von Nachrichten mittels disjunkter Übertragungswege scheint immer dann angebracht zu sein, wenn die defekten Stellen im Netz *nicht erkennbar* sind. In der Regel wird ein Prozessor jedoch beurteilen können, ob er über eine bestimmte Leitung mit dem Prozessor am anderen Ende in korrekter Weise kommunizieren kann. Unter der Annahme, daß nur Leitungsdefekte vorliegen, kann ein Prozessor v, der ein Paket über eine defekte Leitung $v \rightarrow w$ an w verschicken möchte, versuchen, dieses Paket über einen möglichst kurzen Umweg $U_{v,w}$ an w zu übermitteln. Dieses Vorgehen stellt eine Form der *adaptiven* Routenwahl dar, denn je nach der vorliegenden Fehlersituation kann ein Prozessor die Route eines Paketes so abwandeln, daß fehlerhafte Leitungen des Netzes *lokal* umgangen werden.
Untersucht man wieder das Routing von Permutationsanforderungen auf dem n-dim Würfel, so stellt sich die Frage, wie man die bewährten Routingstrategien so abwandeln kann, daß Pakete vor fehlerhaften Leitungen umgelenkt werden. Wie muß das Gesamtsystem der Umwege – falls ein solches überhaupt existiert – beschaffen sein, damit die resultierende Laufzeit akzeptabel bleibt?

Betrachtet man das deterministische Routing mit BITONIC-SORT auf dem n-dim Würfel (siehe Kapitel I), so erkennt man, daß zu jedem Zeittakt nur die Kanten einer bestimmten Dimension d zum Paketaustausch benutzt werden. Ist nun die Kante der Dimension d vom Knoten v zum Knoten w defekt, so kann man auf die dazu benachbarten Kanten der Dimension d ausweichen, um einen Pakettransfer von v nach w zu ermöglichen. Es ergeben sich $n-1$ kürzeste Umwege der Länge 3:

$$v \xrightarrow{d} w \qquad \text{sei defekt}$$

$$U^i_{v,w}: \quad v \xrightarrow{i} u_i \xrightarrow{d} u'_i \xrightarrow{i} w \qquad \text{für } \ 1 \leq i \leq n\ ,\ i \neq d$$

Sei q die Ausfallwahrscheinlichkeit für eine Kante im Modell der statischen Kantenfehler, dann stellt sich die Frage, wie groß die Wahrscheinlichkeit

dafür ist, daß zu jeder defekten Kante im n-dim Würfel ein intakter Umweg der obigen Form existiert.

$$\begin{aligned} \Pr(U^i_{v,w} \text{ defekt}) &\leq 3q \\ \Pr(U^i_{v,w} \text{ defekt } \forall i) &\leq (3q)^{n-1} \end{aligned}$$

$$\begin{aligned} &\Pr(\exists\, v \to w \text{ defekt und } U^i_{v,w} \text{ defekt } \forall i) \\ &\leq n2^n \cdot q \cdot (3q)^{n-1} \leq e^{-S \cdot n} \text{ falls } q \leq \frac{1}{7 \cdot e^S}\,. \end{aligned}$$

Wenn man die Kantenausfallwahrscheinlichkeit q auf $\leq 1/(7 \cdot e^S)$ beschränken kann, so existiert also mit hoher Sicherheit $1 - e^{-S \cdot n}$ ein Umweg der Länge 3 zu jeder defekten Kante im Netz.

Wenn zu jeder defekten Kante $v_i \xrightarrow{d} w_i$ zufällig ein Umweg $U^i_{v_i,w_i}$ gewählt wird, so kann es passieren, daß mehrere Umwege ein und dieselbe Umwegkante der Dimension d benutzen.
z.B:

$$U^i_{v_i,w_i}: \quad v_i \xrightarrow{i} u \xrightarrow{d} u' \xrightarrow{i} w_i \qquad i \neq d$$

Wenn $0 \leq \gamma \leq n-1$ Umwege gemeinsam die Kante $u \xrightarrow{d} u'$ benutzen, so dauert es im Fall von BITONIC-SORT, wegen des sich bildenden Staus auf dieser Kante, $\gamma + 2$ Zeiteinheiten, bis alle Knoten v_i ihre Pakete zu den w_i geschickt haben. Dies war vorher in einer Zeiteinheit zu schaffen.

Um eine solche Situation zu vermeiden, muß man die Auswahl der Umwege koordinieren. Die geringste Verzögerung erhält man offensichlich, wenn über jede intakte Kante $u \xrightarrow{d} u'$ höchstens ein Umweg $U^i_{v,w}: v \xrightarrow{i} u \xrightarrow{d} u' \xrightarrow{i} w$ führt. Wenn dies nicht möglich ist, so sollte man zumindest versuchen, die Anzahl der Umwege zu minimieren, die über intakte Kanten führen, um so die Zusatzbelastung möglichst gleichmäßig zu verteilen.

Ein Umwegesystem obiger Form für alle defekten Kanten im n-dim Würfel soll $\gamma-$*konjunkt* heißen, wenn über jede intakte Kante höchstens γ Umwege führen, die diese Kante als zweite Umwegkante benutzen.
Abb.3 zeigt ein 3-konjunktes und Abb.4 ein 1-konjunktes Umwegesystem auf dem 4-dim Würfel bei gleicher Verteilung der Kantenfehler. Die unterbrochen gezeichneten Kanten $(0 \to 2)$,$(9 \to 11)$ und $(5 \to 7)$ seien fehlerhaft. In Abb.3 ist $U_{0,2}: (0 \to 1 \to 3 \to 2)$, $U_{9,11}: (9 \to 1 \to 3 \to 11)$ und $U_{5,7}: (5 \to 1 \to 3 \to 7)$. In Abb.4 ist $U_{0,2}: (0 \to 4 \to 6 \to 2)$, $U_{9,11}: (9 \to 1 \to 3 \to 11)$ und $U_{5,7}: (5 \to 13 \to 15 \to 7)$.

Die Koordination der Umwege für defekte Kanten der Dimension d kann z.B. so geschehen, daß man versucht, über jede intakte Kante der Dimension

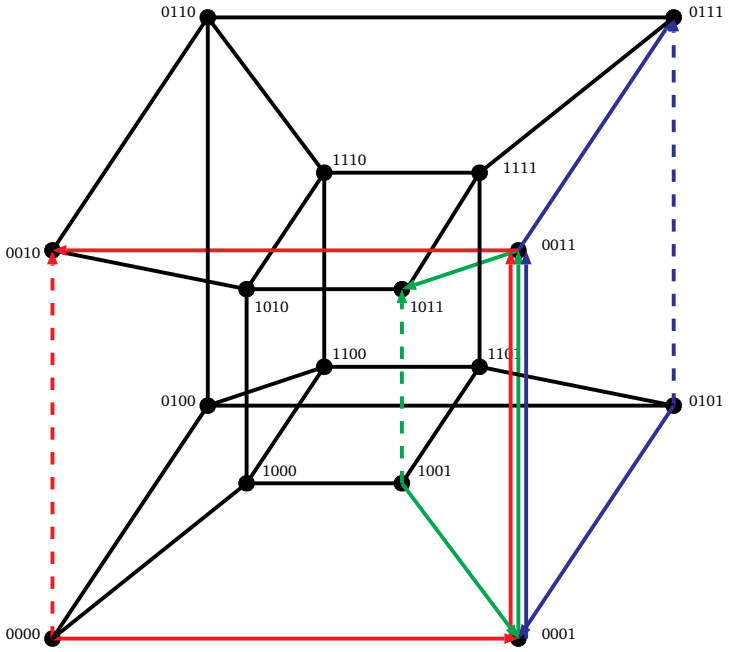

Abbildung 3:

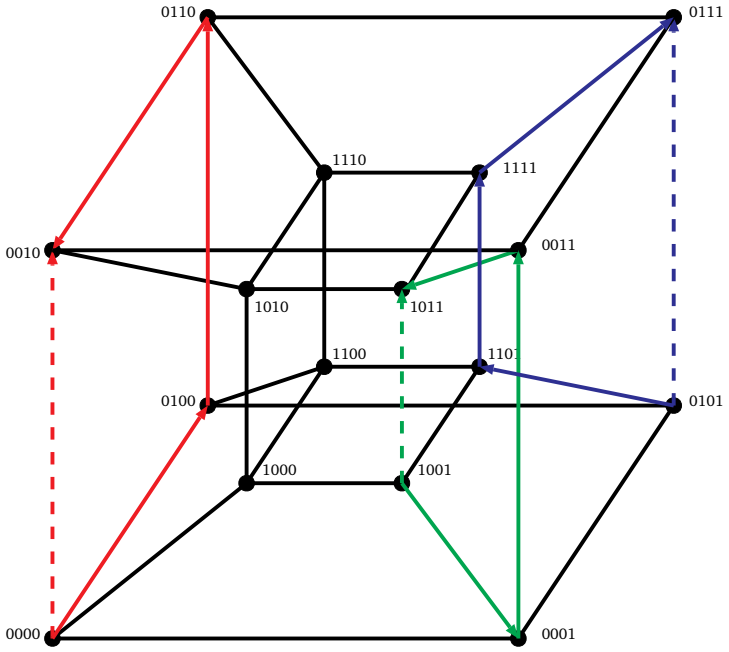

Abbildung 4:

d nur einen Umweg zu lenken. Dies führt zu folgendem *Perfect-Matching*-Problem auf dem bipartiten Graphen BG^d, dessen Knoten die Leitungen der Dimension d repräsentieren, und dessen Kanten andeuten, über welche Leitungen Umwege geführt werden können:

$$\begin{aligned} BG^d &= (BV^d, BE^d) \\ BV^d &= E^d_{int} \cup E^d_{def} \quad \text{mit} \quad E^d_{int} = E^d \cap E_{int} \text{ und } E^d_{def} = E^d \cap E_{def} \\ BE^d &= \{((v,w),(u,u')) \mid (v,w) \in E^d_{def} \text{ und } v \xrightarrow{i} u \xrightarrow{d} u' \xrightarrow{i} w \text{ intakt}\} \end{aligned}$$

Existiert auf dem Graphen BG^d für $d = 1, \ldots, n$ ein Matching, so hat man ein 1-konjunktes Umwegesystem gefunden. Da dies nicht immer der Fall sein muß, sollte man versuchen ein $\gamma-$konjunktes Umwegesystem mit kleinstmöglichem γ zu bestimmen. Dies kann z.B. wie folgt geschehen:

1. Konstruiere aus dem bipartiten Graphen BG^d einen Flußgraphen

   $$\begin{aligned} FG^d_m &= (FV^d, FE^d, s, t, c_m) \quad \text{mit} \\ FV^d &= BV^d \cup \{s,t\} \quad \text{mit} \quad s,t \notin BV^d \\ FE^d &= BE^d \cup \{s\} \times E^d_{def} \cup E^d_{int} \times \{t\} \end{aligned}$$

   s steht für die Quelle und t für die Senke des Flußgraphen. Die Quelle s wird mit allen Knoten des bipartiten Graphen verbunden, die defekte Leitungen der Dimension d repräsentieren und die Senke t mit allen Knoten für intakte Leitungen der Dimension d.

   Zur vollständigen Definition eines Flußgraphen gehört des weiteren eine Kapazitätsfunktion c_m.

   $$\begin{aligned} c_m : \quad & FE^d \to \mathbf{N} \\ \text{mit} \quad & c_m(e) = \begin{cases} 1 & \text{für} \quad e \in BE^d \cup \{s\} \times E^d_{def} \\ m & \text{für} \quad e \in E^d_{int} \times \{t\} \end{cases} \end{aligned}$$

2. Suche das kleinstmögliche m, das es noch erlaubt, einen Fluß der Größe $f = |E^d_{def}|$ von s nach t zu transportieren.

Die Konstruktion ist so beschaffen, daß der maximale Fluß $maxflow(FG^d_m) \leq |E^d_{def}|$ ist. Wenn für ein $1 \leq m \leq n$ gilt, daß $maxflow(FG^d_m) = |E^d_{def}|$ ist, dann impliziert das gleichzeitig, daß über jede intakte Kante der Dimension d höchstens m Umwege geführt werden müssen. Dies sind gerade die Umwege, repräsentiert durch eine Kante $e \in BE^d$, für die $f(e) = 1$ gilt. (Da die Kapazitäten $c_m(e) \in \mathbf{N}_0$ sind, läßt sich ein maximaler Fluß $f_{max}(e) \in \mathbf{N}_0$ bestimmen. Die wichtigsten Algorithmen für Flußprobleme findet man z.B.

in [**Meh2**].) Da $maxflow(FG^d_{m_1}) \leq maxflow(FG^d_{m_2})$ für $m_1 \leq m_2$, läßt sich das Minimum $\gamma^d = \min\{m | maxflow(FG^d_m) = |E^d_{def}|\}$ durch Binärsuche bestimmen. γ ergibt sich dann als $\gamma = \max\{\gamma^d | d = 1, \ldots, n\}$.

Setzt man zur Bestimmung eines $\gamma-$konjunkten Umwegesystems für den n-dim Würfel mit möglichst kleinem γ den oben beschriebenen Algorithmus ein, so hat das den Nachteil, daß man globale Information über den Zustand des Netzes braucht. Zudem wäre die Laufzeit eines sequentiellen Algorithmus relativ groß (exponentiell in n). Deshalb sollte es das Ziel sein, einen parallelen Algorithmus zu benutzen, der mit lokalem Wissen schnell ein geeignetes Umwegesystem bestimmen kann, indem er z.B. das oben beschriebene Matching-Problem löst, um ein 1-konjunktes Umwegesystem zu erhalten.

Da man aufgrund der zufälligen Verteilung der Kantenfehler ohnehin nicht gewährleisten kann, daß stets ein Umwegesystem existiert, mag es ausreichen, auf einen heuristischen Algorithmus zu vertrauen, der bei nicht allzu hoher Kantenausfallwahrscheinlichkeit fast immer ein solches Umwegesystem schnell bestimmen kann.

Jeder Knoten v, von dem eine defekte Kante $v \xrightarrow{d} w$ der Dimension d ausgeht, führt gemeinsam mit allen anderen Knoten des Netzes folgendes Programmstück aus, um einen Umweg $U_{v,w}$ für diese Kante zu finden.

/* $\forall\, u \xrightarrow{d} u'$ intakt: frei$(u, u') =$ **true** */
/* $v \xrightarrow{d} w$ defekt */

$U_{v,w} :=$ **nil**;
$i := 1 + d \bmod n$;
repeat
  if $v \xrightarrow{i} u_i \xrightarrow{d} u'_i \xrightarrow{i} w$ intakt **and** frei$(u_i, u'_i) =$ **true**
    then frei$(u_i, u'_i) :=$ **false**;
      $U_{v,w} := (v \xrightarrow{i} u_i \xrightarrow{d} u'_i \xrightarrow{i} w)$
  fi;
  $i := 1 + i \bmod n$
until $i = d$ **or** $U_{v,w} \neq$ **nil**

Wie groß ist die Wahrscheinlichkeit, daß zu einer defekten Kante $v \xrightarrow{d} w$ nach Ablauf dieses Programms noch kein intakter Umweg ($U_{v,w} =$ **nil**) gefunden wurde?

Sei $g(k)$ die Wahrscheinlichkeit, daß zu einer defekten Kante nach dem k-ten Schritt noch kein intakter freier Umweg gefunden ist, und $f(k)$ sei die

Wahrscheinlichkeit, daß eine intakte Kante nach dem k-ten Schritt reserviert ist. Dann gilt:

$$\begin{aligned} f(1) &\leq q\,, \\ f(k+1) &\leq f(k) + q \cdot g(k)\,, \end{aligned}$$

$$\begin{aligned} g(1) &\leq 3q\,, \\ g(k+1) &\leq g(k) \cdot (3q + f(k))\,. \end{aligned}$$

Denn eine intakte Kante ist nach $k+1$ Schritten reserviert, wenn sie nach k Schritten schon reserviert war *oder* wenn sie im $(k+1)$-ten Schritt reserviert wird. Dies geschieht mit Wahrscheinlichkeit $\leq q$, wenn es eine entsprechend benachbarte defekte Kante gibt, die in den k bisherigen Schritten noch keinen freien intakten Umweg gefunden hat.
Eine defekte Kante hat nach $k+1$ Schritten noch keinen freien intakten Umweg gefunden, wenn das nach k Schritten der Fall war *und* wenn sie im $(k+1)$-ten Schritt keinen solchen Umweg findet. Dies ist der Fall, wenn der entsprechende Umweg defekt ist, *oder* wenn dieser schon reserviert ist.

Beh: $\exists\, c, d, q \in \mathbf{R}$: $\quad g(k) \leq (cq)^k \quad$ und $\quad f(k) \leq dq$.

Bew: (durch Induktion nach k)

$k = 1$: Die Beh. ist richtig für $c \geq 3$ und $d \geq 1$.

$k \to k+1$:

$$\begin{aligned} g(k+1) &\leq (cq)^k \cdot (3q + dq) \leq (cq)^{k+1} \\ &\quad \text{falls} \quad 3 + d \leq c\,. \end{aligned}$$

$$\begin{aligned} f(j+1) - f(j) &\leq q \cdot (cq)^j \quad \text{für} \quad 1 \leq j \leq k\,, \\ f(k+1) - f(1) &= \sum_{j=1}^{k} (f(j+1) - f(j)) \leq q \cdot \sum_{j=1}^{k} (cq)^j\,. \end{aligned}$$

$$\begin{aligned} f(k+1) &\leq f(1) + q \cdot \sum_{j=1}^{\infty} (cq)^j \\ &\leq q + q \cdot \frac{cq}{1-cq} \leq \frac{q}{1-cq} \leq dq \\ &\quad \text{falls} \quad q \leq \frac{d-1}{dc}\,. \end{aligned}$$

Setze $c = 5$ und $d = 2$, dann gilt für $q \leq 1/10$:

$$g(k) \leq (5q)^k \quad \text{und} \quad f(k) \leq 2q\,.$$

■

$$
\begin{aligned}
\forall\ d &= 1,\ldots,n: \\
&\quad \Pr(\exists\ (v \to w) \in E^d_{def} \text{ und } U_{v,w} = \text{nil}) \\
&\leq\ q \cdot 2^n \cdot (5q)^{n-1} = \frac{1}{5} \cdot (10q)^n \ \text{ falls } \ q \leq \frac{1}{10}\ .
\end{aligned}
$$

$$
\begin{aligned}
&\Pr(\exists\ (v \to w) \in E_{def} \text{ und } U_{v,w} = \text{nil}) \\
\leq\ &\frac{n}{5} \cdot (10q)^n \leq (11q)^n \leq e^{-S \cdot n} \ \text{ falls } \ q \leq \frac{1}{11e^S}\ .
\end{aligned}
$$

Lemma III.1:
Wendet man den heuristischen Matching-Algorithmus nacheinander für alle Dimensionen $d = 1,\ldots,n$ an, so findet sich zu allen defekten Kanten mit Wahrscheinlichkeit $1 - e^{-S \cdot n}$ ein intakter Umweg, wenn die Kantenausfallwahrscheinlichkeit $q \leq 1/(11e^S)$ ist. Und das berechnete Umwegesystem ist 1-konjunkt.

Hat man erst einmal für einen n-dim Würfel mit Kantenfehlern ein $\gamma-$konjunktes Umwegesystem berechnet, so kann man das deterministische Routing mit BITONIC-SORT unter Beachtung der richtigen Synchronisation auf diesem Netzwerk mit Verzögerungsfaktor $\gamma + 2$ ausführen, weil für einen Pakettransfer zwischen zwei benachbarten Prozessoren nun statt einem Schritt bis zu $\gamma + 2$ Schritte benötigt werden.

Bem: In dieser Weise läßt sich jeder ASCEND-DESCEND-Algorithmus mit Verzögerungsfaktor $\gamma + 2$ auf einem defekten n-dim Würfel mit $\gamma-$-konjunktem Umwegesystem durchführen.

2. Probabilistisches Routing mit lokalen Umwegen

Im ersten Abschnitt dieses Kapitels wurde gezeigt, wie man auf einem n-dim Würfel trotz einiger Kantenfehler eine Permutationsanforderung routen kann. Dabei erwies sich die Methode der lokalen Umwege als nützlich, und die Hauptaufgabe bestand darin, für eine Koordination der Umwege zu sorgen, um so von vorne herein einen staufreien Paketfluß über die Umwege zu garantieren. Dieses ganze Konzept scheint auf den ersten Blick nur für einige spezielle Routingstrategien auf dem n-dim Würfel zugeschnitten zu sein,

die pro Zeittakt nur die Kanten einer festen Dimension zur Kommunikation benutzen.
Dieser Abschnitt versucht, das Prinzip der lokalen Umwege auch auf das probabilistische Routing auf dem n-dim Würfel – wie es in Kapitel I beschrieben ist – zu übertragen. Die Hoffnung besteht dabei darin, daß man mit den Umwegsystemen des vorherigen Abschnitts und dem probabilistischen Routing eine Laufzeit linear in n für Permutationsanforderungen erzielen kann, selbst wenn einige Verbindungsleitungen im Netz defekt sind.

Ein solches Ergebnis kann man mittels allgemeinerer Überlegungen erhalten. Diese gehen der Frage nach, welche Eigenschaften ein "brauchbares" Umwegesystem für ein beliebiges Kommunikationsnetz mit Leitungsdefekten haben sollte. Die Grundidee besteht wieder darin, die Arbeitsweise des intakten Netzes auf dem defekten Netz unter Benutzung der Umwege zu simulieren und möglichst wenig Zeit dabei zu verlieren.
Um die Simulation formal exakt beschreiben zu können, ist es notwendig genau festzulegen, wie die Prozessoren im intakten und defekten Netz beim Routen einer Kommunikationsanforderung arbeiten. Was die Arbeitsweise der Prozessoren im intakten Netz betrifft, so werden die in Kapitel I schon eingeführten Notationen benutzt:

$G = (V, E)$ sei ein beliebiger Kommunikationsgraph. Das Innenleben der Prozessoren sei so eingerichtet, daß eintreffende Pakete, die gleichzeitig über alle Eingangsleitungen empfangen werden können, in *eine einzige* Warteschlange gemäß ihrer Priorität eingereiht werden. Zu einem Zeittakt wird nur das erste Paket dieser Warteschlange abgeschickt. Eine Kommunikationsanforderung wird dadurch spezifiziert, daß für jedes zu transportierende Paket π eine Routenfunktion δ_π und eine Prioritätsfunktion p_π festgelegt wird. δ_π und p_π sind nur partiell auf der Knotenmenge V definiert. Man kann die Prioritätsfunktionen p ohne Schwierigkeit so wählen, daß gilt: $p_{\pi'}(v) \neq p_\pi(v)$ für $\pi' \neq \pi$ und $v \in V$. (Man wähle gegebenenfalls als neue Priorität das Paar aus alter Priorität und Paketnummer!) Dadurch ist bei jedem Schritt eindeutig bestimmt, welches Paket einen Prozessor verläßt.

$$
\begin{aligned}
R(\pi): \quad & v_0 \to v_1 \to \ldots \to v_{l-1} \to v_l \ , \\
& \text{start}(\pi) = v_0 \ , \ \text{ziel}(\pi) = v_l \ , \\
& \text{mit } (v_{i-1} \to v_i) \in E \text{ für } i = 1, \ldots, l \ .
\end{aligned}
$$

$$
\begin{aligned}
& \delta_\pi(v_{i-1}) = v_i \ \text{ für } \ i = 1, \ldots, l \quad ; \quad \delta_\pi(v_l) = x \notin V \ , \\
& p_\pi(v_i) = \ \text{Priorität des Paketes } \pi \text{ am Knoten } v_i
\end{aligned}
$$

Arbeitsweise:

$$\begin{aligned} M(w)(t) &= \text{Menge der Pakete, die sich zum Zeitpunkt } t \text{ in der} \\ &\quad \text{Warteschlange des Knotens } w \text{ befinden.} \\ \text{first}M(w)(t) &= \begin{cases} \pi & \text{falls } \forall\, \pi' \in M(w)(t) :\ p_{\pi'}(w) > p_\pi(w) \\ \diamond & \text{falls } M(w)(t) = \emptyset \end{cases} \end{aligned}$$

$$\begin{aligned} \forall\, w \in V :\ M(w)(t+1) &= M(w)(t) \setminus \{\text{first}M(w)(t)\} \cup \\ &\quad \{\pi | \exists v : \delta_\pi(v) = w \text{ und } \pi = \text{first}M(v)(t)\} \\ M(x)(t+1) &= M(x)(t) \cup \\ &\quad \{\pi | \exists v : \delta_\pi(v) = x \text{ und } \pi = \text{first}M(v)(t)\} \end{aligned}$$

Die Einführung des zusätzlichen Knotens $x \notin V$ dient lediglich zur Beschreibung des Umstandes, daß Pakete, die ihr Ziel erreicht haben, aus der Warteschlange des Zielknotens herausgenommen werden.
Wenn $M(w)(0)$ die Anfangsverteilung der Pakete zu Beginn des Routings beschreibt, dann läßt sich die Gesamtlaufzeit des Routings wie folgt charakterisieren:

$$T = \min\{t | M(x)(t) = \bigcup_{w \in V} M(w)(0)\}$$

Zerfällt die Kantenmenge $E = E_{int} \cup E_{def}$ in die intakten und defekten Kanten, so ist es das Ziel, ein Umwegsystem für die defekten Kanten zu schaffen, so daß das defekte Netz das intakte Netz Schritt für Schritt simulieren kann. Der Paketaustausch, der im intakten Netz während eines Zeittaktes stattfindet, wird im defekten Netz erst nach mehreren Schritten erreicht, da manche Pakete über die Umwege gelenkt werden müssen.
Eine günstige Situation liegt sicherlich dann vor, wenn die Umwege möglichst kurz und knotendisjunkt sind, denn dann kann es zu keinerlei Konflikten kommen, und man kann bei der Simulation offensichtlich einen Verzögerungsfaktor erreichen, der der Länge des längsten Umweges entspricht.
Bei der Simulation eines Schrittes kann ein fließender Pakettransfer ohne Staus aber auch dann gewährleistet werden, wenn Pakete gemeinsam benutzte Umleitungsstrecken zeitlich versetzt passieren, wie folgende Beispiele in Abb.5 verdeutlichen sollen:
Die farbigen Zahlen geben an, zu welchen Zwischenzeitpunkten ein Paket auf seinem Umweg die entsprechenden Knoten passiert. Die gleichfarbigen Zahlen entlang eines festen Umweges bilden eine monoton steigende Folge, und

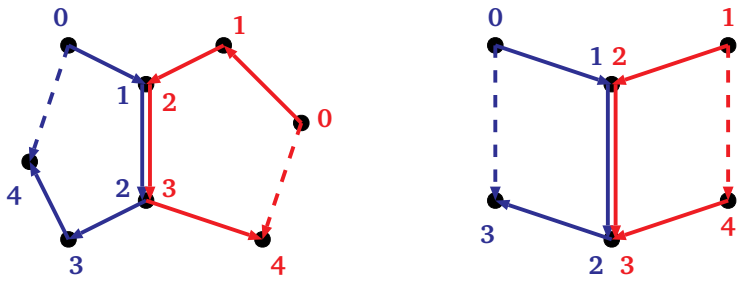

Abbildung 5:

die Zahlen, die sich auf einen Knoten beziehen, sind paarweise verschieden. Die größte vorkommende Zahl – im folgenden Dilatation genannt – entspricht dabei dem Verzögerungsfaktor bei der Simulation. Deshalb ist es das Ziel, ein Umwegesystem mit möglichst kleiner Dilatation zu finden.

Da die Prozessoren im intakten Netz pro Zeittakt höchstens *ein* Paket abschikken, lassen sich die Bedingungen für brauchbare Umwegsysteme noch etwas abschwächen:
Umwege, die vom gleichen Knoten starten, dürfen offensichtlich ein gemeinsames Anfangsstück haben (siehe Abb.6 (links)), da dieses Stück bei der Simulation eines Kommunikationsschrittes von höchstens einem Paket benutzt wird. Es stellt sich heraus, daß der hierzu symmetrische Fall, wie er in Abb.6 (rechts) angedeutet ist, ebenfalls zugelassen werden darf. Dies bereitet jedoch eine kleine Schwierigkeit: Wenn die Dilatation des Umwegesystems d ist, dann kann man bei der Simulation eines Schrittes nicht mehr fordern, daß alle Pakete, die im intakten Netz einen bestimmten Prozessor w erreichen, diesen Prozessor im defekten Netz auch nach d Zeittakten erreicht haben. Man kann aber sicherstellen, daß das Paket mit der kleinsten Priorität, das als nächstes den Prozessor w im intakten Netz verlassen wird, im defekten Netz beim Prozessor w rechtzeitig ankommt: Die Priorität eines Paketes auf seinem lokalen Umweg im defekten Netz wird mit der Priorität identifiziert, die dieses Paket am Endknoten des Umweges hat. Treffen mehrere Pakete gleichzeitig auf einen Umwegknoten, so wird – wie üblich – das Paket mit der kleinsten Priorität weitergeschickt, und die übrigen warten dort bis zum entsprechenden Zwischentakt des nächsten Simulationsschrittes.
Diese Überlegungen führen zum Begriff des *knoten-konfluenten Umwegesystems*:

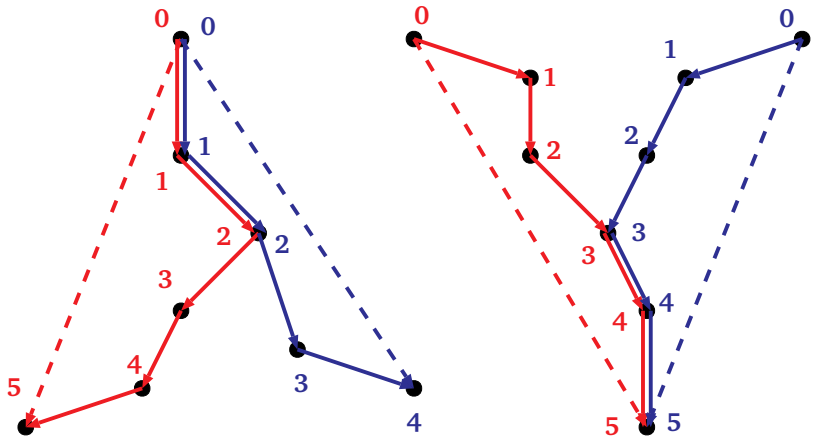

Abbildung 6:

(1) Knoten-konfluentes Umwegesystem mit Dilatation d:
Es sei

$$\begin{aligned} \varphi = (\varphi_0, \ldots, \varphi_d) &: E_{def} \to (V \cup \{\diamond\})^{d+1} \\ \varphi_f &: E_{def} \to V \cup \{\diamond\} \quad \forall\, 0 \le f \le d\ . \end{aligned}$$

Für $(v \to w) \in E_{def}$ seien $0 = f_0 < f_1 < \ldots < f_{l-1} < f_l = d$ die Indizes mit $\varphi_{f_i}(v \to w) = u_i \neq \diamond$, dann soll gelten:

$$(u_{i-1} \to u_i) \in E_{int} \text{ für } i = 1, \ldots, l \text{ mit } u_0 = v \text{ und } u_l = w\ .$$

D.h: Der Umweg zur defekten Kante $v \to w$ und die Zwischenzeitpunkte, zu denen eine Beförderung eines Paketes auf diesem Umweg stattfindet, werden durch die Folge $(v, 0) = (u_0, f_0), (u_1, f_1), \ldots, (u_l, f_l) = (w, d)$ beschrieben.

Definition:
φ heißt *knoten-konfluent*, wenn folgende Eigenschaft erfüllt ist:
$\forall (v \to w), (v' \to w') \in E_{def}$:

$$\begin{aligned} &\exists\, f: \quad \varphi_f(v \to w) = \varphi_f(v' \to w') \neq \diamond \\ &\Longrightarrow \quad \varphi_i(v \to w) = \varphi_i(v' \to w') \quad \forall\, 0 \le i \le f \text{ oder } \forall f \le i \le d \end{aligned}$$

Hat man für ein Netzwerk mit Kantenfehlern ein *knoten-konfluentes* Umwegesystem gefunden, so ist es möglich, das korrespondierende intakte Netzwerk

durch das defekte Netzwerk mit Verzögerungsfaktor d zu simulieren. Die Route eines Paketes π, im intakten Netz durch die Funktion δ_π charakterisiert, muß im defekten Netz durch eine neue Übergangsfunktion Δ_π ersetzt werden, die das Paket richtig über die Umwege leitet. Außerdem müssen die Prioritätszahlen auf den Umwegen spezifiziert werden.

Wenn $\delta_\pi(v) = w$ ist, und $0 = f_0 < f_1 < \ldots < f_{l-1} < f_l = d$ die Indizes sind mit $\varphi_{f_i}(v \to w) = u_i \neq \diamond$ für $i = 0, \ldots, l$, wobei $u_0 = v$ und $u_l = w$, dann sei

$$\begin{aligned} \Delta_\pi(u_{i-1}, f_{i-1}) &= (u_i, f_i) \quad \text{für } i = 1, \ldots, l \\ \Delta_\pi(\text{ziel}(\pi), 0) &= (x, 0) \quad \text{wobei } x \notin V \\ p_\pi(u_i) &= p_\pi(w) \quad \text{für } i = 1, \ldots, l-1\ . \end{aligned}$$

Jeder Prozessor u im defekten Netz sei mit d Warteschlangen ausgestattet: $\mathcal{M}(u, 0), \ldots, \mathcal{M}(u, d-1)$. (Zur Vereinfachung der Schreibweise soll stets gelten: $\mathcal{M}(u, d) = \mathcal{M}(u, 0)$.) Die Funktion Δ_π beschreibt nicht nur die Route des Paketes π, sondern gibt gleichzeitig an, in welche Warteschlange das Paket bei seiner Ankunft an einem Knoten entsprechend seiner Priorität eingeordnet werden soll.

Arbeitsweise:

$$\mathcal{M}(u, f)(t) = \text{Menge der Pakete, die sich zum Zeitpunkt } t \text{ in der } f\text{-ten Warteschlange des Knotens } u \text{ befinden.}$$

$t \equiv f \bmod d$:
$\forall\, u, u' \in V$:

$$\begin{aligned} \mathcal{M}(u, f)(t+1) &= \mathcal{M}(u, f)(t) \setminus \{\text{first}\mathcal{M}(u, f)(t)\} \\ \mathcal{M}(u', f')(t+1) &= \mathcal{M}(u', f')(t) \cup \\ &\quad \{\pi | \exists (u, f) : \Delta_\pi(u, f) = (u', f') \text{ und } \pi = \text{first}\mathcal{M}(u, f)(t)\} \end{aligned}$$

$t \equiv 0 \bmod d$:
$x \notin V$:

$$\begin{aligned} \mathcal{M}(x, 0)(t+1) &= \mathcal{M}(x, 0)(t) \cup \\ &\quad \{\pi | \exists u : \Delta_\pi(u, 0) = (x, 0) \text{ und } \pi = \text{first}\mathcal{M}(u, 0)(t)\} \end{aligned}$$

Jeder Prozessor bedient nacheinander seine d Warteschlangen, indem er das jeweils erste Paket einer jeden Warteschlange abschickt. Zu den Zeitpunkten $t \equiv f \bmod d$ wird die f-te Warteschlange bedient.

Es bleibt zu zeigen, daß diese Strategie auf dem defekten Netz mit knotenkonfluentem Umwegesystem gewährleisten kann, daß die Gesamtlaufzeit für das Routing gegenüber der Laufzeit auf dem intakten Netz nur um den Faktor d anwächst.
Zu diesem Zweck wird nun der Kommunikationsgraph $\mathcal{G}_\varphi = (\mathcal{V}, \mathcal{E})$ definiert: Ein Knoten aus $\mathcal{V}$ stellt eine der d Warteschlangen eines Knotens aus V dar. Die Kanten verdeutlichen, in welcher Weise sich ein Paket entsprechend dem konfluenten Umwegesystem von einer Warteschlange eines Knotens zu einer Warteschlange eines anderen Knotens bewegen kann.

$$
\begin{aligned}
\mathcal{G}_\varphi &= (\mathcal{V}, \mathcal{E}) \\
\mathcal{V} &= V \times \{0, \ldots, d-1\} \\
\mathcal{V}_w &= \{(u,f) \,|\, \exists\, (v \to w) \in E :\ \varphi_f(v \to w) = u \neq \diamond\,,\ 0 < f < d\} \\
\mathcal{E} &= \bigcup_{w \in V} \mathcal{E}_w \\
\mathcal{E}_w &= \{(u,f) \to (u',f') \,|\, \exists (v \to w) \in E\,,\, 0 \leq f < f' \leq d : \\
&\qquad \varphi_f(v \to w) = u\,,\ \varphi_{f'}(v \to w) = u'\,,\ \varphi_i(v \to w) = \diamond \text{ für } f < i < f'\}
\end{aligned}
$$

Ist $(u,f) \in \mathcal{V}_w$, so liegt der Knoten u auf einem Umweg mit dem Endknoten w, und Pakete, die diesen Umweg benutzen, verlassen den Knoten u zum Zwischenzeitpunkt f. Der Ingrad eines Knotens $(u,f) \in \mathcal{V}$ im Graphen $\mathcal{G}_\varphi$ gibt die Anzahl der Umwege an, die über den Knoten u führen, und auf denen Pakete den Knoten u zum Zwischentakt f passieren. (Für den Knoten mit der Markierung 3 aus Abb.6 (rechts) gilt z.B: indeg > 1.)

$$
\begin{aligned}
\mathcal{V}_w &= \mathcal{Y}_w \cup \mathcal{I}_w \\
\mathcal{Y}_w &= \{(u,f) \in \mathcal{V}_w | \operatorname{indeg}(u,f) > 1\} \\
\mathcal{I}_w &= \{(u,f) \in \mathcal{V}_w | \operatorname{indeg}(u,f) = 1\}
\end{aligned}
$$

Stellt man einem intakten Kommunikationsnetzwerk $G = (V,E)$ ein korrespondierendes defektes Netzwerk mit konfluentem Umwegesystem, repräsentiert durch den Graphen $\mathcal{G}_\varphi = (\mathcal{V}, \mathcal{E})$, gegenüber, so ergeben sich beim Routing ein und derselben Kommunikationsanforderung folgende Invarianten, wenn man die oben definierten Arbeitsweisen zugrundelegt:
Lemma III.1:
Zum Zeitpunkt $t = 0$ gelte:

$$
\begin{aligned}
\forall\, u \in V\,,\, 0 < f < d : \quad M(u)(0) &= \mathcal{M}(u,0)(0) \\
\emptyset &= \mathcal{M}(u,f)(0)
\end{aligned}
$$

Dann ergibt sich zu späterer Zeit folgende Verteilung der Pakete auf die Warteschlangen, wenn man das intakte mit dem defekten Netzwerk vergleicht:

(i) $$M(w)(t) = \mathcal{M}(w,0)(d\cdot t) \cup \bigcup_{(u,f)\in\mathcal{Y}_w} \mathcal{M}(u,f)(d\cdot t)$$

(ii) $$\emptyset = \mathcal{M}(u,f)(d\cdot t) \qquad \forall\ (u,f) \in \mathcal{I}_w$$

(iii) $$\mathrm{first}M(w)(t) = \mathrm{first}\mathcal{M}(w,0)(d\cdot t)$$

Bem:
Wegen der Konfluenzeigenschaft von φ und der Vergabe der Prioritätszahlen auf den Umwegen gilt:

$$\begin{aligned}&(u,f) \in \mathcal{Y}_w\ ,\ \pi \in \mathcal{M}(u,f)(t')\\ &\Longrightarrow \exists m\ \Delta_\pi^m(u,f) = (w,0) \text{ und } p_\pi(u) = p_\pi(w)\ .\end{aligned}$$

Aufgrund der Arbeitsweise ergibt sich:

$$\begin{aligned}\mathcal{M}(w,0)(t') &\subseteq \mathcal{M}(w,0)(t'+1) \text{ für } d\cdot t < t' < d\cdot t + d\ ,\\ \mathcal{M}(u,f)(t') &\subseteq \mathcal{M}(u,f)(t'+1) \text{ für } d\cdot t < t' < d\cdot t + f\ .\end{aligned}$$

Bew: (durch Induktion nach t)
$t = 0$:
(i),**(ii)** und **(iii)** gelten, weil sowohl für das intakte als auch für das defekte Netzwerk die gleiche Kommunikationsanforderung vorliegt.

$t \to t+1$:

$$\begin{aligned}M(w)(t+1) &= M(w)(t) \setminus \{\mathrm{first}M(w)(t)\} \cup\\ &\quad \{\pi | \exists v : \delta_\pi(v) = w \,\mathrm{und}\, \pi = \mathrm{first}M(v)(t)\}\\ &= \Big(\mathcal{M}(w,0)(d\cdot t) \cup \bigcup_{(u,f)\in\mathcal{Y}_w} \mathcal{M}(u,f)(d\cdot t)\Big) \setminus \{\mathrm{first}\mathcal{M}(w,0)(d\cdot t)\}\\ &\quad \cup \{\pi | \exists v : \delta_\pi(v) = w \text{ und } \pi = \mathrm{first}\mathcal{M}(v,0)(d\cdot t)\}\\ &= \mathcal{M}(w,0)(d\cdot t + d) \cup \bigcup_{(u,f)\in\mathcal{Y}_w} \mathcal{M}(u,f)(d\cdot t + d)\end{aligned}$$

Das erste Gleichheitszeichen ergibt sich aus der Arbeitsweise des intakten Netzes, das zweite folgt aus der Induktionsannahme, und das dritte kann man wie folgt einsehen:

Sei $\pi = \text{first}\mathcal{M}(v,0)(d \cdot t)$ mit $\delta_\pi(v) = w$ und $\Delta_\pi(v,0) = (u,f)$, dann gilt:

$$\pi \in \mathcal{M}(u,f)(d \cdot t + 1) \subseteq \mathcal{M}(u,f)(d \cdot t + f) \text{ mit } (u,f) \in \mathcal{I}_w \cup \mathcal{Y}_w \ .$$

Sei $\pi \in \mathcal{M}(u,f)(d \cdot t + f)$ mit $\Delta_\pi(u,f) = (u',f')$ und $(u,f) \in \mathcal{I}_w \cup \mathcal{Y}_w$.

1.Fall: $(u,f) \in \mathcal{I}_w$
Da $\mathcal{M}(u,f)(d \cdot t) = \emptyset$ und $\text{indeg}(u,f) = 1$ gilt: $\{\pi\} = \mathcal{M}(u,f)(d \cdot t + f)$.
Also folgt:

$$\begin{aligned} \pi \in \mathcal{M}(u',f')(d \cdot t + f + 1) &\subseteq \mathcal{M}(u',f')(d \cdot t + f') \ , \\ \mathcal{M}(u,f)(d \cdot t + f + 1) &= \mathcal{M}(u,f)(d \cdot t + d) = \emptyset \ . \end{aligned}$$

2.Fall: $(u,f) \in \mathcal{Y}_w$ und $\pi = \text{first}\mathcal{M}(u,f)(d \cdot t + f)$, dann gilt:

$$\pi \in \mathcal{M}(u',f')(d \cdot t + f + 1) \subseteq \mathcal{M}(u',f')(d \cdot t + f') \ .$$

3.Fall: $(u,f) \in \mathcal{Y}_w$ und $\pi \neq \text{first}\mathcal{M}(u,f)(d \cdot t + f)$, dann gilt:

$$\pi \in \mathcal{M}(u,f)(d \cdot t + f + 1) = \mathcal{M}(u,f)(d \cdot t + d) \ .$$

Daher gilt:

$$M(w)(t+1) \subseteq \mathcal{M}(w,0)(d \cdot t + d) \cup \bigcup_{(u,f) \in \mathcal{Y}_w} \mathcal{M}(u,f)(d \cdot t + d) \ .$$

Zusammen mit $\mathcal{Y}_{w_1} \cap \mathcal{Y}_{w_2} = \emptyset$ für $w_1 \neq w_2$ folgt daraus auch die Gleichheit von linker und rechter Seite, weil die Gesamtheit aller Pakete unverändert bleibt.
Damit ist die Gültigkeit der Induktionsbehauptung in den Punkten **(i)** und **(ii)** gezeigt.
Des weiteren gilt für alle $d \cdot t < t' \leq d \cdot t + d$:

$$M(w)(t+1) \supseteq \mathcal{M}(w,0)(t') \cup \bigcup_{(u,f) \in \mathcal{Y}_w} \mathcal{M}(u,f)(t') \ .$$

Sei $\pi_0 = \text{first}M(w)(t+1)$, dann gilt für alle $d \cdot t < t' < d \cdot t + d$:

$$\pi_0 \in \mathcal{M}(u,f)(t') \text{ für } (u,f) \in \mathcal{Y}_w \Longrightarrow \pi_0 = \text{first}\mathcal{M}(u,f)(t') \ ,$$

da $p_{\pi_0}(u) = p_{\pi_0}(w)$.
Bei der Beförderung des Paketes π_0 auf dem Umweg zu den Zeitpunkten $d \cdot t < t' < d \cdot t + d$ tritt also nie der **3.Fall** ein, so daß spätestens zum Zeitpunkt $d \cdot t + d$ gilt:

$$\pi_0 \in \mathcal{M}(w,0)(d \cdot t + d) \quad \text{und somit} \quad \pi_0 = \text{first}\mathcal{M}(w,0)(d \cdot t + d) \; .$$

Dies zeigt **(iii)**. ∎

Berücksichtigt man noch die Rolle des Knotens $x \notin V$, zu dem alle Pakete geschickt werden, die ihr Ziel erreicht haben, so ergibt sich:

$$M(x)(t) = \mathcal{M}(x,0)(d \cdot t) \; .$$

Dies bedeutet für die Gesamtlaufzeit $\mathcal{T}$ des Routings auf dem defekten Netz:

$$d \cdot T = \mathcal{T} \; .$$

Satz III.1:
Gegeben sei ein intaktes Kommunikationsnetzwerk $G = (V, E)$, ausgestattet mit Prozessoren, die gleichzeitig über alle eingehenden Leitungen Pakete empfangen können, aber pro Zeiteinheit nur *ein* Paket verschicken. Kommunikationsanforderungen seien durch Angabe von Routen- und Prioritätsfunktion für jedes einzelne Paket spezifiziert. Fällt ein Teil der Leitungen $E_{def} \subset E = E_{def} \cup E_{int}$ aus, existiert aber ein *knoten-konfluentes Umwegesystem mit Dilatation d*, so läßt sich jede Kommunikationsanforderung auf dem Restnetzwerk $G = (V, E_{int})$ mit Verzögerungsfaktor d im Vergleich zur Laufzeit auf dem intakten Netz bewältigen. Dabei verschickt jeder Prozessor pro Zeiteinheit auch nur *ein* Paket.

Es wird im allgemeinen eine schwierige Aufgabe sein, ein *knoten-konfluentes Umwegsystem* mit kleiner Dilatation für ein Kommunikationsnetzwerk mit Leitungsdefekten zu bestimmen, selbst wenn es sich um den n-dim Würfel handelt. Versucht man aus den γ-*konjunkten Umwegsystemen*, die im ersten Abschnitt dieses Kapitels behandelt wurden, *knoten-konfluente Umwegsysteme* zu konstruieren, so stellt man fest, daß man keine konstante Dilatation gewährleisten kann. Abb.7 zeigt ein entsprechendes Beispiel, an dem man erkennt, daß die Dilatation linear mit n anwachsen kann, obwohl das Umwegsystem 1-konjunkt ist.

Will man dennoch die γ-konjunkten Umwegsysteme zum probabilistischen Routing auf dem n-dim Würfel verwenden und einen konstanten Verzögerungsfaktor erreichen, so darf man nicht die *Knoten* des Kommunikationsgraphen als die Engpässe bei der Übermittlung der Pakete betrachten. Man sollte den Prozessoren – wie bei Valiant – erlauben, über alle intakten Leitungen gleichzeitig zu kommunizieren. Diese Betrachtungsweise führt dann zu dem Begriff der *kanten-konfluenten Umwegsysteme*.

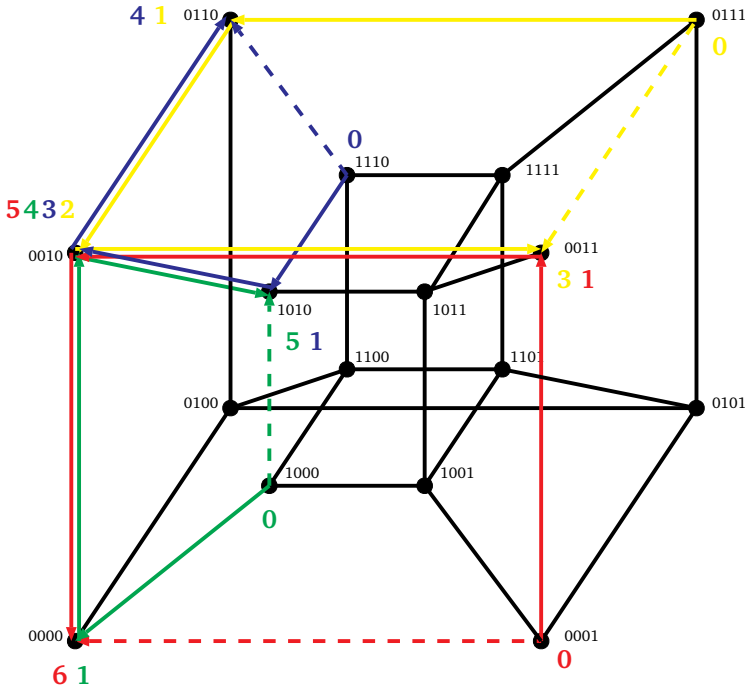

Abbildung 7:

(2) Kanten-konfluentes Umwegesystem mit Dilatation d:
Es sei

$$\begin{aligned} \psi = (\psi_0, \ldots, \psi_{d-1}) \quad &: \quad E_{def} \to (E_{int} \cup \{\diamond\})^d \\ \psi_f \quad &: \quad E_{def} \to E_{int} \cup \{\diamond\} \qquad \forall\, 0 \le f \le d-1 \,. \end{aligned}$$

Für $(v \to w) \in E$ seien $0 \le f_1 < \ldots < f_{l-1} < f_l \le d-1$ die Indizes mit $\psi_{f_i}(v \to w) = (u_{i-1} \to u_i) \neq \diamond$ für $i = 1, \ldots, l$, dann soll gelten:

$$(u_{i-1} \to u_i) \in E_{int} \text{ für } i = 1, \ldots, l \text{ mit } u_0 = v \text{ und } u_l = w \,.$$

Definition:
ψ heißt *kanten-konfluent*, wenn folgende Eigenschaft erfüllt ist:
$\forall\, (v \to w), (v' \to w') \in E_{def}$:

$$\begin{aligned} \exists\, f : \quad & \psi_f(v \to w) = \psi_f(v' \to w') \neq \diamond \\ \Longrightarrow \quad & \psi_i(v \to w) = \psi_i(v' \to w') \quad \forall\, 0 \le i \le f \text{ oder } \forall f \le i \le d-1 \end{aligned}$$

Auch hier müssen die Übergangsfunktion ∇_π und die Prioritätszahlen auf den Umwegen neu spezifiziert werden:

Wenn $\delta_\pi(v) = w$ ist und $0 \leq f_1 < f_2 < \ldots < f_l \leq d-1$ die Indizes sind mit $\psi_{f_i}(v \to w) = (u_{i-1} \to u_i) \neq \diamond$ für $i = 1, \ldots, l$, wobei $u_0 = v$ und $u_l = w$ ist, dann sei:

$$\begin{aligned}
\nabla_\pi(v, \diamond, 0) &= (u_1, u_2, f_2) \\
\nabla_\pi(u_{i-1}, u_i, f_i) &= (u_i, u_{i+1}, f_{i+1}) \quad \text{für } i = 2, \ldots, l-1 \\
\nabla_\pi(u_{l-1}, w, f_l) &= (w, \diamond, 0) \\
\nabla_\pi(\text{ziel}(\pi), \diamond, 0) &= (x, \diamond, 0) \qquad \text{wobei } x \notin V \\
p_\pi(u_i) &= p_\pi(w) \qquad \text{für } i = 1, \ldots, l-1 \; .
\end{aligned}$$

Jeder Prozessor $v \in V$ verwaltet die Pakete in $(d-1) \cdot \text{outdeg}(v) + 1$ Warteschlangen. Zu jeder Kante $(v \to w) \in E_{int}$ existieren $d-1$ Warteschlangen $\mathcal{M}(v, w, 1), \ldots, \mathcal{M}(v, w, d-1)$. Außerdem gibt es noch die Warteschlange $\mathcal{M}(v, \diamond, 0)$.

Arbeitsweise:

$t \equiv f \bmod d$:

$\forall\, u \in V$ mit $(u \to u') \in E_{int}$ oder $u' = \diamond$:

$$\mathcal{M}(u, u', f)(t+1) = \mathcal{M}(u, u', f)(t) \setminus \{\text{first}\mathcal{M}(u, u', f)(t)\}$$

$\forall\, u' \in V$ mit $(u' \to u'') \in E_{int}$ oder $u'' = \diamond$:

$$\begin{aligned}
\mathcal{M}(u', u'', f')(t+1) \;=\; & \mathcal{M}(u', u'', f')(t) \;\cup \{\pi | \exists u, f' : \nabla_\pi(u, u', f) \\
& = (u', u'', f') \,\text{und}\, \pi = \text{first}\mathcal{M}(u, u', f)\}
\end{aligned}$$

$t \equiv 0 \bmod d$:

$$\begin{aligned}
\mathcal{M}(x, \diamond, 0)(t+1) \;=\; & \mathcal{M}(x, \diamond, 0)(t) \;\cup \\
& \{\pi | \exists u : \nabla_\pi(u, \diamond, 0) = (x, \diamond, 0) \,\text{und}\, \pi = \text{first}\mathcal{M}(u, \diamond, 0)\}
\end{aligned}$$

Analog zur *knoten-konfluenten* Version läßt sich folgender Satz zeigen:

Satz III.2:
Gegeben sei ein intaktes Kommunikationsnetzwerk $G = (V, E)$, ausgestattet mit Prozessoren, die gleichzeitig über alle eingehenden Leitungen Pakete empfangen können, aber pro Zeiteinheit nur *ein* Paket verschicken.

Kommunikationsanforderungen seien durch Angabe von Routen- und Prioritätsfunktion für jedes einzelne Paket spezifiziert. Fällt ein Teil der Leitungen $E_{def} \subset E = E_{def} \cup E_{int}$ aus, existiert aber ein *kanten-konfluentes Umwegesystem mit Dilatation d*, so läßt sich jede Kommunikationsanforderung auf dem Restnetzwerk $G = (V, E_{int})$ mit Verzögerungsfaktor d im Vergleich zur Laufzeit auf dem intakten Netz bewältigen, wenn man erlaubt, daß jeder Prozessor im defekten Netz pro Zeiteinheit über alle ausgehenden intakten Kanten gleichzeitig Pakete verschicken darf.

Anwendung von Satz III.2:
Gegeben sei der Kommunikationsgraph $G = (V, E)$ des n-dim Würfels. In Kapitel I wurde gezeigt, daß sich auf dem intakten Netz jede Permutationsanforderung mit großer Wahrscheinlichkeit in Zeit $c \cdot n$ routen läßt, wobei die Prozessoren pro Zeiteinheit nur ein Paket abschicken.
Wenn ein nicht zu großer Teil der Verbindungsleitungen $E_{def} \subset E$ ausgefallen ist, so sollte es möglich sein, ein kanten-konfluentes Umwegesystem mit kleiner Dilatation d zu bestimmen. Erlaubt man den Prozessoren dann, über alle intakten Ausgangsleitungen gleichzeitig zu senden, so kann man nach Satz III.2 erreichen, daß sich jede Permutationsanforderung nun mit hoher Wahrscheinlichkeit noch in Zeit $d \cdot c \cdot n$ routen läßt, wenn man von dem zusätzlichen prozessorinternen Verwaltungsaufwand absieht.
Es muß also das Ziel sein, zu einer vorgegebenen Verteilung von Kantenfehlern auf dem n-dim Würfel ein kanten-konfluentes Umwegesystem mit kleinst möglicher Dilatation zu bestimmen.

Betrachtet man die $\gamma-$konjunkten Umwegessysteme aus dem ersten Teil dieses Kapitels etwas näher, so stellt man fest, daß diese die Bedingung der Kantenkonfluenz erfüllen, wobei der Dilatationsfaktor $\gamma + 2$ beträgt.
Bei einem $\gamma-$konjunkten Umwegesystem gibt es zu jeder intakten Kante $u \to u' \in E_{int}$ höchstens γ Umwege, die diese Kante als *zweite* Umwegkante benutzen.
Wenn für $i = 1, \ldots, l \leq \gamma$ der zur defekten Kante $v_i \to w_i \in E_{def}$ festgelegte Umweg $U_{v_i,w_i} : v_i \to u \to u' \to w_i$ ist, dann definiere:

$$\psi_j(v_i \to w_i) = \begin{cases} (v_i \to u) & \text{für} \quad j = 0 \\ (u \to u') & \text{für} \quad j = i \\ (u' \to w_i) & \text{für} \quad j = \gamma + 1 \\ \diamond & \text{sonst} \end{cases}$$

Beh: ψ beschreibt ein kanten-konfluentes Umwegesystem mit Dilatation $\gamma + 2$.

Bew: Sei $\psi_j(v \to w) = \psi_j(v' \to w') = (u \to u') \neq \diamond$.

1.Fall: $1 \leq j \leq \gamma$
$(u \to u')$ ist die jeweils *zweite* Kante auf den Umwegen $U_{v,w}$ und $U_{v',w'}$. Dann muß nach obiger Definition von ψ aber gelten: $(v \to w) = (v' \to w')$

2.Fall: $j = 0$ oder $j = \gamma + 1$
$(u \to u')$ ist die jeweils *erste* bzw. *letzte* Kante auf den Umwegen $U_{v,w}$ und $U_{v',w'}$. Also gilt die Beh.

■

Literaturverzeichnis

[**Bat**] Batcher:
Sorting networks and their applications., AFIPS Spring Joint Comp. Conf. 32 (1968) pp. 307-314

[**BDFS**] A. Broder, D. Dolev, M. Fischer and B. Simons:
Efficient Fault Tolerant Routings in Networks, STOC 16 (1984), pp. 536-541

[**BS**] B. Becker and H.U. Simon:
How Robust is the n-Cube?, FOCS (1986), pp. 283-291

[**Che**] H. Chernoff:
A measure of asymptotic efficiency for tests of hypothesis based on the sum of observations. Ann. Math. Stat. 23 (1952), pp. 493-507

[**DHSS**] D. Dolev, J. Halpern, B. Simons and R. Strong:
A new look at fault tolerant network routing., STOC 16 (1984), pp. 526-535

[**Hoe**] W. Hoeffding:
On the distribution of the number of successes in independent trials. Ann. Math. Stat. 27 (1956), pp. 713-721

[**Knu**] D.E. Knuth:
Sorting and Searching. Addison Wesley (1973)

[**Meh1**] K. Mehlhorn:
Parallele Algorithmen. Vorlesung an der Universität des Saarlandes im WS 85/86

[**Meh2**] K. Mehlhorn:
Graph Algorithms and NP-Completeness. Springer-Verlag (1984)

[**PU**] D. Peleg and E. Upfal:
The token Distribution problem., FOCS (1986), pp. 418-427

[**Sta**] P. Stadtmüller:
Untersuchungen zum Laufzeitverhalten bei balancierten Kommunikationsschemata. Diplomarbeit an der Universität des Saarlandes, (Jan. 1987)

[**Upf**] E. Upfal:
Efficient Schemes for Parallel Communication. J. ACM 31, (1984), pp. 507-517

[**Val**] L.G. Valiant:
A scheme for fast parallel computation. SIAM J. Comput. 11, (1982), pp. 350-361

[**VB**] L.G. Valiant and G.J. Brebner:
Universal schemes for parallel communication. STOC 13 (1981) pp. 263-277