

Trellis Diagramme und Viterbi-Decoder

Michael Dienert

20. März 2006

1 Fehlertolerante Datenübertragung bei Gigabit-Ethernet

Um 1000 MBit/s auf Kat 5 Kupferkabeln übertragen zu können, sind eine Reihe technischer Kunstgriffe notwendig: neben der 4D-PAM5 Übertragung und der Verwendung aller 4 Leitungen eines Kabels, wird eine spezielle Codierung eingesetzt, die *automatisch* Bitfehler *erkennen* und *korrigieren* kann.

Damit die Spezifikation von Kat 5 Kabeln eingehalten wird, darf die Signalfrequenz auf den Leitungen dabei 125 MHz nicht überschreiten. 125 MHz entspricht einer Periodendauer von 8 ns.

Durch die 4D-5PAM Übertragung hat man $5^4 = 625$ verschiedene Symbole zur Verfügung. Damit kann man in *einem Schritt* 9 Bit übertragen, da man für 9 Bit genau 512 verschiedene Symbole benötigt. Damit man auf die 125 MHz Grenzfrequenz kommt, dauert so ein Übertragungsschritt genau 8 ns. Damit erreicht man folgende Bitrate:

$$\begin{aligned} \text{Bitrate} &= \frac{9\text{Bit}}{8\text{ns}} = \frac{9}{8} \cdot 10^9 \text{ Bit/s} \\ \text{Bitrate} &= 1.125\text{GBit/s} \end{aligned}$$

Da die Nutzdatenrate aber nur 8 GBit/s beträgt, kann man *zusätzliche* Informationen übertragen.

Diese zusätzlichen Bits im Datenstrom nennt man *Redundanz*. Durch die redundante Information wird es möglich, Fehler bei der Übertragung zu erkennen und zu korrigieren.

1.1 Ein einfaches Beispiel: Parity-Checking

Ein sehr einfaches Verfahren der Fehlererkennung ist das bekannte Parity-Bit: ist die Quersumme der Nutzdatenbits gerade, setzt man das Parity-Bit, ist sie ungerade, ist das Parity-Bit = 0. Der Empfänger kann nun erkennen, wenn sich bei der Übertragung 1 Bit geändert hat. Welches Bit das ist, kann er allerdings nicht mehr ermitteln, eine Korrektur des Fehlers ist also unmöglich.

1.2 Prinzip der Fehlerkorrektur bei GigabitEthernet

Bei GigabitEthernet kommt ein gegenüber dem Parity-Check ungleich komplizierteres Verfahren zur Anwendung.

Wie dieses Verfahren, das Fehler erkennt und korrigieren kann, funktioniert lässt sich mit einem Beispiel aus der Sprache erklären. Hierzu ein Zitat aus Goethes Faust:

Mein schönes Zräulein dart ich wagen, meinen Grm und Geleit Ihr anzutragen.

Obwohl der Satz drei Buchstabenfehler enthält ist es nicht schwer, diese zu finden und zu korrigieren. Möglich wird dies, da die Sprache festgelegten Regeln folgt, die Sender und Empfänger kennen: Buchstaben dürfen nicht in beliebiger Reihenfolge angeordnet sein sondern es sind nur bestimmte Sequenzen, eben die uns bekannten Wörter erlaubt. Tauchen unbekannte Sequenzen auf sind Fehler enthalten, die durch Vergleich mit bekannten Sequenzen korrigiert werden können.

Auf die gleiche Weise funktioniert die Viterbi-Decodierung: der Datenstrom wird in bestimmte Sequenzen, die auch dem Empfänger bekannt sind, umcodiert.

1.3 Codierung mit Trellis-Diagramm

Abb. 1 zeigt ein sehr 'einfaches' Trellis-Diagramm (engl. Trellis = Spalier) mit dem der Übertragungscode erzeugt wird.

Das Diagramm sieht auf den ersten Blick nicht einfach aus, es ist aber einfach, wenn man es mit dem Trellis-Diagramm von GigabitEthernet vergleicht: dort führen von jedem Knoten 4 Pfade weg. Dabei sind 8 Knoten übereinander angeordnet!

Zurück zum einfachen Trellis-Diagramm, das hier als Beispiel dient: Ist das Nutzdatenbit '0', folgt man der durchgezogenen Linie, bei einer '1' der gestrichelten Linie. Gesendet werden dabei die Bitpaare, die entlang des Pfades stehen.

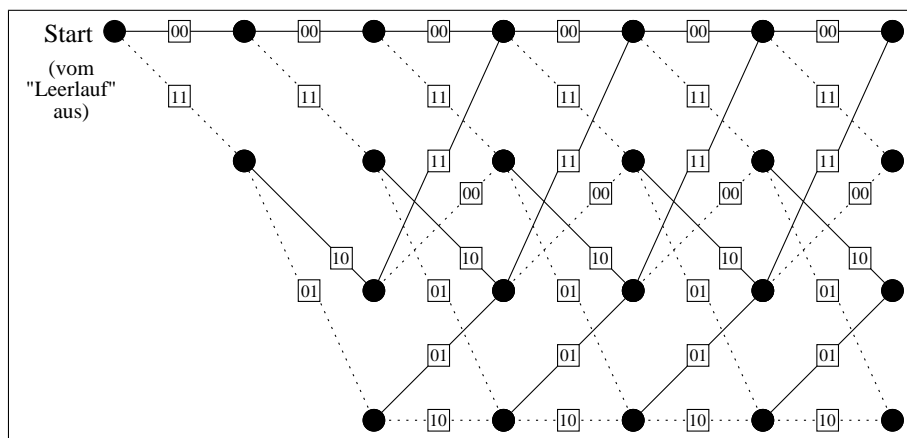


Abbildung 1: Trellis-Diagramm

1.4 Decodierung mit Viterbi-Decoder

Um eine empfangene Bitfolge wieder zu dekodieren, geht man nach folgenden Regeln vor:

- Ein empfangenes Bitpaar legt den Pfad fest. Ist der Pfad gestrichelt dargestellt, stellt das Bitpaar eine '1' dar, bei einem durchgezogenen Pfad ist es eine '0'.
- Ist ein Bitfehler aufgetreten kann es sein, dass von einem Knoten aus kein passender Pfad ausgeht. Dann muss man *beide* Pfade verfolgen. Die Pfade werden

dabei *bewertet*: Man berechnet den *Bitabstand* zwischen Pfad und empfangenen Bitpaar. Beispiel: Wir sind am Startpunkt, haben aber '10' empfangen. Vom Startpunkt aus gibt es aber nur die Alternativen '00' und '11'. Beide unterscheiden sich aber jeweils an einer Stelle vom empfangenen Bitpaar. Der Bitabstand ist in beiden Fällen also $d = 1$. Ist das empfangene Bitpaar z.B. '01' der Pfad aber '10' wäre der Bitabstand $d = 2$.

- Sind Bitfehler in der Empfangsdatenfolge, führt das im Trellis-Diagramm zu vielen Verzweigungen, d.h. man hat mehrere Pfade zu verfolgen. Man summiert nun alle Bitabstände entlang eines Pfades und wählt nach einer gewissen Zeit den Pfad mit der kleinsten Summe als gültigen Pfad aus.

1.5 Das Trellis-Diagramm bei Gigabit-Ethernet auf UTP-Kabel

Die Abb. 4 zeigt nun das Trellis-Diagramm, wie es bei Gigabit-Ethernet eingesetzt wird. Man erkennt, dass es gegenüber unserem obigen Beispiel *deutlich komplizierter* geworden ist.

Hier der Versuch, eine kleine Erläuterung zu liefern:

- Wir erinnern uns an die Codierung von GE: 4D-5PAM. Mit dieser Codierung lassen sich 625 Symbole übertragen, von denen wir jetzt 512 (entsprechend 9 bit) benötigen.
- Diese 512 Symbole werden in 8 Gruppen zu je 64 Symbolen geteilt. Dabei werden die Symbolgruppen so gebildet, dass innerhalb einer Gruppe keine direkt benachbarten Symbole vorkommen, sondern man hat sehr darauf geachtet, dass die Symbole innerhalb einer Gruppe einen möglichst grossen Abstand (im 4D-Körper) voneinander haben.
- Unsere 8 Gruppen werden nochmals in 2 Familien unterteilt: in der einen Familie sind die sog. *geraden* Symbole, in der anderen die *ungeraden* Symbole.
- Nun wollen wir ein 9-Bit-Wort übertragen. Die neun Bits D8 - D0 werden dabei folgendermassen gesendet:
 - b5, b4, b3, b2, b1, b0** Diese 6 Bit wählen das Symbol innerhalb einer Gruppe aus. Wir erinnern uns: in jeder der 8 Gruppen sind 64 Symbole, die sich also mit 6 Bit bestimmen lassen.
 - b7, b6** Mit diesen beiden Bits wird ausgewählt, auf welchem Pfad in den nächsten Zustand gewechselt wird. Das heisst also, dass von jedem der Zustände des Trellis-Diagramms 4 Pfade weggehen.
 - b8** Dieses Bit wählt nun aus, ob wir Symbole aus der **geraden** oder der **ungeraden** Familie senden dürfen. Betrachtet man Bild 4 fällt auf, dass dieses Bit b8 vom Zustand der Bits b7 und b6 *abhängt*: Je nach dem wo man sich im Trellis-Diagramm befindet, muss man innerhalb einer Familie bleiben (muss also z.B. von **gerade nach gerade** wechseln) oder muss die Familie wechseln (z.B. von **gerade nach ungerade**). Im Bild 4 sind die Familien durch Farbe, Form und Schraffur der Zustände gekennzeichnet. Technisch realisiert ist das durch ein digitales Schaltwerk, dessen Folgezustand vom aktuellen Zustand und von den Bits b7 und b6 abhängt.

2 Aufgaben

1. Gesendet werden soll die Nutzdatenfolge: **01011**. Ermittle mit Hilfe eines Trellis-Diagramms die codierte Sende-Bitfolge. Verwende dazu die nachfolgenden Diagramme.
2. Die empfangene Bit(paar)folge sei: '10 11 00 00 01'. Dekodiere diese Folge und korrigiere eventuelle Bitfehler.

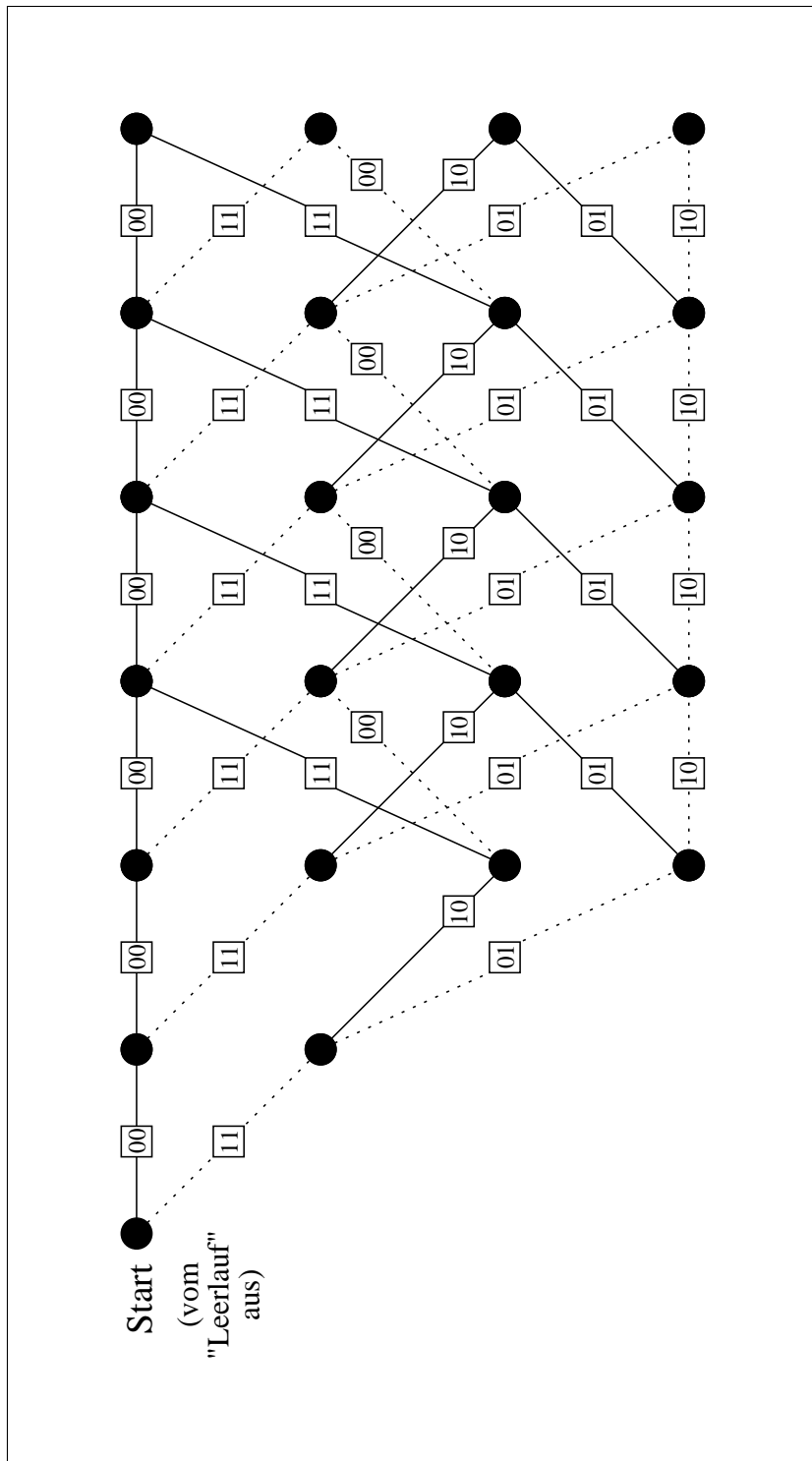


Abbildung 2: Trellis-Diagramm: durchgezogene Linie = '0', gestrichelte Linie = '1'

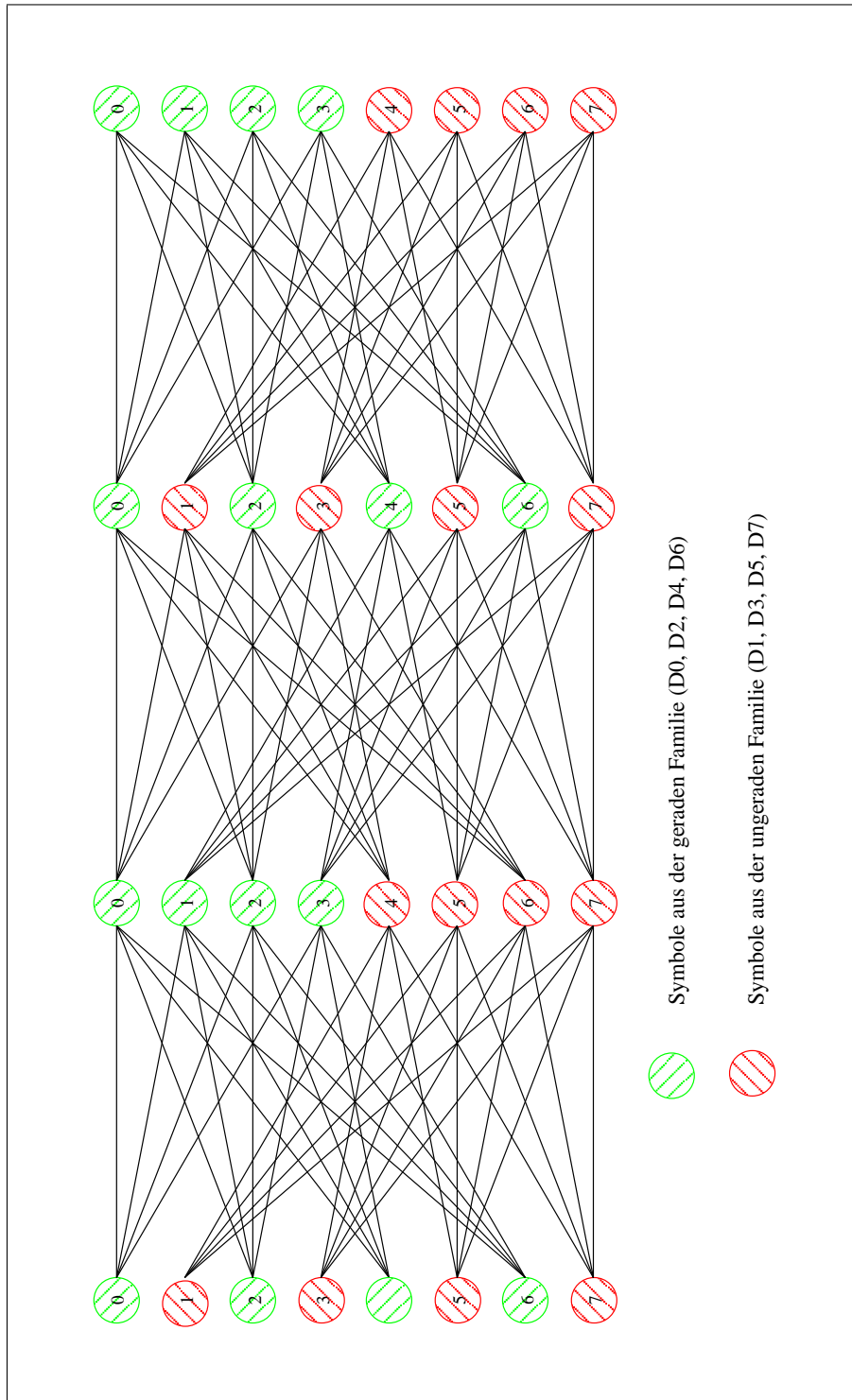


Abbildung 4: Das Trellis-Diagramm von GigabitEthernet