net2o: Fort(h)schritte

Bernd Paysan

Forth–Tagung 2011, Goslar

Outline

Hirntraining, um wieder richtig in Form zu kommen

- Ohne Aufgabe kein Spaß im Leben
- Der Frust im Job hat sich so zugespitzt, dass ich gegangen bin — Sabbatical!
- Wieder Status–Report und Todo–Liste

Hirntraining, um wieder richtig in Form zu kommen

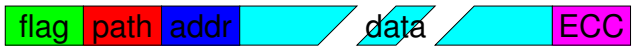
- Ohne Aufgabe kein Spaß im Leben
- Der Frust im Job hat sich so zugespitzt, dass ich gegangen bin — Sabbatical!
- Wieder Status–Report und Todo–Liste

Hirntraining, um wieder richtig in Form zu kommen

- Ohne Aufgabe kein Spaß im Leben
- Der Frust im Job hat sich so zugespitzt, dass ich gegangen bin — Sabbatical!
- Wieder Status–Report und Todo–Liste

Geändert: Packet Header

	Size
Flags	*2*
Path	*2/8*
Address	*2/8*
Junk	*0/8*
Data	*32/64/128/256/512/1k/2k/4k*
checksum	*0/8*



Ein Anfang ist gemacht

- net2o über UDP (IPv4/IPv6) als „physical layer“
- UDP stellt ein einfaches Interface zur Verfügung, das auch viele Verbindungen gleichzeitig beherrscht.
- IPv4 komplett, IPv6 bisher nur ohne Packet fragmentation
- Zwei Teile: Packet–Handler und Kommando–Interpreter

Ein Anfang ist gemacht

- net2o über UDP (IPv4/IPv6) als „physical layer“
- UDP stellt ein einfaches Interface zur Verfügung, das auch viele Verbindungen gleichzeitig beherrscht.
- IPv4 komplett, IPv6 bisher nur ohne Packet fragmentation
- Zwei Teile: Packet–Handler und Kommando–Interpreter

Ein Anfang ist gemacht

- net2o über UDP (IPv4/IPv6) als „physical layer“
- UDP stellt ein einfaches Interface zur Verfügung, das auch viele Verbindungen gleichzeitig beherrscht.
- IPv4 komplett, IPv6 bisher nur ohne Packet fragmentation
- Zwei Teile: Packet–Handler und Kommando–Interpreter

Ein Anfang ist gemacht

- net2o über UDP (IPv4/IPv6) als „physical layer“
- UDP stellt ein einfaches Interface zur Verfügung, das auch viele Verbindungen gleichzeitig beherrscht.
- IPv4 komplett, IPv6 bisher nur ohne Packet fragmentation
- Zwei Teile: Packet–Handler und Kommando–Interpreter

Switching

- Ein Hash wird benutzt, um IP–Addressen zu switchen: Hash=Prefix
- Noch zu tun: Kolissionen mit längerem Prefix vermeiden
- Prefixgranularität: Byte

  MSB=0 Direkt routebar
  MSB=1 Längerer Prefix, nächstes Byte angucken

Switching

- Ein Hash wird benutzt, um IP–Addressen zu switchen: Hash=Prefix
- Noch zu tun: Kolissionen mit längerem Prefix vermeiden
- Prefixgranularität: Byte
  - MSB=0 Direkt routebar
  - MSB=1 Längerer Prefix, nächstes Byte angucken

Switching

- Ein Hash wird benutzt, um IP–Addressen zu switchen: Hash=Prefix
- Noch zu tun: Kolissionen mit längerem Prefix vermeiden
- Prefixgranularität: Byte

  MSB=0 Direkt routebar

  MSB=1 Längerer Prefix, nächstes Byte angucken

Switching

- Ein Hash wird benutzt, um IP–Addressen zu switchen: Hash=Prefix
- Noch zu tun: Kolissionen mit längerem Prefix vermeiden
- Prefixgranularität: Byte

  MSB=0 Direkt routebar
  MSB=1 Längerer Prefix, nächstes Byte angucken

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Shared Memory

- Map von Adresse zum Kontext funktioniert
- Context enthält
  - Returnaddresse
  - Commando–Buffer
  - cryptographic keys
  - Data&Code Maps
  - Achnowledge–Informationen
  - und noch mehr (noch nicht fertig)

Kommandos

- UTF–8–encodedete Kommandos: Einfaches ist ASCII, ein Byte, komplexeres kann länger sein
- Kommandos in 8–Byte–Blöcke gepackt
- Literals (8 Bytes) und Strings in Kommandostrom eingebaut
- Kommando–Assembler erlaubt direktes Einbinden in normalen Forth–Code

Kommandos

- UTF–8–encodedete Kommandos: Einfaches ist ASCII, ein Byte, komplexeres kann länger sein
- Kommandos in 8–Byte–Blöcke gepackt
- Literals (8 Bytes) und Strings in Kommandostrom eingebaut
- Kommando–Assembler erlaubt direktes Einbinden in normalen Forth–Code

Kommandos

- UTF–8–encodedete Kommandos: Einfaches ist ASCII, ein Byte, komplexeres kann länger sein
- Kommandos in 8–Byte–Blöcke gepackt
- Literals (8 Bytes) und Strings in Kommandostrom eingebaut
- Kommando–Assembler erlaubt direktes Einbinden in normalen Forth–Code

Kommandos

- UTF–8–encodedete Kommandos: Einfaches ist ASCII, ein Byte, komplexeres kann länger sein
- Kommandos in 8–Byte–Blöcke gepackt
- Literals (8 Bytes) und Strings in Kommandostrom eingebaut
- Kommando–Assembler erlaubt direktes Einbinden in normalen Forth–Code

Testcase

Server loop

```
init-server
server-loop
```

Debugging output

```
init-client
s" localhost" net2o-udp insert-ipv4
              constant lserver
net2o-code s" This is a test" $, type
              '!' char, emit cr end-code
lserver 0 send-cmd
```

Testcase

Server loop

```
init-server
server-loop
```

Debugging output

```
init-client
s" localhost" net2o-udp insert-ipv4
          constant lserver
net2o-code s" This is a test" $, type
          '!' char, emit cr end-code
lserver 0 send-cmd
```

Testcase 2

Datei–Transfer

```
net2o-code new-context
$80000 lit, $80000 lit, new-map
$10000 lit, $1000 lit, new-code-map
$80000 lit, $80000 lit, new-data
$10000 lit, $1000 lit, new-code
s" net2o.fs" $, r/o lit, 0 lit, open-file
s" file size: " $, type 0 lit, file-size . cr
0 lit, slurp-chunk send-chunks
0 lit, close-file
s" doc/internet-2.0.pdf" $, r/o lit, 0 lit, open-file
s" file size: " $, type 0 lit, file-size . cr
0 lit, slurp-chunk send-chunks
0 lit, close-file
end-code
```

Testcase 3

Client Loop

```
$80000 $80000 n2o:new-map
$10000 $1000 n2o:new-code-map
$80000 $80000 n2o:new-data
$10000 $1000 n2o:new-code
$80000 $80000 net2o:unacked
client-loop
```

Flusskontrolle

- UDP ist per Definition unzuverlässig
- Zuverlässigkeit durch Resend–Kommandos — größtenteils implementiert.
- Flusskontrollen–Idee: PLL–basiert, Auswertung der Δts ist eingebaut, Senderlimitierung fehlt noch.

Flusskontrolle

- UDP ist per Definition unzuverlässig
- Zuverlässigkeit durch Resend–Kommandos — größtenteils implementiert.
- Flusskontrollen–Idee: PLL–basiert, Auswertung der Δts ist eingebaut, Senderlimitierung fehlt noch.

Flusskontrolle

- UDP ist per Definition unzuverlässig
- Zuverlässigkeit durch Resend–Kommandos — größtenteils implementiert.
- Flusskontrollen–Idee: PLL–basiert, Auswertung der Δts ist eingebaut, Senderlimitierung fehlt noch.

Cryptography

- Ellyptic Curve Cryptography: Dan Bernsteins NaCl–Library wird verwendet
- NaCl hat zwar einen brauchbaren Stream Cipher, aber noch keinen Hash — doch Wurstkessel?
- Verschlüsselung überall ist wichtig — die Alternative ist das chinesische Internet

Cryptography

- Ellyptic Curve Cryptography: Dan Bernsteins NaCl–Library wird verwendet
- NaCl hat zwar einen brauchbaren Stream Cipher, aber noch keinen Hash — doch Wurstkessel?
- Verschlüsselung überall ist wichtig — die Alternative ist das chinesische Internet

Cryptography

- Ellyptic Curve Cryptography: Dan Bernsteins NaCl–Library wird verwendet
- NaCl hat zwar einen brauchbaren Stream Cipher, aber noch keinen Hash — doch Wurstkessel?
- Verschlüsselung überall ist wichtig — die Alternative ist das chinesische Internet

Presentation/Browser

- Typesetting engine
- Bilder, Audio, Video einbetten — aber keine Plugins!
- Scripting vermeiden, wo's geht, z.B. sollte Aggregation aus Seitenteilen schon eine Grundfunktion sein
- Events beim Server registrieren statt ständiges Pollen

Presentation/Browser

- Typesetting engine
- Bilder, Audio, Video einbetten — aber keine Plugins!
- Scripting vermeiden, wo's geht, z.B. sollte Aggregation aus Seitenteilen schon eine Grundfunktion sein
- Events beim Server registrieren statt ständiges Pollen

Presentation/Browser

- Typesetting engine
- Bilder, Audio, Video einbetten — aber keine Plugins!
- Scripting vermeiden, wo's geht, z.B. sollte Aggregation aus Seitenteilen schon eine Grundfunktion sein
- Events beim Server registrieren statt ständiges Pollen

Presentation/Browser

- Typesetting engine
- Bilder, Audio, Video einbetten — aber keine Plugins!
- Scripting vermeiden, wo's geht, z.B. sollte Aggregation aus Seitenteilen schon eine Grundfunktion sein
- Events beim Server registrieren statt ständiges Pollen

Summary

- Es gibt schon fast 1000 Zeilen Code
- Immer noch viel zu tun, aber dafür jetzt auch mehr Zeit
- Wenn's fertig ist: RFC, IETF–Diskussionen, Präsentation auf Netzwerk–orientierten Konferenzen

Summary

- Es gibt schon fast 1000 Zeilen Code
- Immer noch viel zu tun, aber dafür jetzt auch mehr Zeit
- Wenn's fertig ist: RFC, IETF–Diskussionen, Präsentation auf Netzwerk–orientierten Konferenzen

Summary

- Es gibt schon fast 1000 Zeilen Code
- Immer noch viel zu tun, aber dafür jetzt auch mehr Zeit
- Wenn's fertig ist: RFC, IETF–Diskussionen, Präsentation auf Netzwerk–orientierten Konferenzen

For Further Reading I

Bernd Paysan
Internet 2.0
http://www.jwdt.com/~paysan/internet-2.0.html