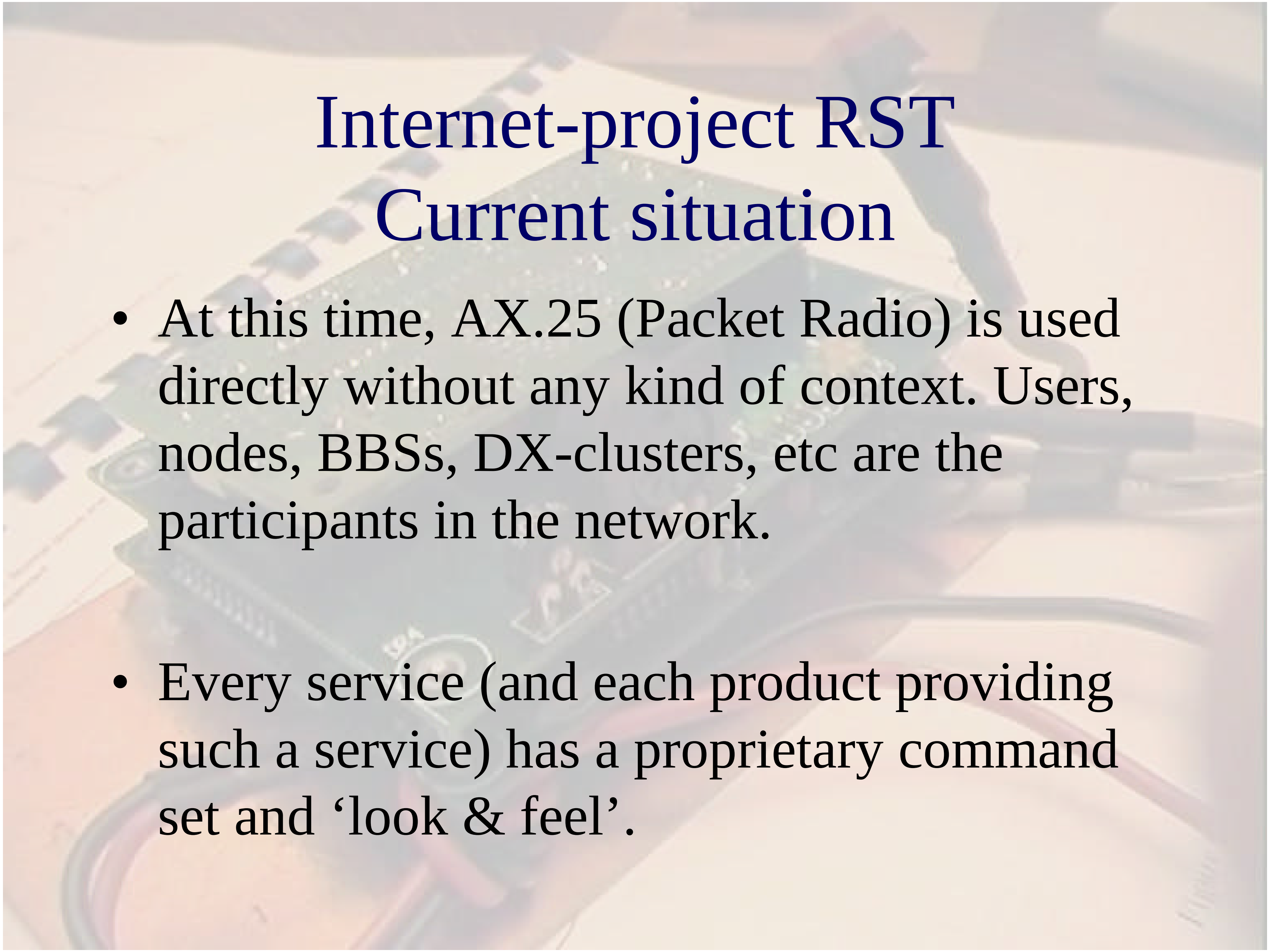
TCP/IP ("Internet") project Sint-Truiden Amateur Radio TCP/IP TaskForce

Main goal: starting a wireless network, which is part of the Internet

Gert Leunen (ON1BLU)

Internet-project RST Program

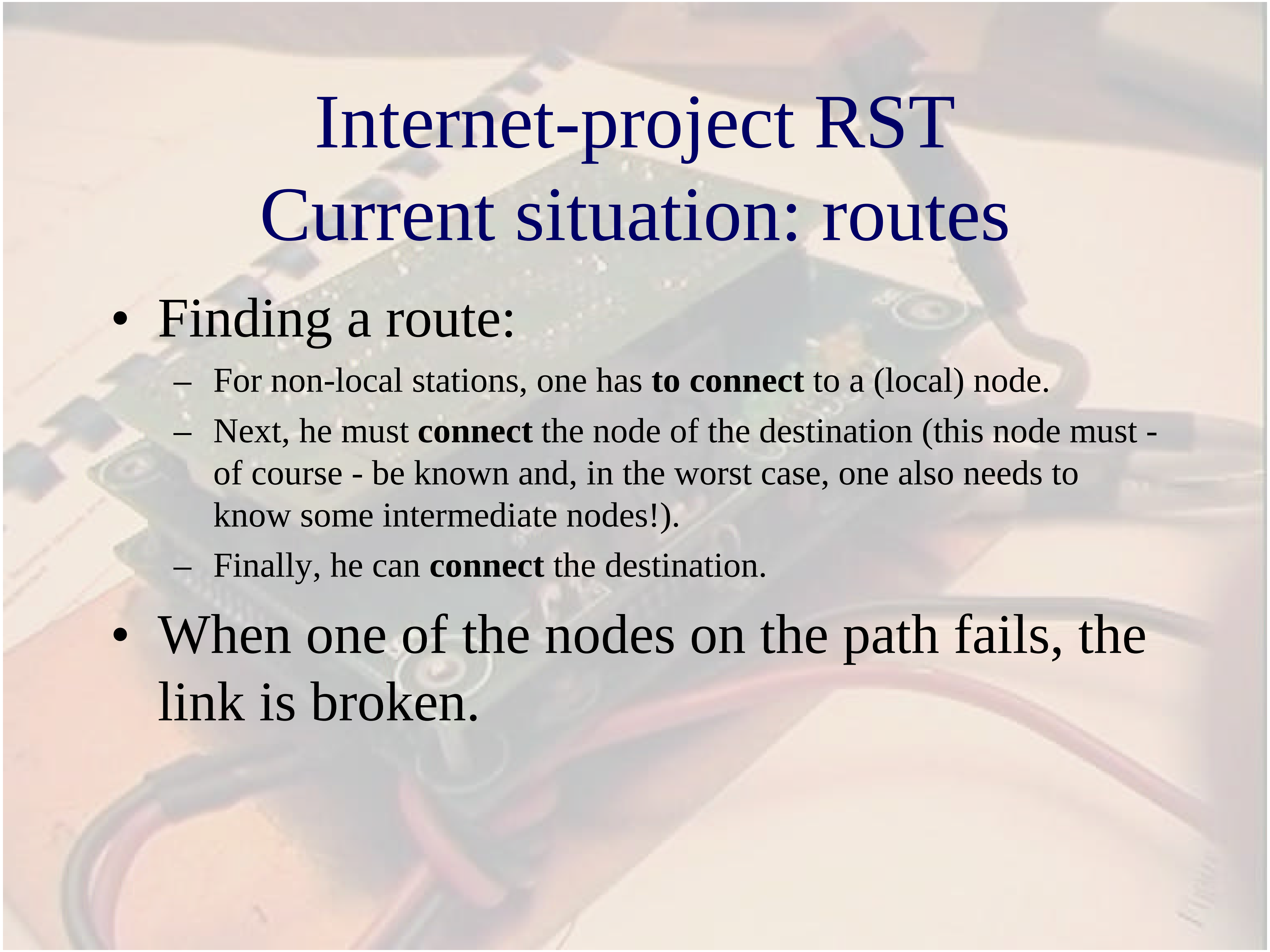
- **It’s hard to exchange (c.f. Packet-Radio)**
- An alternative: introduction to TCP/IP
- Realizing it: some development required!



Internet-project RST
Current situation

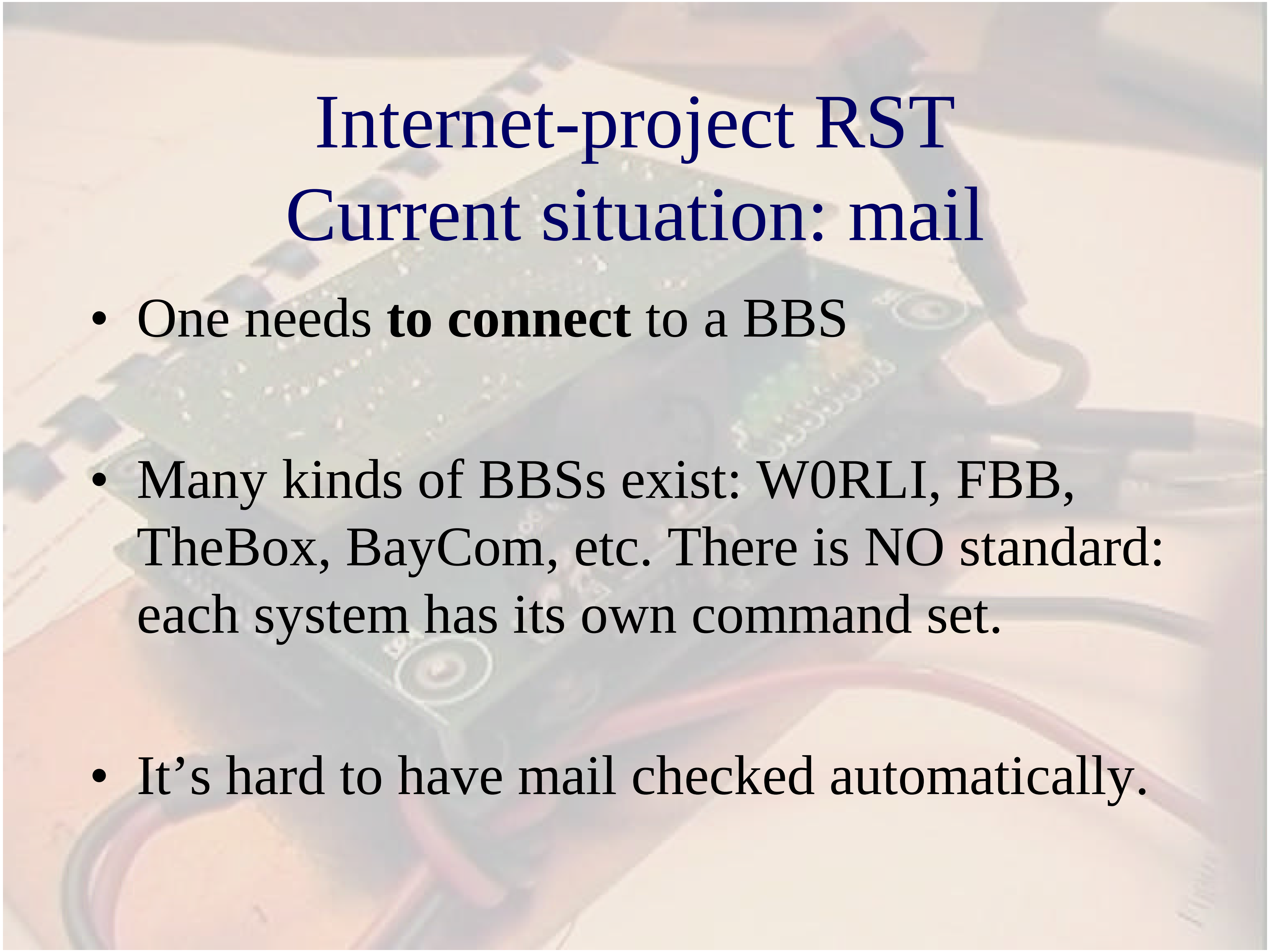
- At this time, AX.25 (Packet Radio) is used directly without any kind of context. Users, nodes, BBSs, DX-clusters, etc are the participants in the network.

- Every service (and each product providing such a service) has a proprietary command set and ‘look & feel’.



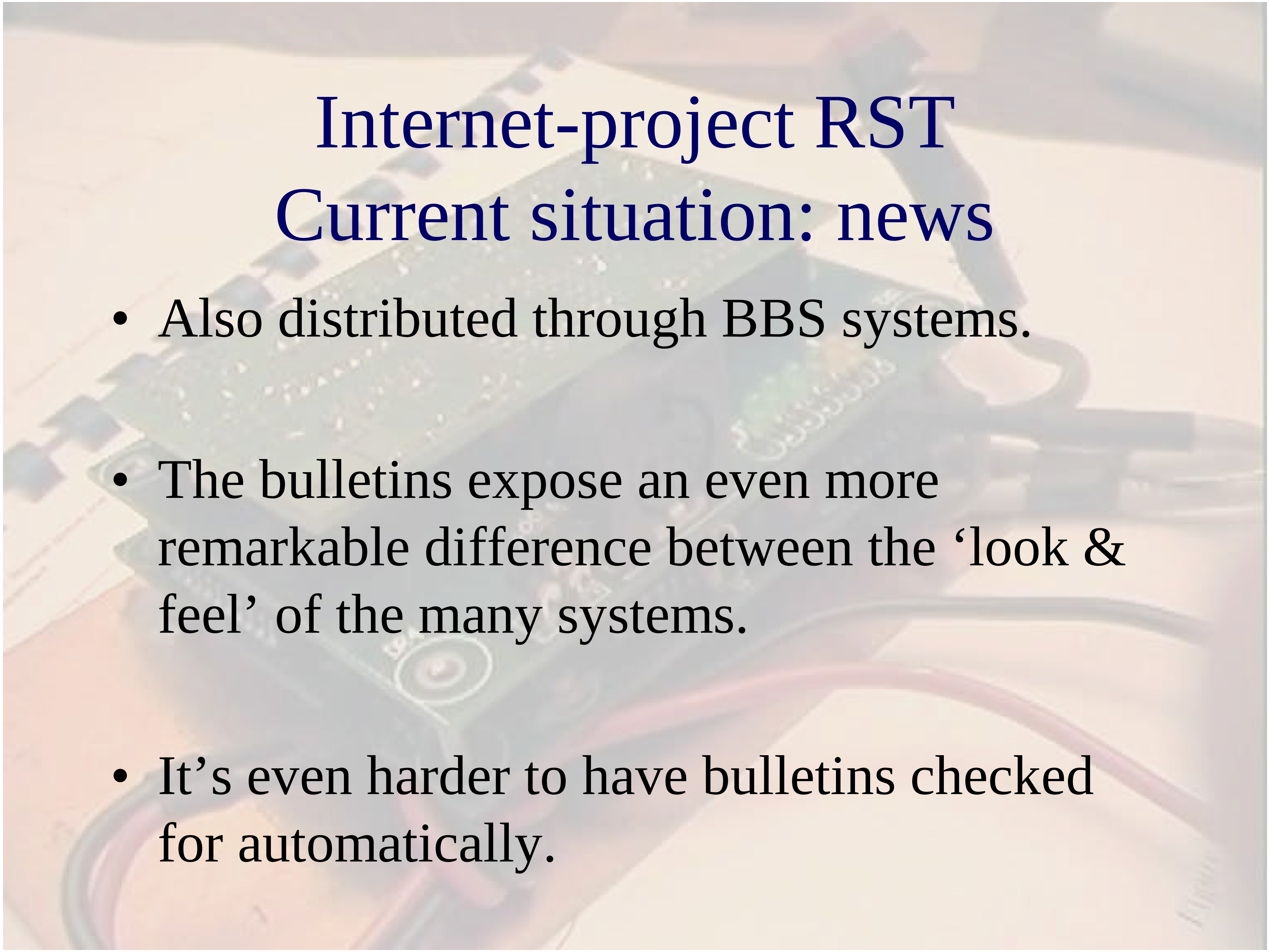
Internet-project RST
Current situation: routes

- Finding a route:
  - For non-local stations, one has **to connect** to a (local) node.
  - Next, he must **connect** the node of the destination (this node must - of course - be known and, in the worst case, one also needs to know some intermediate nodes!).
  - Finally, he can **connect** the destination.
- When one of the nodes on the path fails, the link is broken.



Internet-project RST
Current situation: mail

- One needs **to connect** to a BBS
- Many kinds of BBSs exist: W0RLI, FBB, TheBox, BayCom, etc. There is NO standard: each system has its own command set.
- It’s hard to have mail checked automatically.

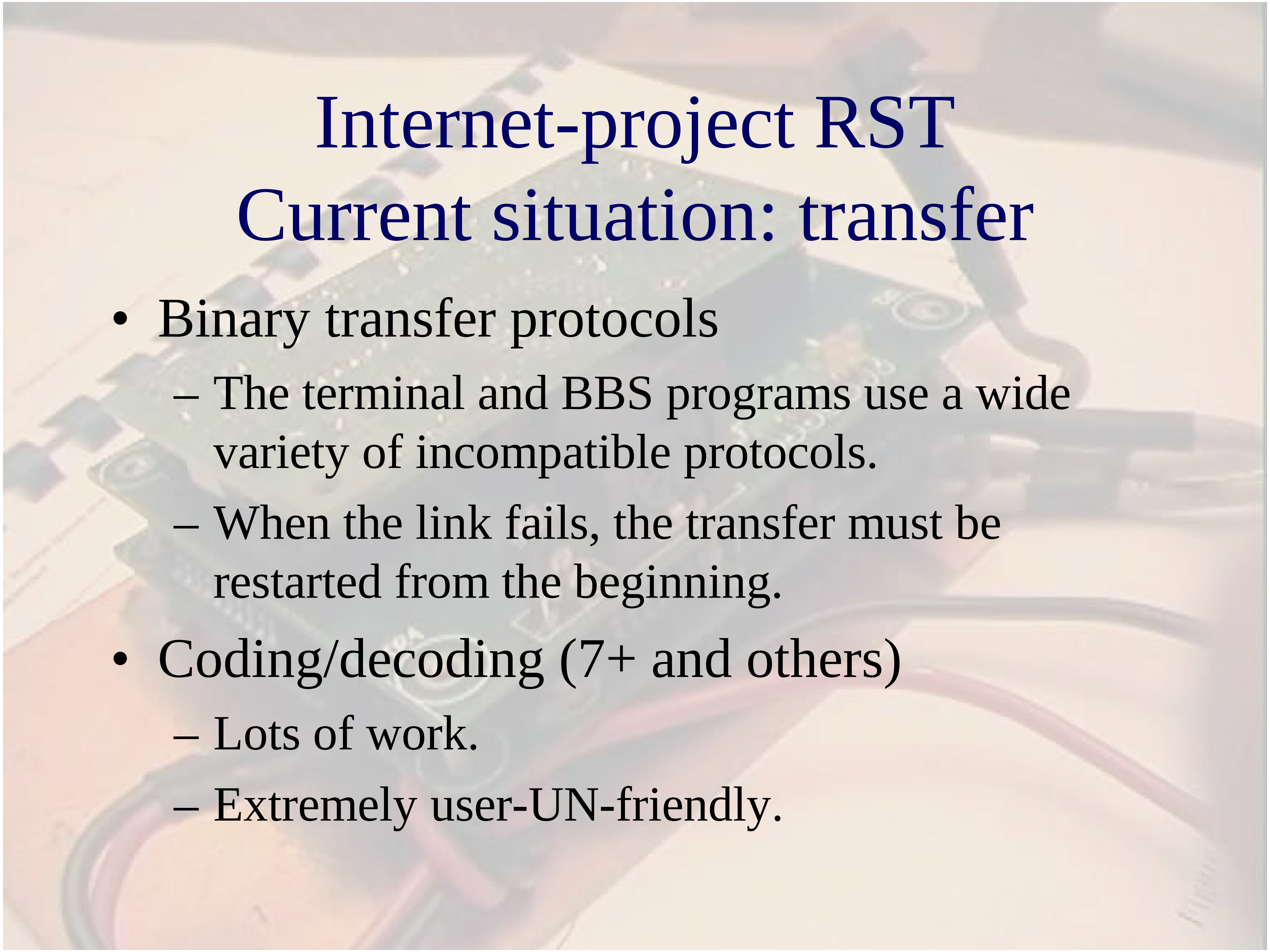


Internet-project RST
Current situation: news

- Also distributed through BBS systems.

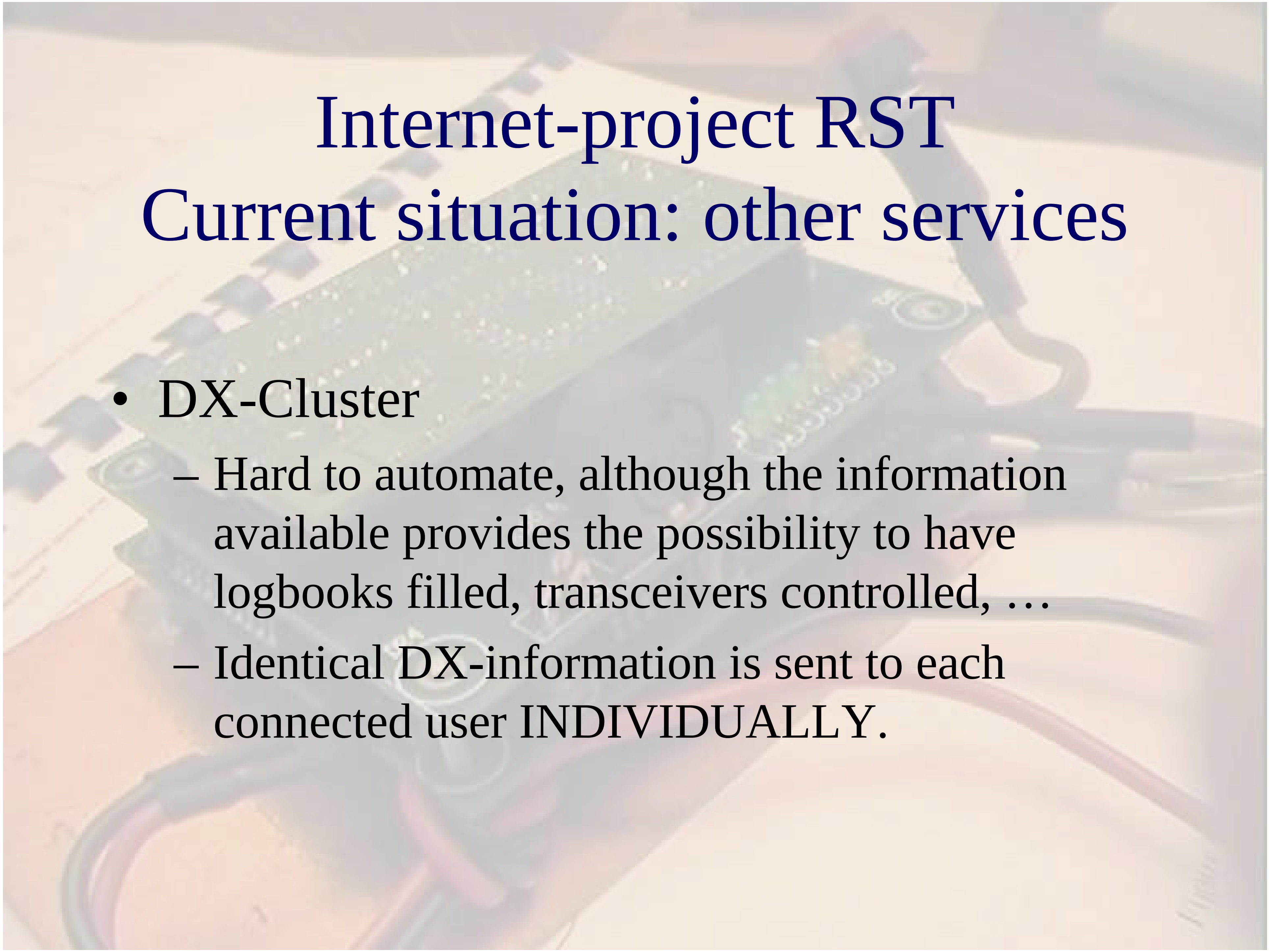
- The bulletins expose an even more remarkable difference between the ‘look & feel’ of the many systems.

- It’s even harder to have bulletins checked for automatically.



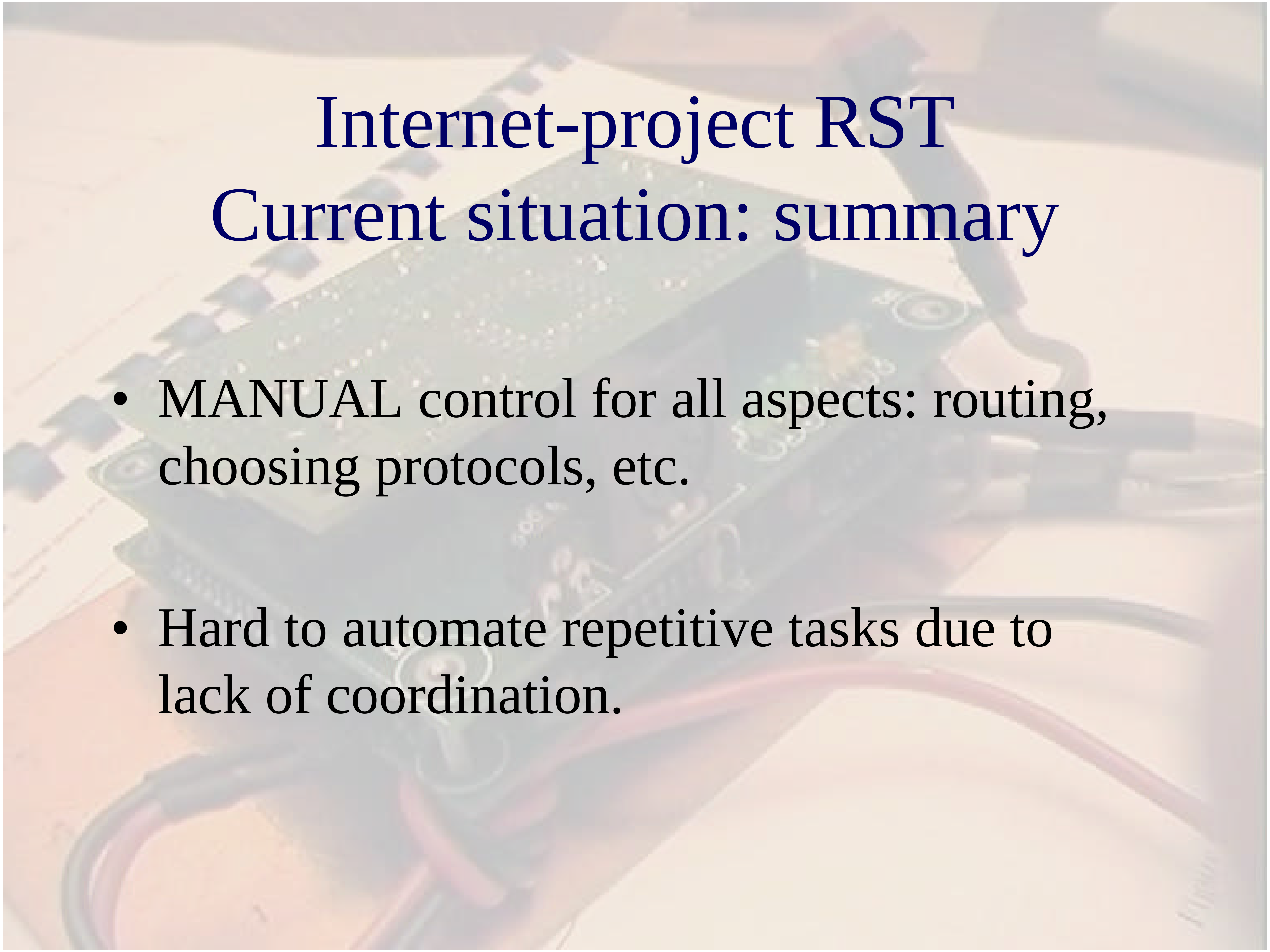
Internet-project RST
Current situation: transfer

- Binary transfer protocols
  - The terminal and BBS programs use a wide variety of incompatible protocols.
  - When the link fails, the transfer must be restarted from the beginning.
- Coding/decoding (7+ and others)
  - Lots of work.
  - Extremely user-UN-friendly.



Internet-project RST
Current situation: other services

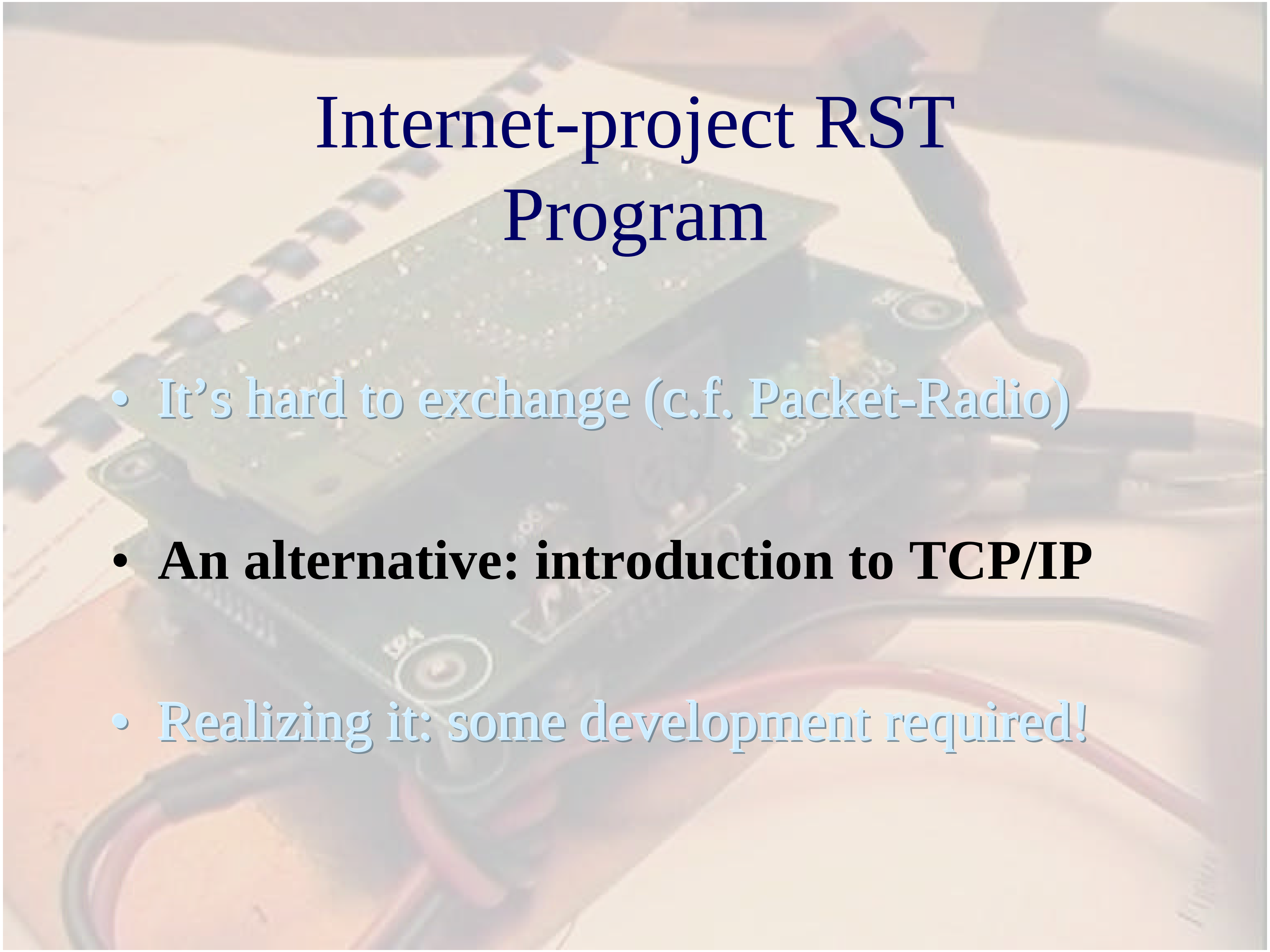
- DX-Cluster
  - Hard to automate, although the information available provides the possibility to have logbooks filled, transceivers controlled, …
  - Identical DX-information is sent to each connected user INDIVIDUALLY.



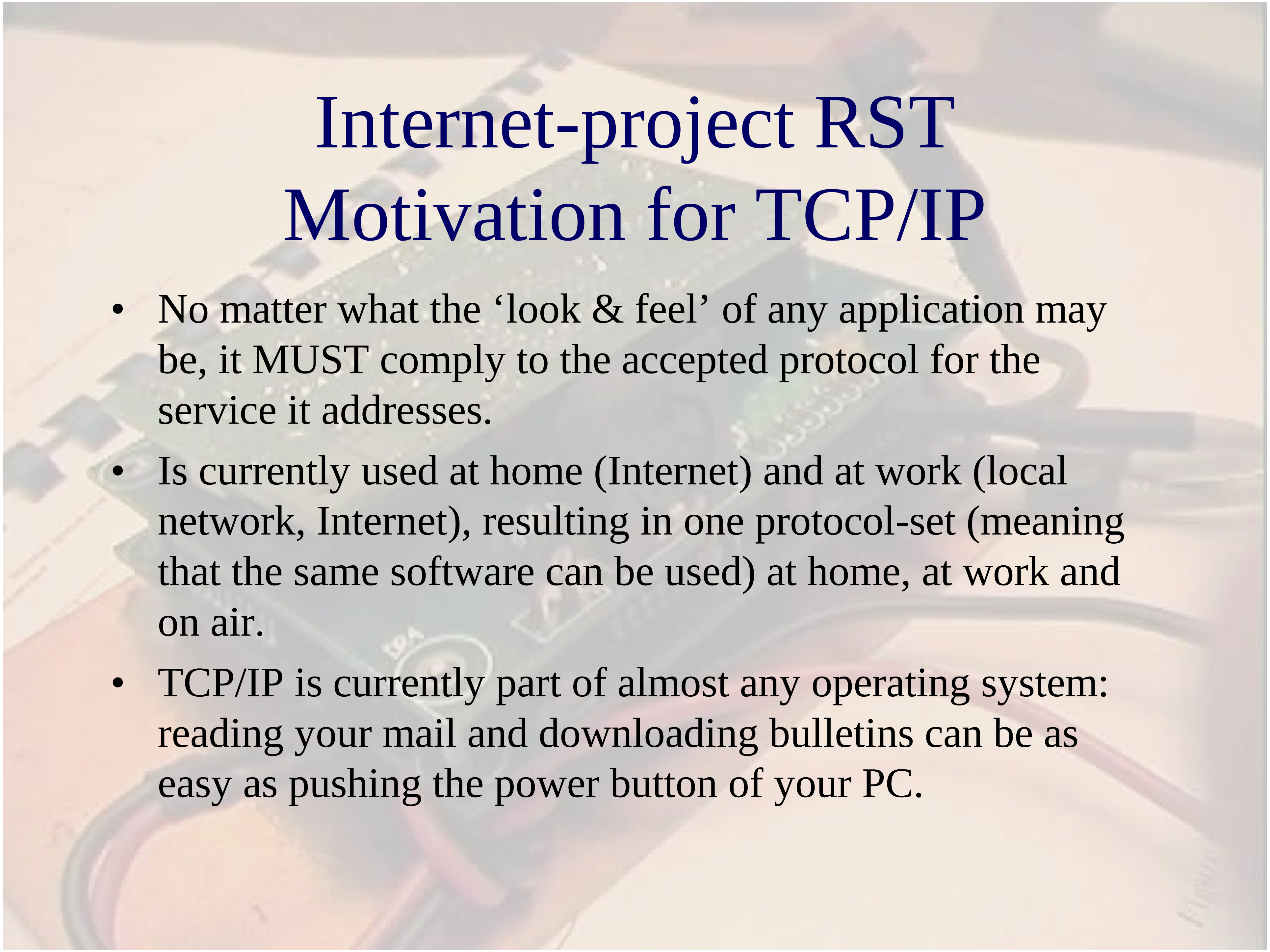
Internet-project RST
Current situation: summary

- MANUAL control for all aspects: routing, choosing protocols, etc.
- Hard to automate repetitive tasks due to lack of coordination.

Internet-project RST Program

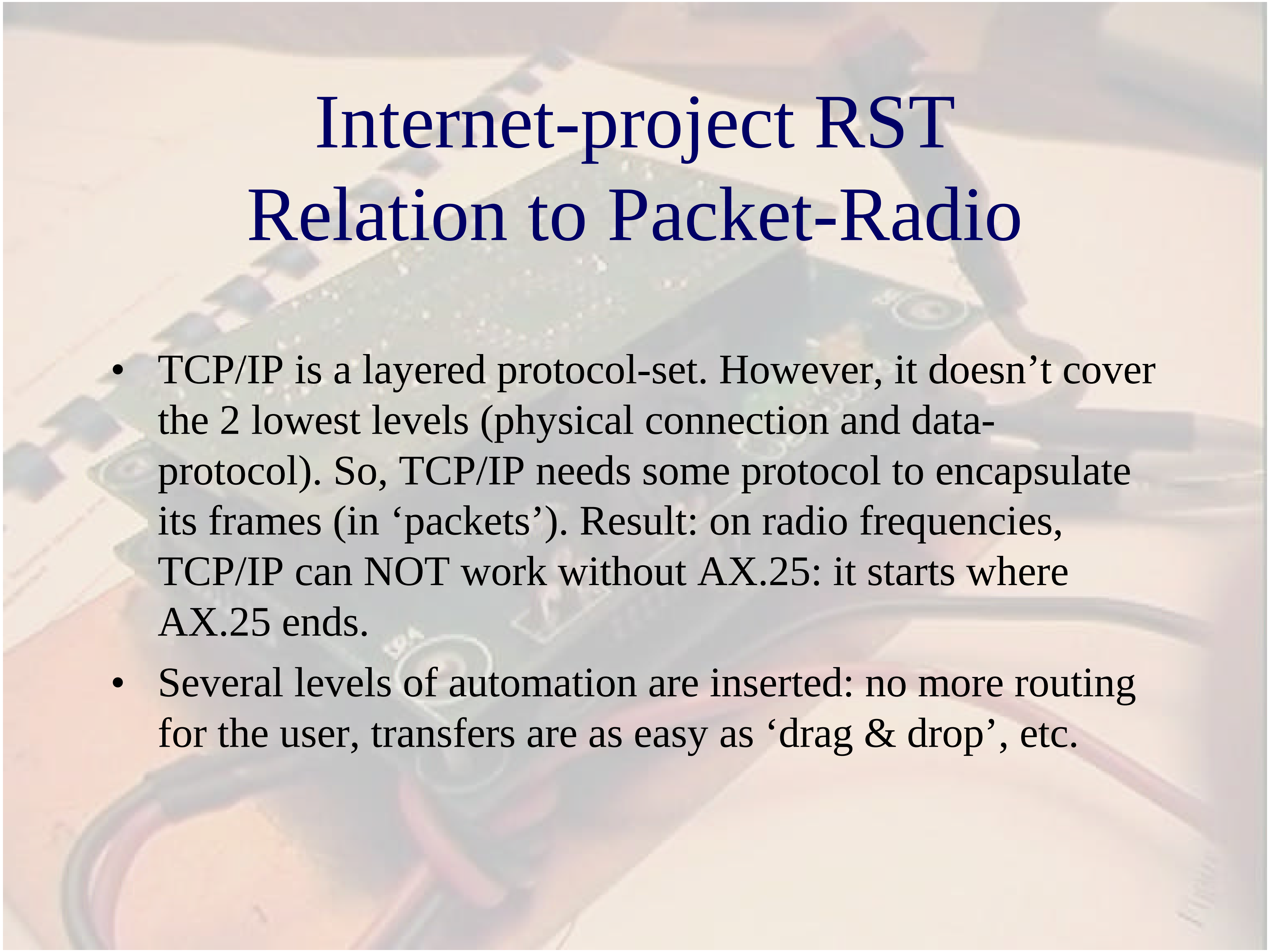


- It’s hard to exchange (c.f. Packet-Radio)
- **An alternative: introduction to TCP/IP**
- Realizing it: some development required!



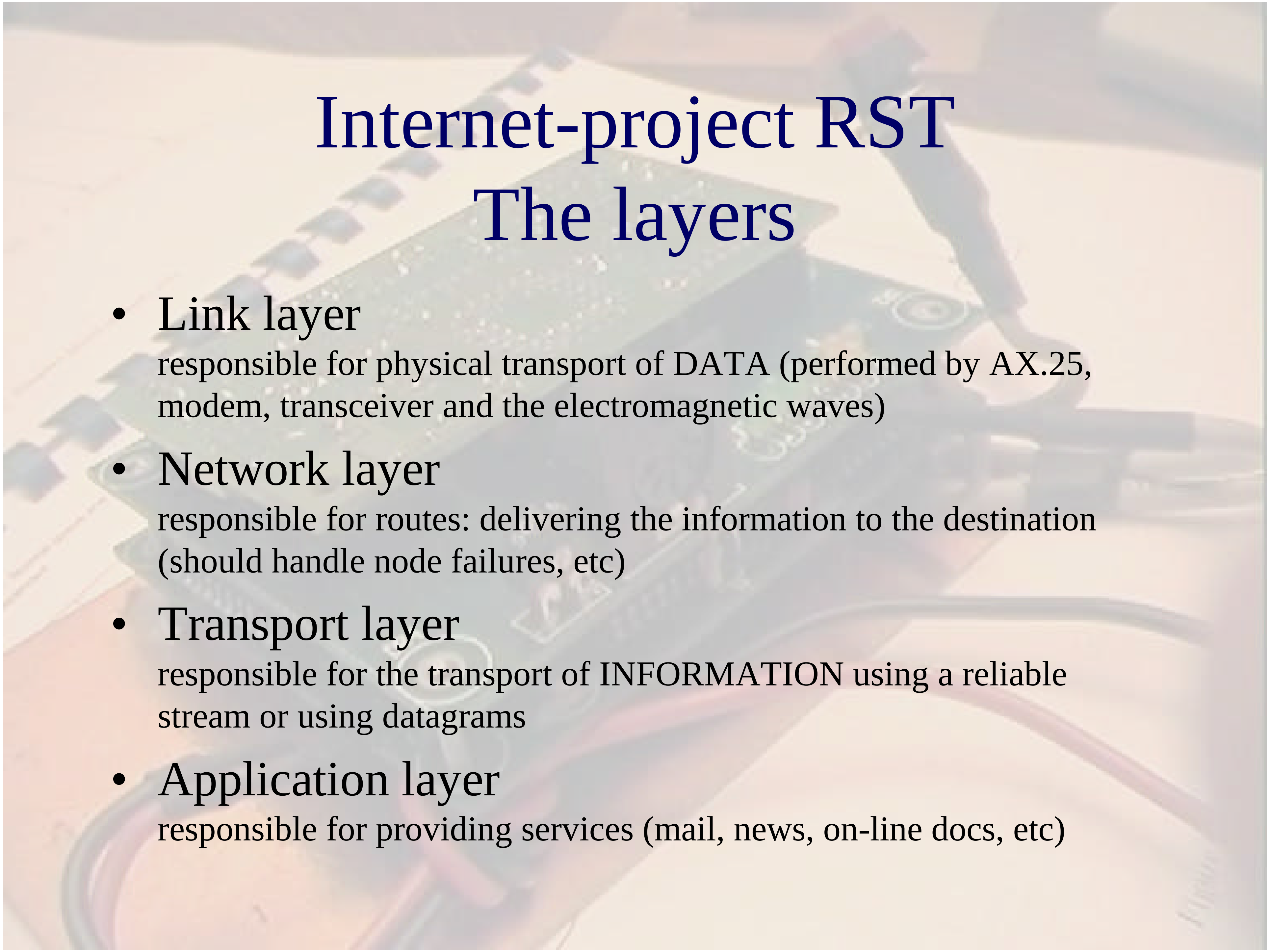
Internet-project RST
Motivation for TCP/IP

- No matter what the ‘look & feel’ of any application may be, it MUST comply to the accepted protocol for the service it addresses.
- Is currently used at home (Internet) and at work (local network, Internet), resulting in one protocol-set (meaning that the same software can be used) at home, at work and on air.
- TCP/IP is currently part of almost any operating system: reading your mail and downloading bulletins can be as easy as pushing the power button of your PC.



Internet-project RST Relation to Packet-Radio

- TCP/IP is a layered protocol-set. However, it doesn’t cover the 2 lowest levels (physical connection and data-protocol). So, TCP/IP needs some protocol to encapsulate its frames (in ‘packets’). Result: on radio frequencies, TCP/IP can NOT work without AX.25: it starts where AX.25 ends.
- Several levels of automation are inserted: no more routing for the user, transfers are as easy as ‘drag & drop’, etc.

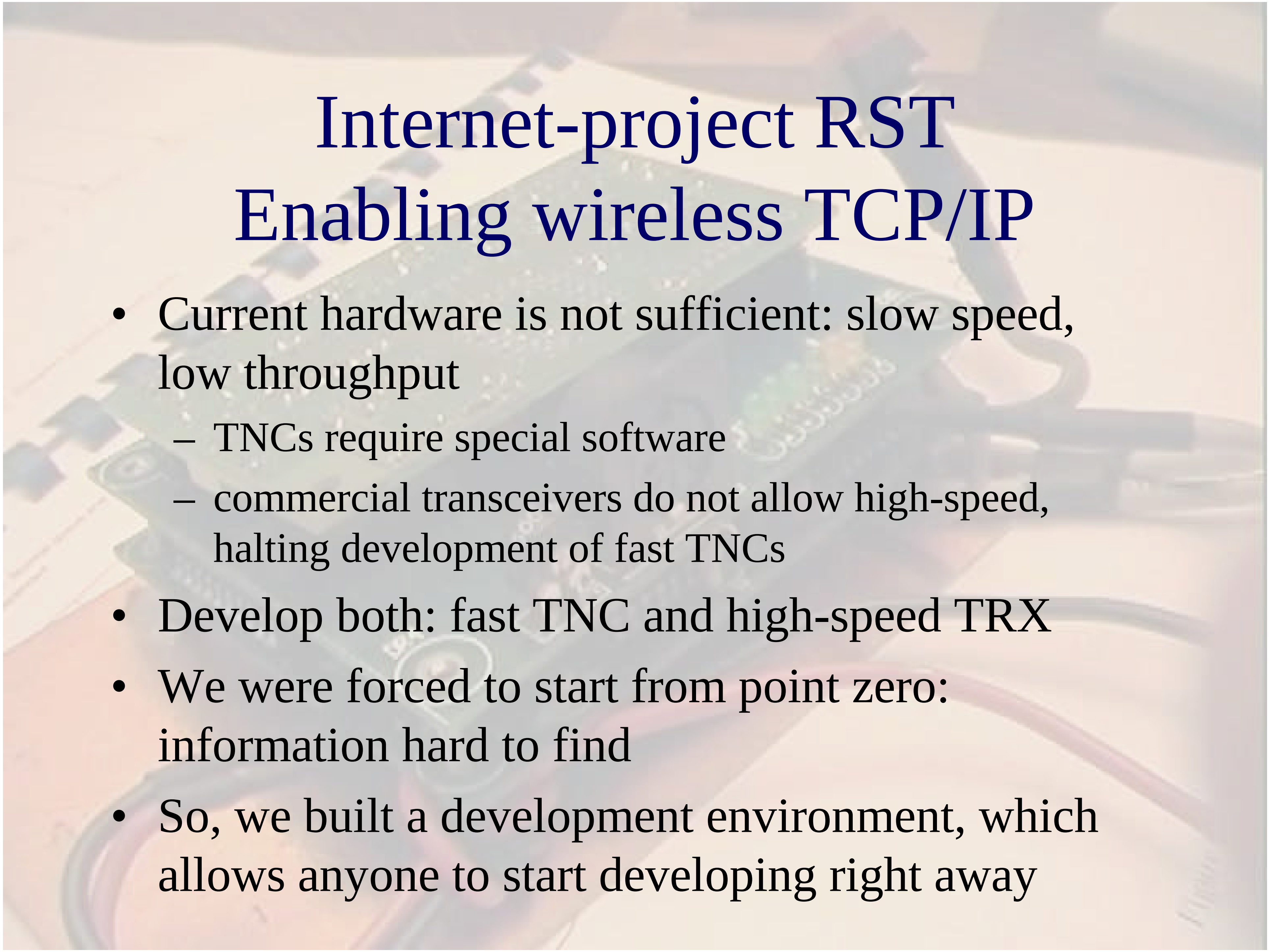


Internet-project RST
The layers

- Link layer
  responsible for physical transport of DATA (performed by AX.25, modem, transceiver and the electromagnetic waves)
- Network layer
  responsible for routes: delivering the information to the destination (should handle node failures, etc)
- Transport layer
  responsible for the transport of INFORMATION using a reliable stream or using datagrams
- Application layer
  responsible for providing services (mail, news, on-line docs, etc)

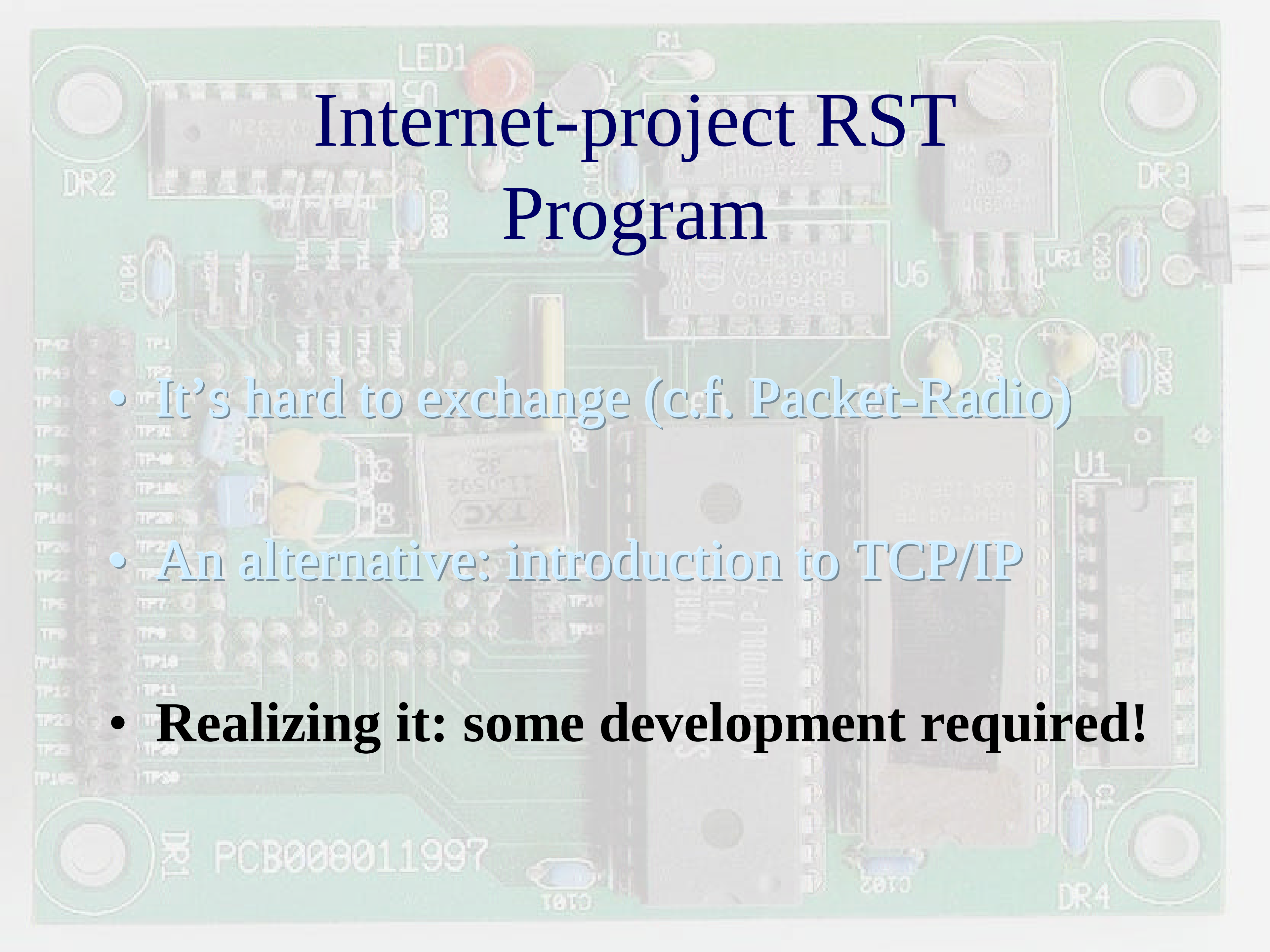
Internet-project RST
The (near) future

- Confrontation:
  - **routes** (handled by IP in network layer): you only need to know the ‘name’ of the destination, and you don’t care about failing nodes
  - **mail** (handled by SMTP & POP in application layer): just boot your computer and use the software YOU want (independent of the server you’re using)
  - **news** (handled by NNTP in application layer): use the software YOU want
  - **transfer** (handled by FTP in application layer): just drag & drop (don’t care about failing nodes, your software, the server software, etc)
  - **other services**: new protocols in application layer
- Transport is handled by the transport layer (cf. “don’t cares”)
- Software development is purely a visual matter: the software only defines the ‘look & feel’, NOT the format of data exchange



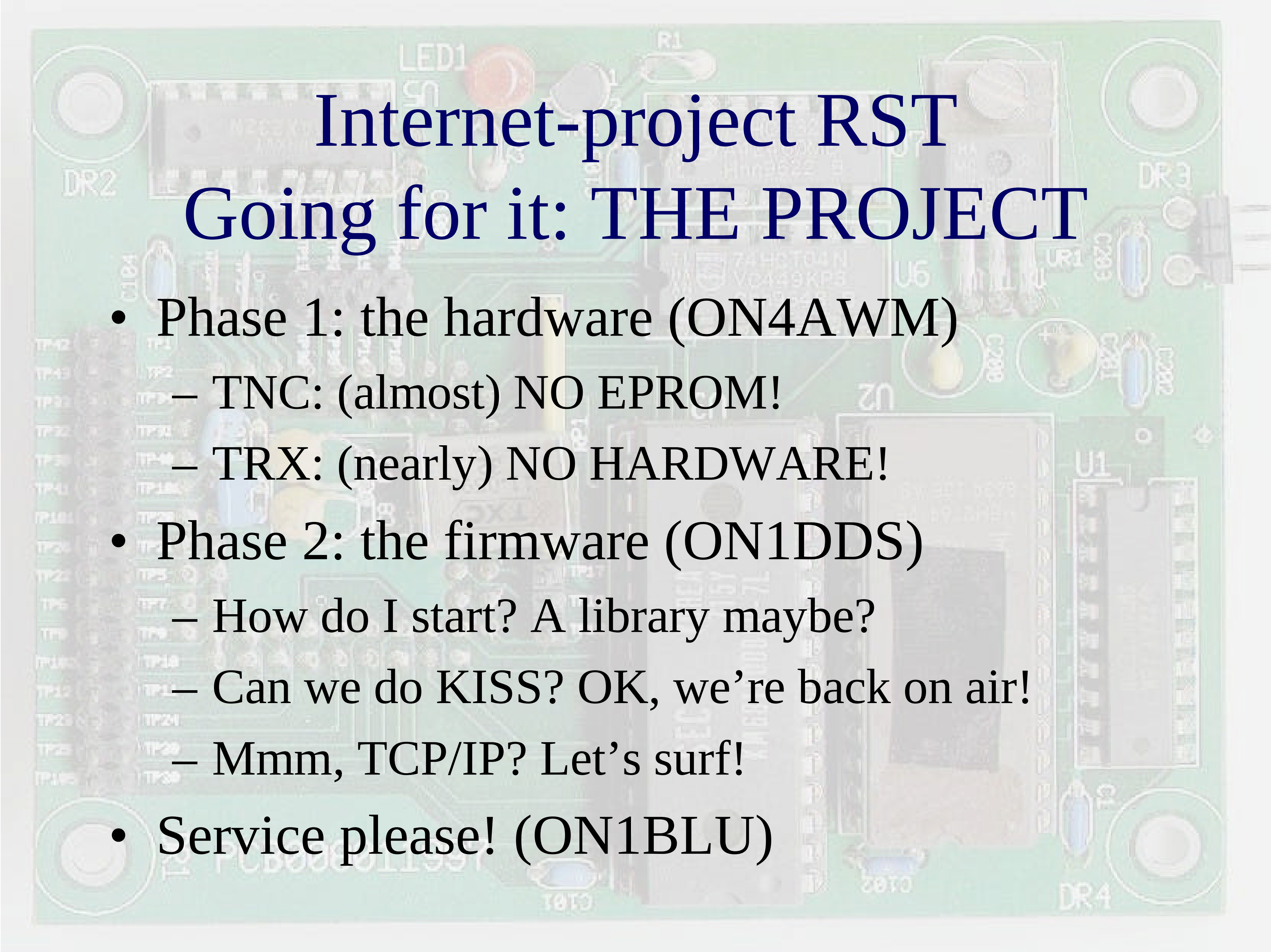
Internet-project RST
Enabling wireless TCP/IP

- Current hardware is not sufficient: slow speed, low throughput
  - TNCs require special software
  - commercial transceivers do not allow high-speed, halting development of fast TNCs
- Develop both: fast TNC and high-speed TRX
- We were forced to start from point zero: information hard to find
- So, we built a development environment, which allows anyone to start developing right away



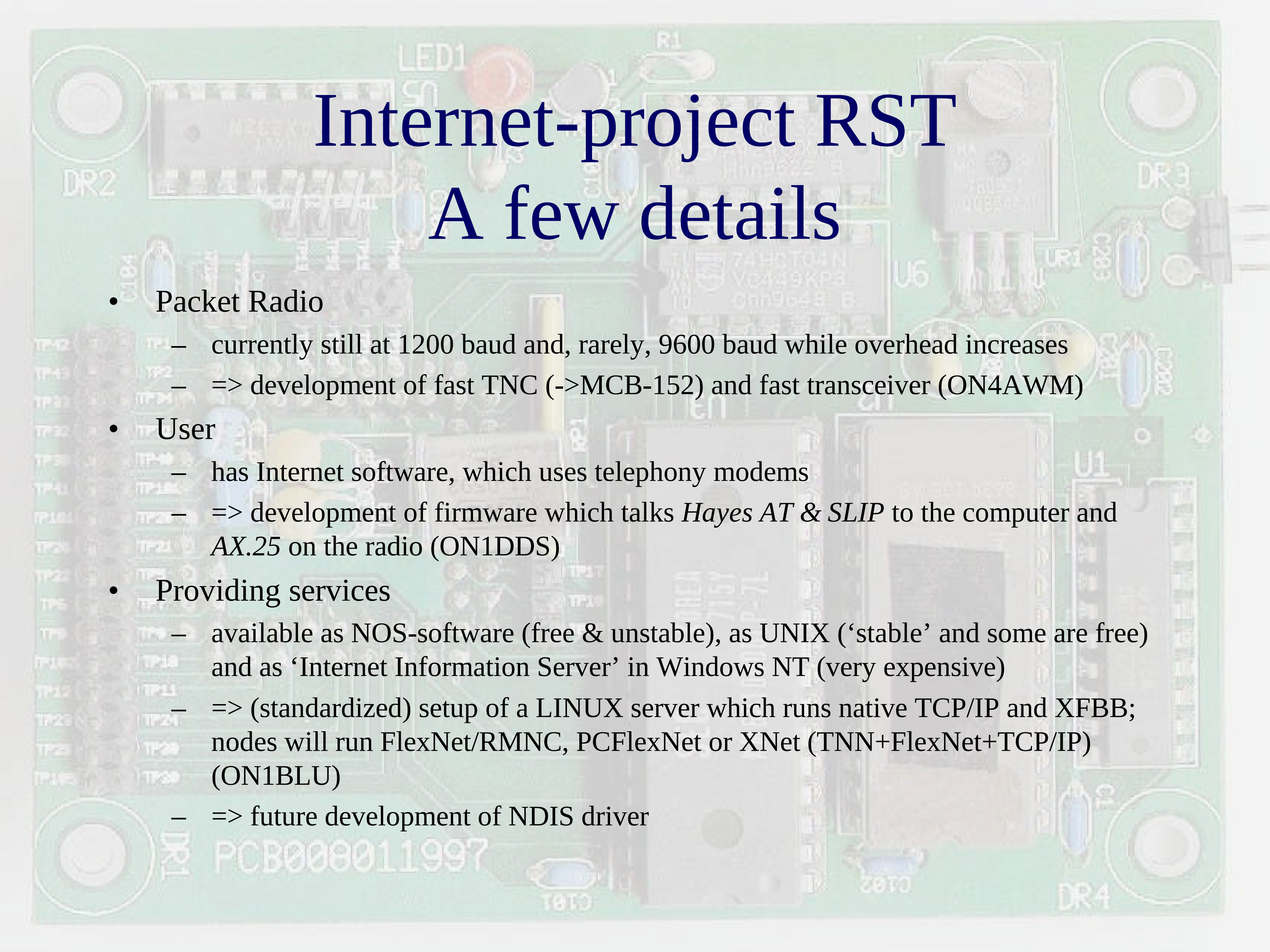
Internet-project RST Program

- It’s hard to exchange (c.f. Packet-Radio)
- An alternative: introduction to TCP/IP
- **Realizing it: some development required!**



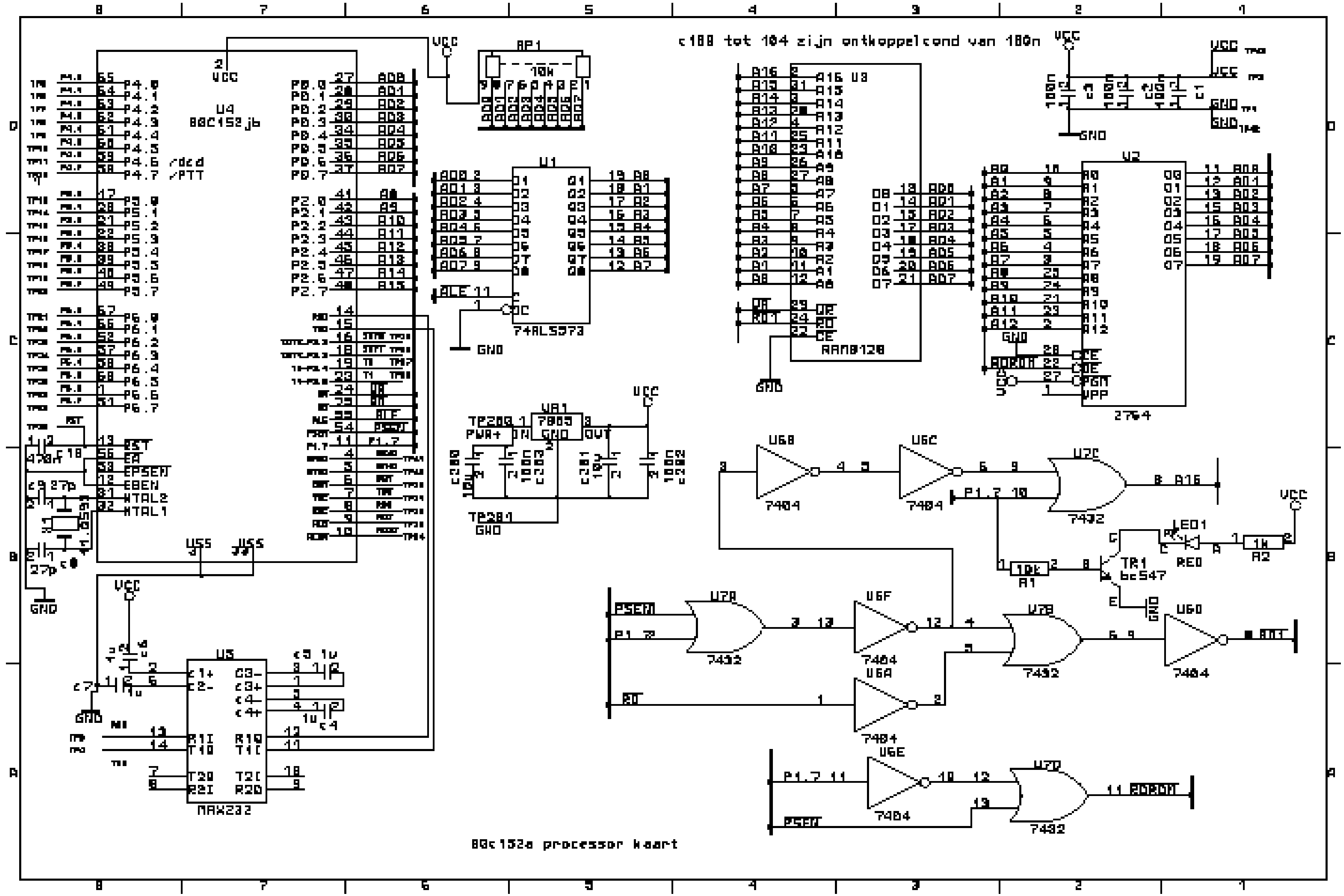
Internet-project RST
Going for it: THE PROJECT

- Phase 1: the hardware (ON4AWM)
  - TNC: (almost) NO EPROM!
  - TRX: (nearly) NO HARDWARE!
- Phase 2: the firmware (ON1DDS)
  - How do I start? A library maybe?
  - Can we do KISS? OK, we’re back on air!
  - Mmm, TCP/IP? Let’s surf!
- Service please! (ON1BLU)



Internet-project RST
A few details

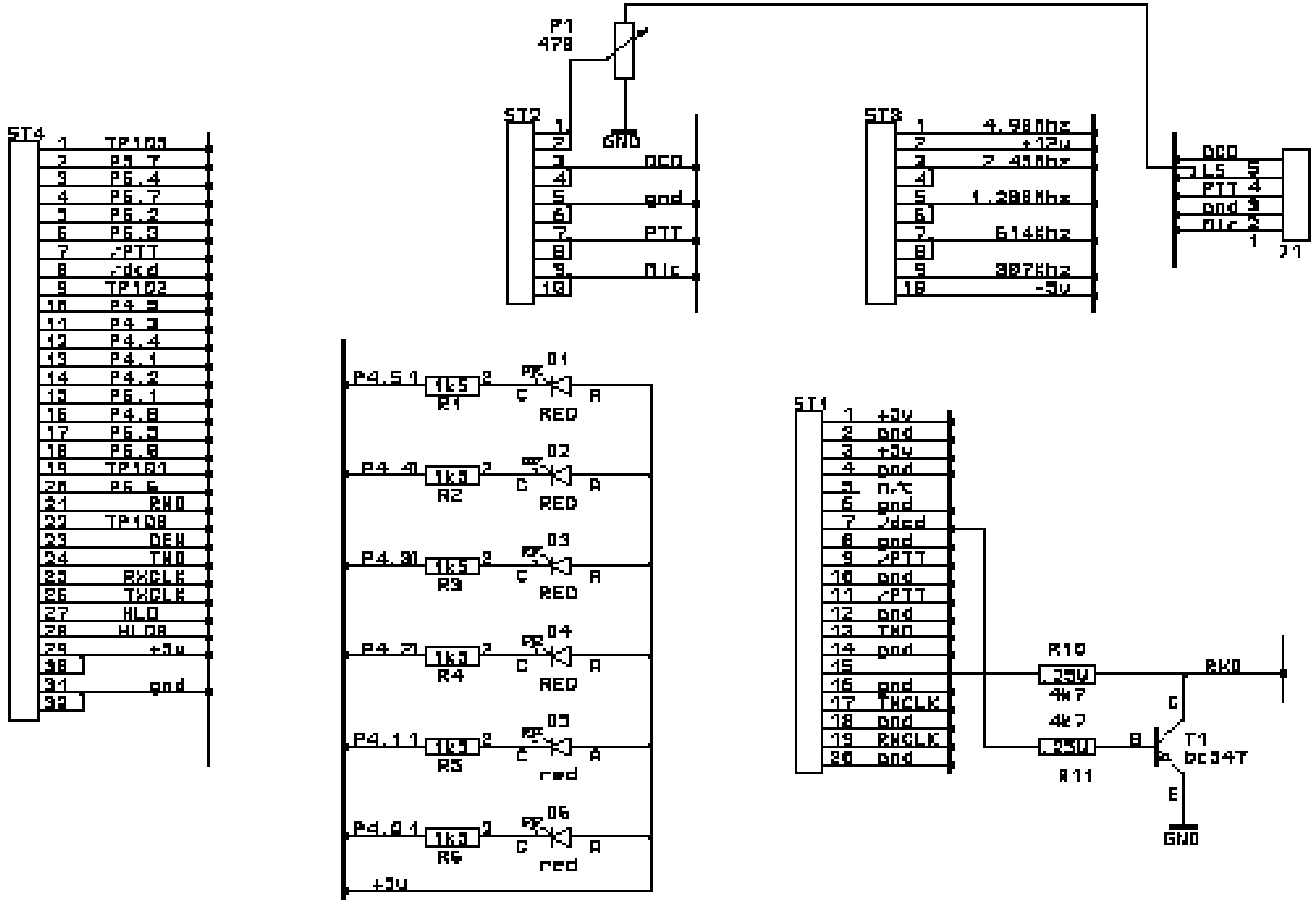
- Packet Radio
  - currently still at 1200 baud and, rarely, 9600 baud while overhead increases
  - => development of fast TNC (->MCB-152) and fast transceiver (ON4AWM)
- User
  - has Internet software, which uses telephony modems
  - => development of firmware which talks *Hayes AT & SLIP* to the computer and *AX.25* on the radio (ON1DDS)
- Providing services
  - available as NOS-software (free & unstable), as UNIX (‘stable’ and some are free) and as ‘Internet Information Server’ in Windows NT (very expensive)
  - => (standardized) setup of a LINUX server which runs native TCP/IP and XFBB; nodes will run FlexNet/RMNC, PCFlexNet or XNet (TNN+FlexNet+TCP/IP) (ON1BLU)
  - => future development of NDIS driver



80c152a processor kaart

Tussenprint voor tnc C--> dk9rr MODEM

c100.. zijn 100n ontkoppelcond


P1
47R
GND
ST4
1 TP103
2 P5.7
3 P6.4
4 P6.7
5 P6.2
6 P6.3
7 /PTT
8 /dcd
9 TP102
10 P4.5
11 P4.3
12 P4.4
13 P4.1
14 P4.2
15 P6.1
16 P4.0
17 P6.5
18 P6.0
19 TP101
20 P6.6
21 RXD
22 TP100
23 DEH
24 TXD
25 RXCLK
26 TXCLK
27 HLD
28 HLDA
29 +5v
30
31 gnd
32
ST2
1
2
3 DCD
4
5 gnd
6
7 PTT
8
9 Mic
10
ST3
1 4.98Mhz
2 +12v
3 2.458Mhz
4
5 1.288Mhz
6
7 614khz
8
9 307khz
10 -5v
DCD
LS 5
PTT 4
gnd 3
Mic 2
1
J1
P4.5 1
1k5
R1
2
D1
C
A
RED
P4.4 1
1k5
R2
2
D2
C
A
RED
P4.3 1
1k5
R3
2
D3
C
A
RED
P4.2 1
1k5
R4
2
D4
C
A
RED
P4.1 1
1k5
R5
2
D5
C
A
red
P4.0 1
1k5
R6
2
D6
C
A
red
+5v
ST1
1 +5v
2 gnd
3 +5v
4 gnd
5 n/c
6 gnd
7 /dcd
8 gnd
9 /PTT
10 gnd
11 /PTT
12 gnd
13 TXD
14 gnd
15
16 gnd
17 TXCLK
18 gnd
19 RXCLK
20 gnd
R10
.25W
4k7
4k7
.25W
R11
RXD
C
B
T1
bc547
E
GND