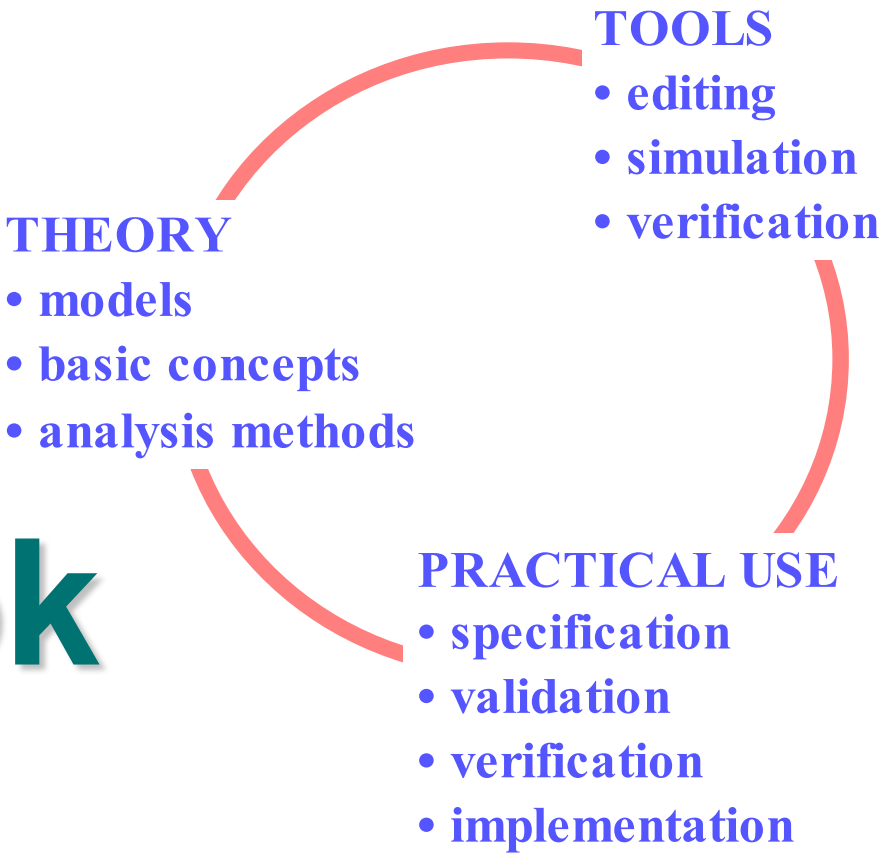
Színezett Petri-hálók



© Kurt Jensen (részlet)

kjensen@daimi.au.dk

Department of Computer Science
Aarhus Egyetem, Dánia

Fordította: Erdélyi Árpád

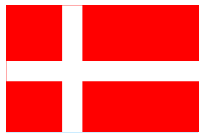
Mi is az a Színezett Petri háló?

- *Modellező nyelv rendszerekhez, ahol szinkronizáció, kommunikáció, és erőforrás megosztás* fontos.

- Egyben *Petri Háló* és *programozási nyelv*.
  - A ***vezérlési struktúrát***, ***szinkronizációt***, ***kommunikációt***, és az ***erőforrás megosztást*** a Petri háló írja le.
  - Az ***adatokért*** és az ***adatkezelésért*** a (Standard ML) funkcionális programozási nyelv a felelős.

- CPN modelleknél *validálás* alatt *szimulációt* értünk, *verifikálás* alatt pedig ***állapotteret*** és ***állapot-invariánsokat***.

- Dániában az Aarhus egyetemen már közel 25 éve kutatják a Színezett Petri hálókat.



Magas-szintű Petri-hálók

- A *CP-háló* és a *normál Petri-háló* közti kapcsolat hasonló a *magasszintű programozási nyelvek* és az *assembly kód* kapcsolatához.
  - Elméletben a két szint kifejezőereje azonos.
  - Gyakorlatban a magas szintű nyelvek sokkal nagyobb modellező erővel rendelkeznek, mivel jobbak a struktúra-leíró képességük, pl. vannak modulok, típusok.
  - Több más magas-szintű Petri háló létezik, de a *Színezett Petri-háló* a legelterjedtebb a gyakorlatban.

CPN-eket nagy rendszerekhez használják

- Egy CPN modell számos *alhálóból* áll.
  - Hasonlóak a *modulokhoz.*
  - Jól-definiált *interfészek* és tiszta *szemantika.*
- A CPN tipikus *ipari alkalmazása*:
  - 10-200 alháló.
  - 50-1000 hely és tranzíció.
  - 10-200 típus.
- Ilyen méretű ipari alkalmazások *elképzelhetetlenek* ezek nélkül:
  - Adattípusok és token értékek.
  - Modulok.
  - Modellező eszköz támogatás.

Felhasználási területek

Protokollok és Hálózatok

- Intelligent Networks at Deutsche Telekom
- IEEE 802.6 Configuration Control at Telstra Research Labs
- Allocation Policies in the Fieldbus Protocol in Japan
- ISDN Services at Telstra Research Laboratories
- Protocol for an Audio/Video System at Bang & Olufsen
- TCP Protocols at Hewlett-Packard
- Local Area Network at University of Las Palmas
- UPC Algorithms in ATM Networks at University of Aarhus
- BRI Protocol in ISDN Networks
- Network Management System at RC International A/S
- Interprocess Communication in Pool IDA at King's College

Szoftver

- Mobile Phones at Nokia
- Bank Transactions & Interconnect Fabric at Hewlett-Packard
- Mutual Exclusion Algorithm at University of Aarhus
- Distributed Program Execution at University of Aarhus
- Internet Cache at the Hungarian Academy of Science
- Electronic Funds Transfer in the US
- Document Storage System at Bull AG
- ADA Program at Draper Laboratories

Vezérlő rendszerek

- Security and Access Control Systems at Dalcotech A/S
- Mechatronic Systems in Cars at Peugeot-Citroën in France
- European Train Control System in Germany
- Flowmeter System at Danfoss
- Traffic Signals in Brazil
- Chemical Production in Germany
- Model Train System at University of Kiel

Hardver

- VLSI Chip in the US
- Arbiter Cascade at Meta Software Corp.

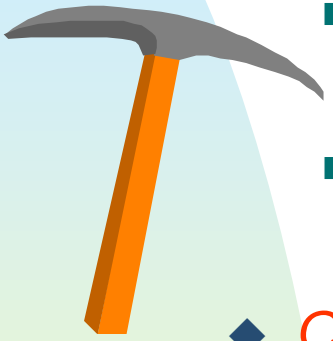
Katonai rendszerek

- Military Communications Gateway in Australia
- Influence Nets for the US Air Force
- Missile Simulator in Australia
- Naval Command and Control System in Canada

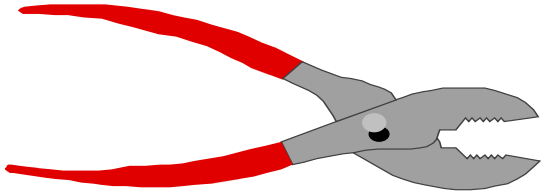
Egyéb

- Bank Courier Network at Shawmut National Coop.
- Nuclear Waste Management Programme in the US

Modellező eszközök



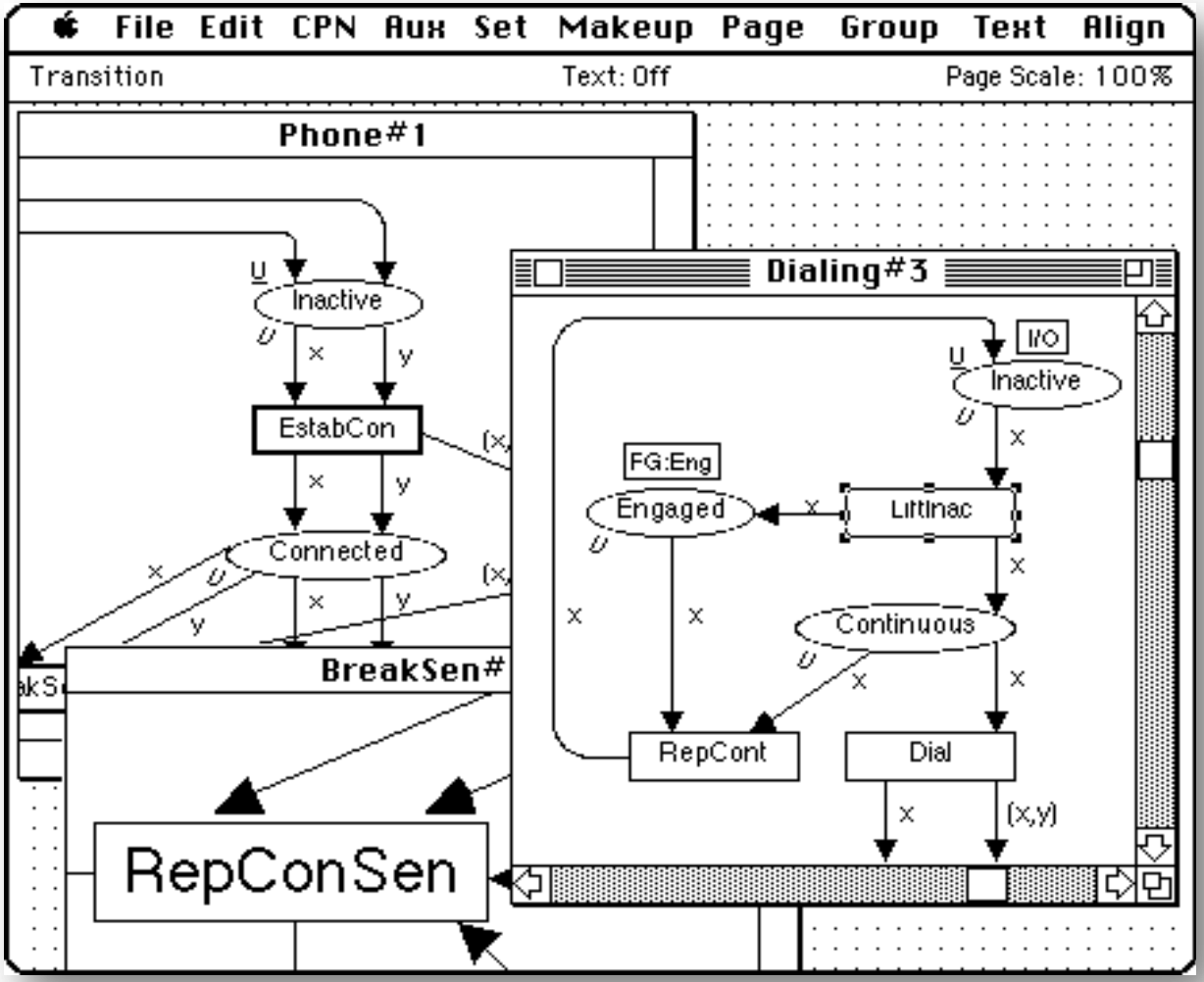
- *Design/CPN* eszközt a 80‘as évek végén és a 90‘as évek elején fejlesztették ki.
  - A maga idejében a legelterjedtebb Petri háló csomag volt.
  - *750 különböző szervezet 50 országban használta* – ezek között *200 kereskedelmi cég van.*
- *CPN Tools* már egy második generációs eszköz a Színezett Petri-hálókhoz.
  - Mára a CPN Tools *átvette a Design/CPN helyét*
  - A fejlesztés *1999-ben kezdődött* és több, mint *20 mérnökév* munka van benne.



Standard ML

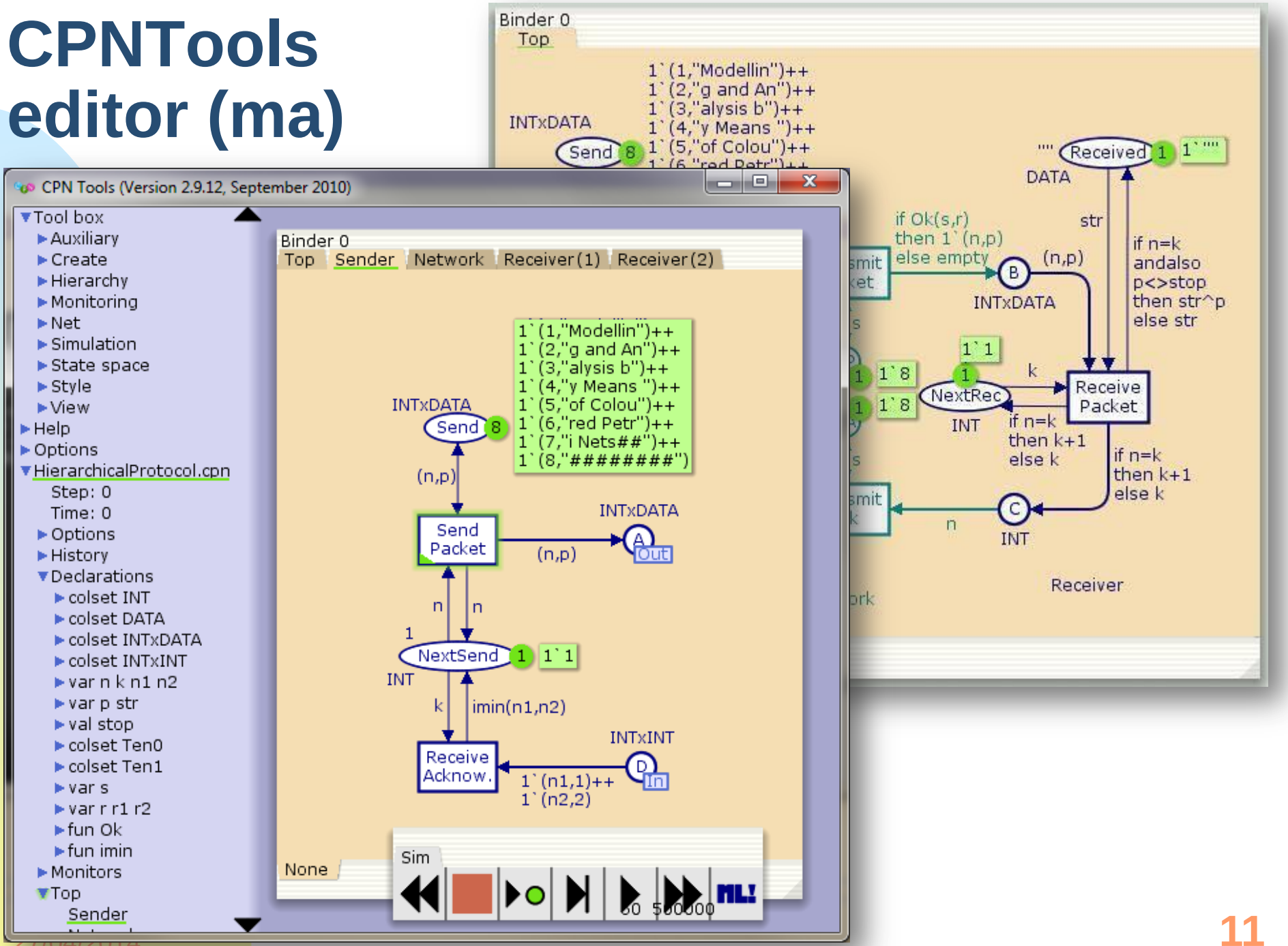
- A típusok, élkifejezések és őrfeltételek *Standard ML-ben* adhatók meg. Ez egy erősen típusos, funkcionális programozási nyelv (*Robin Milner* fejlesztette).
- *Adattípus lehet*:
  - *Atomi* (integer, string, boolean és felsorolás).
  - *Strukturált* (products, record, union, list és subsets).
- Tetszőlegesen bonyolult *függvényeket* és *műveleteket* lehet definiálni benne (pl. polimorfizmus).
- A Standard ML jól-ismert, tesztelt és nagyon általános. Számos *dokumentációja* van.

Design/CPN editor (egykor)

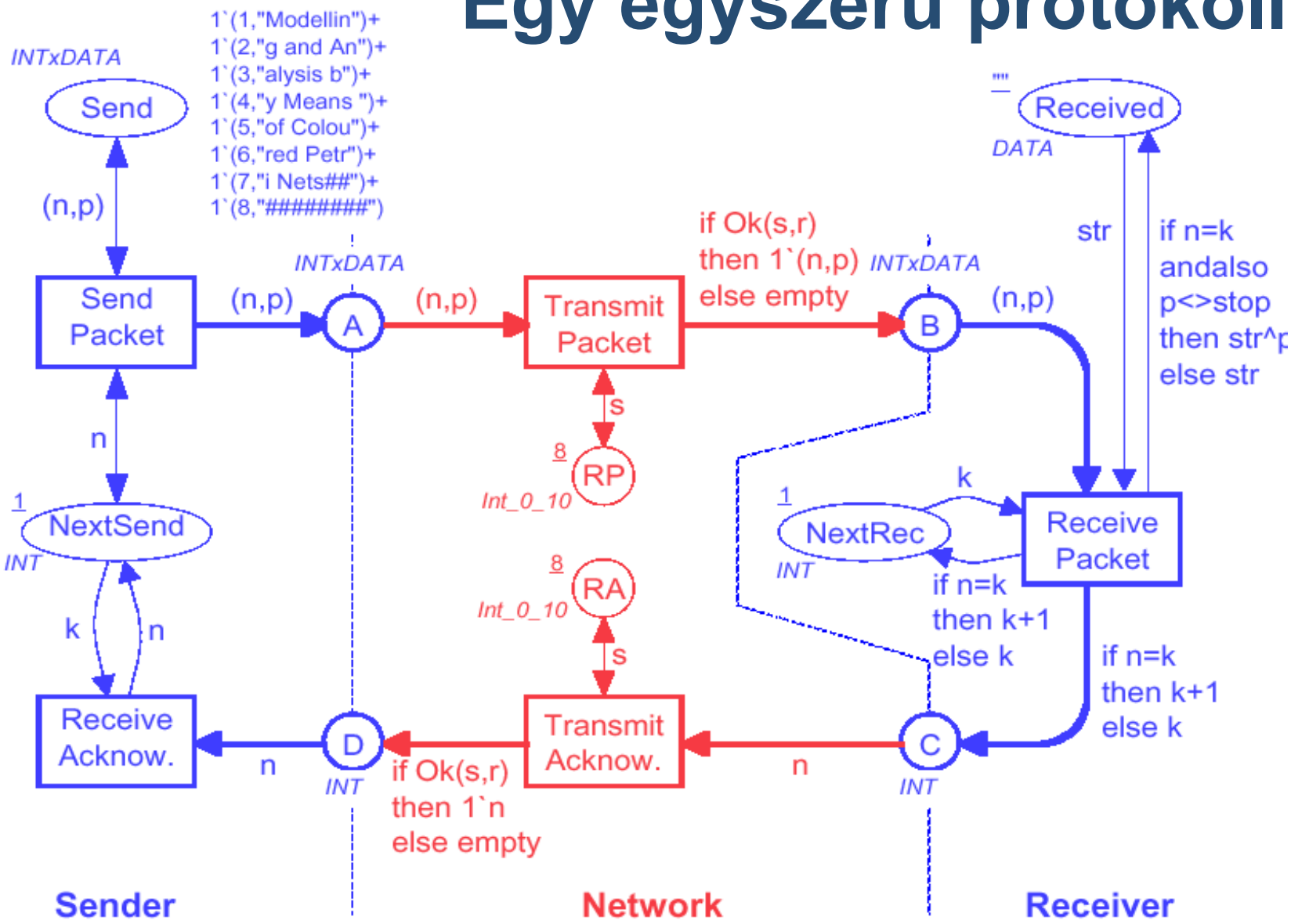


- *WYSIWYG felületről* könnyen módosíthatni tudjuk a modellünket.
- *Szintakszis-vezérelt* – sok *szintaktikai hiba* kiküszöbölhető.

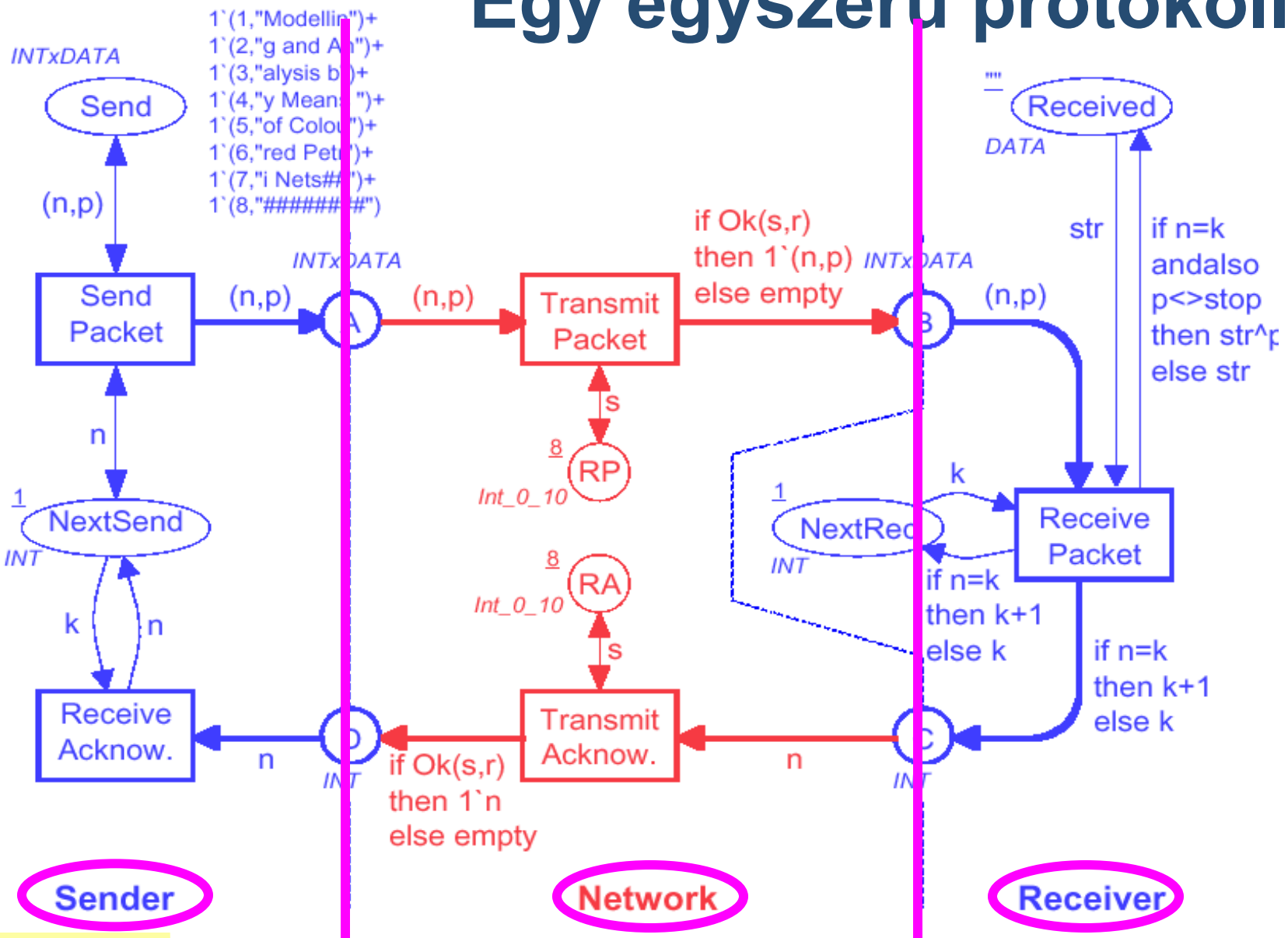
CPNTools editor (ma)



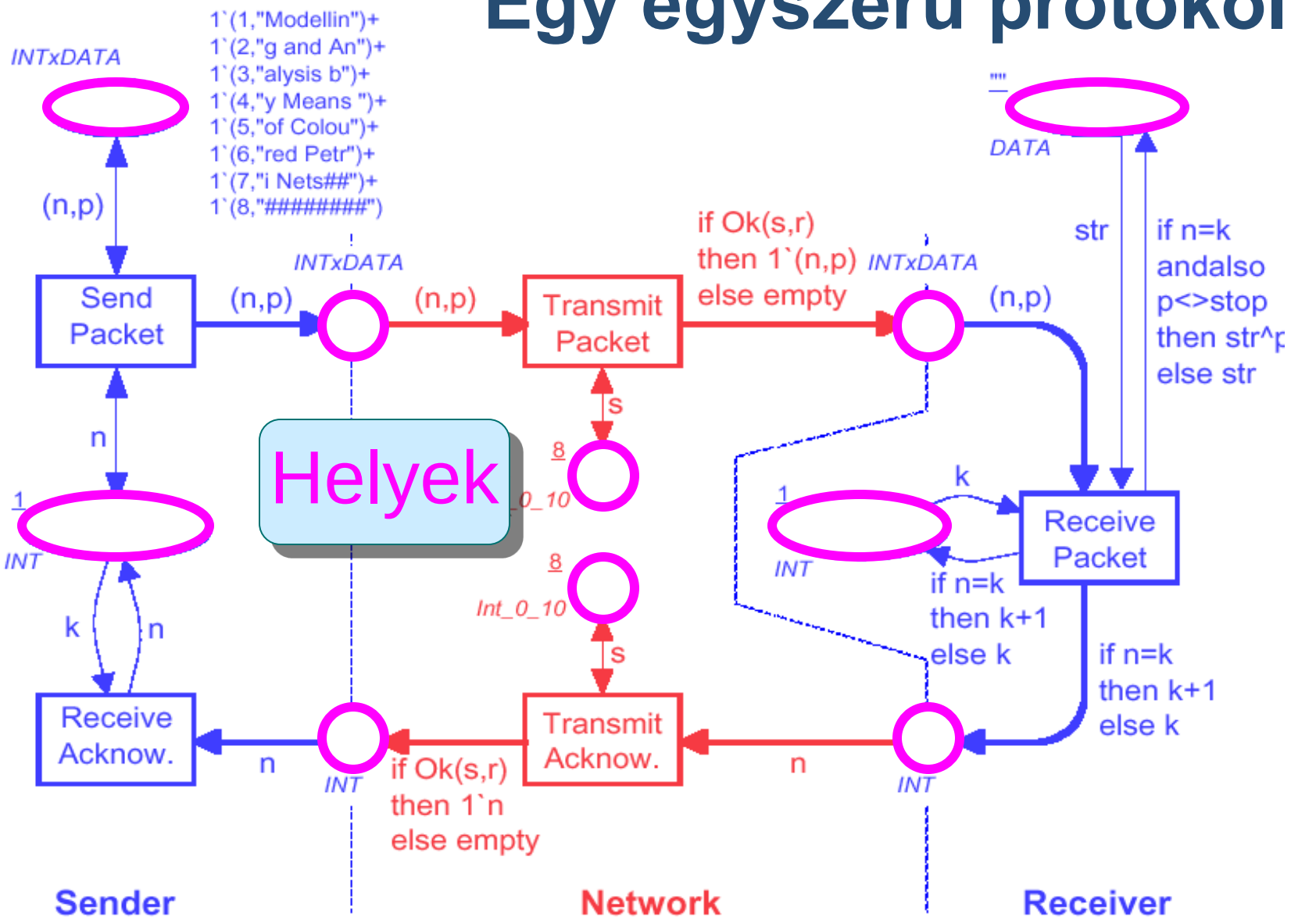
Egy egyszerű protokoll



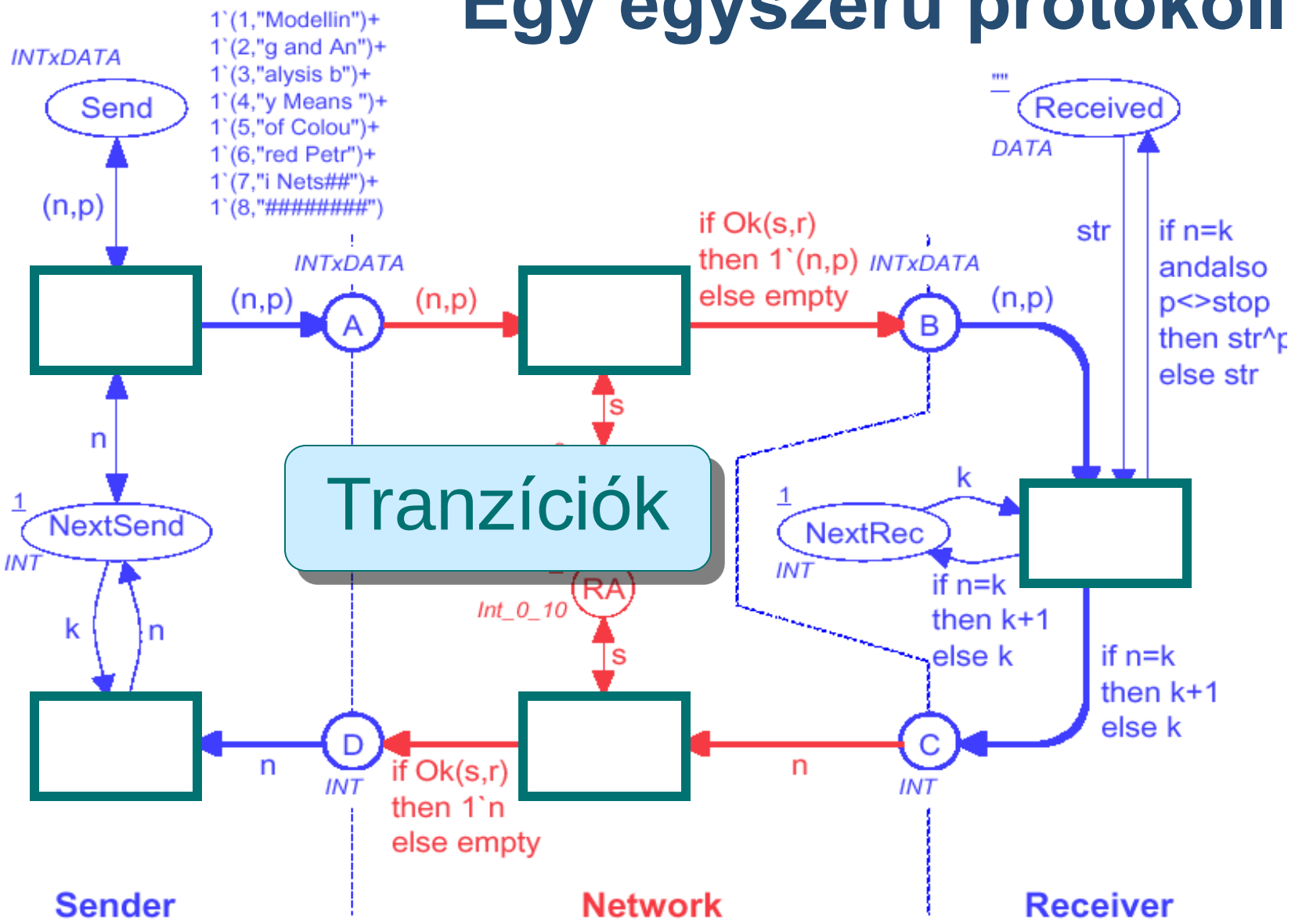
Egy egyszerű protokoll



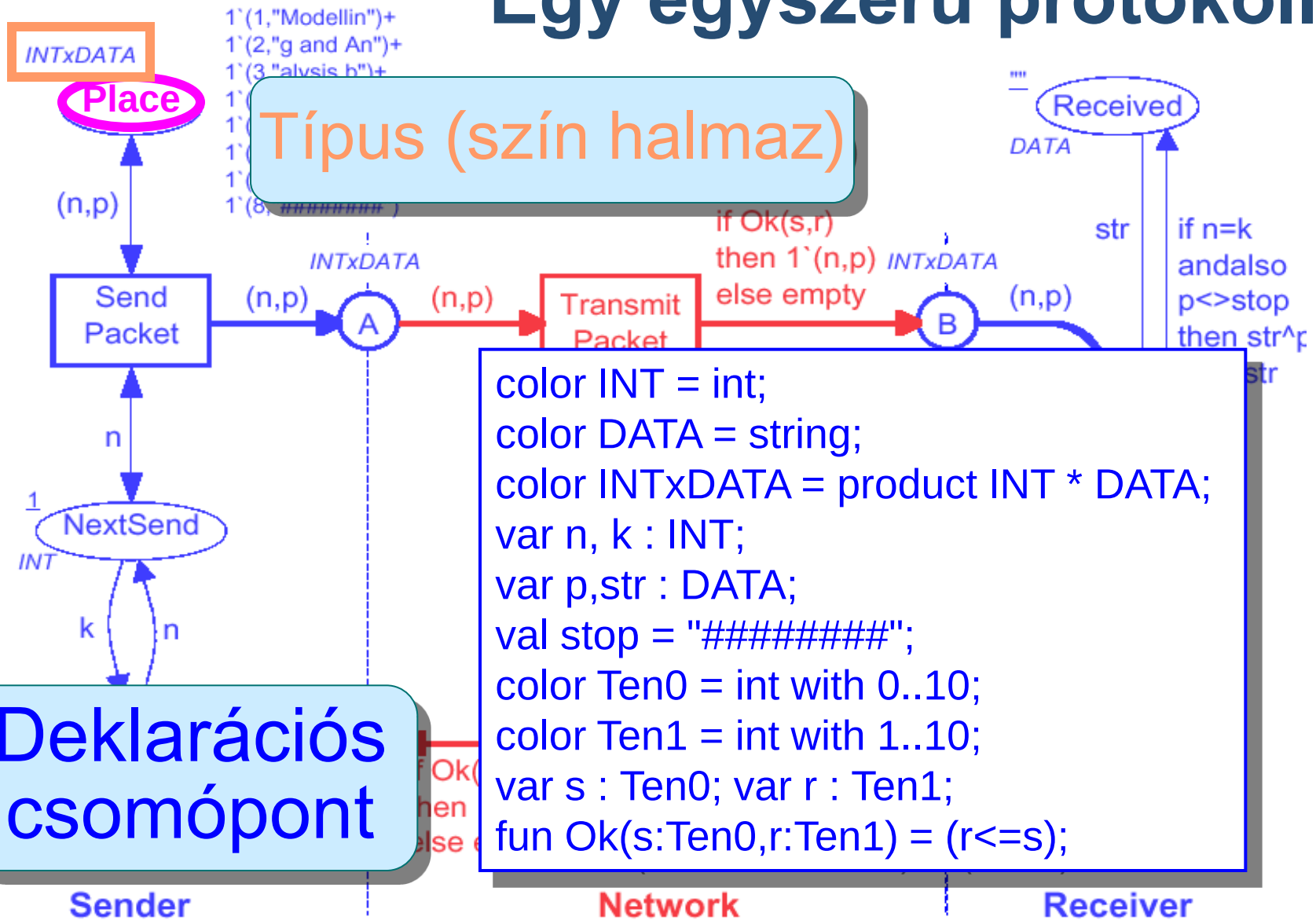
Egy egyszerű protokoll



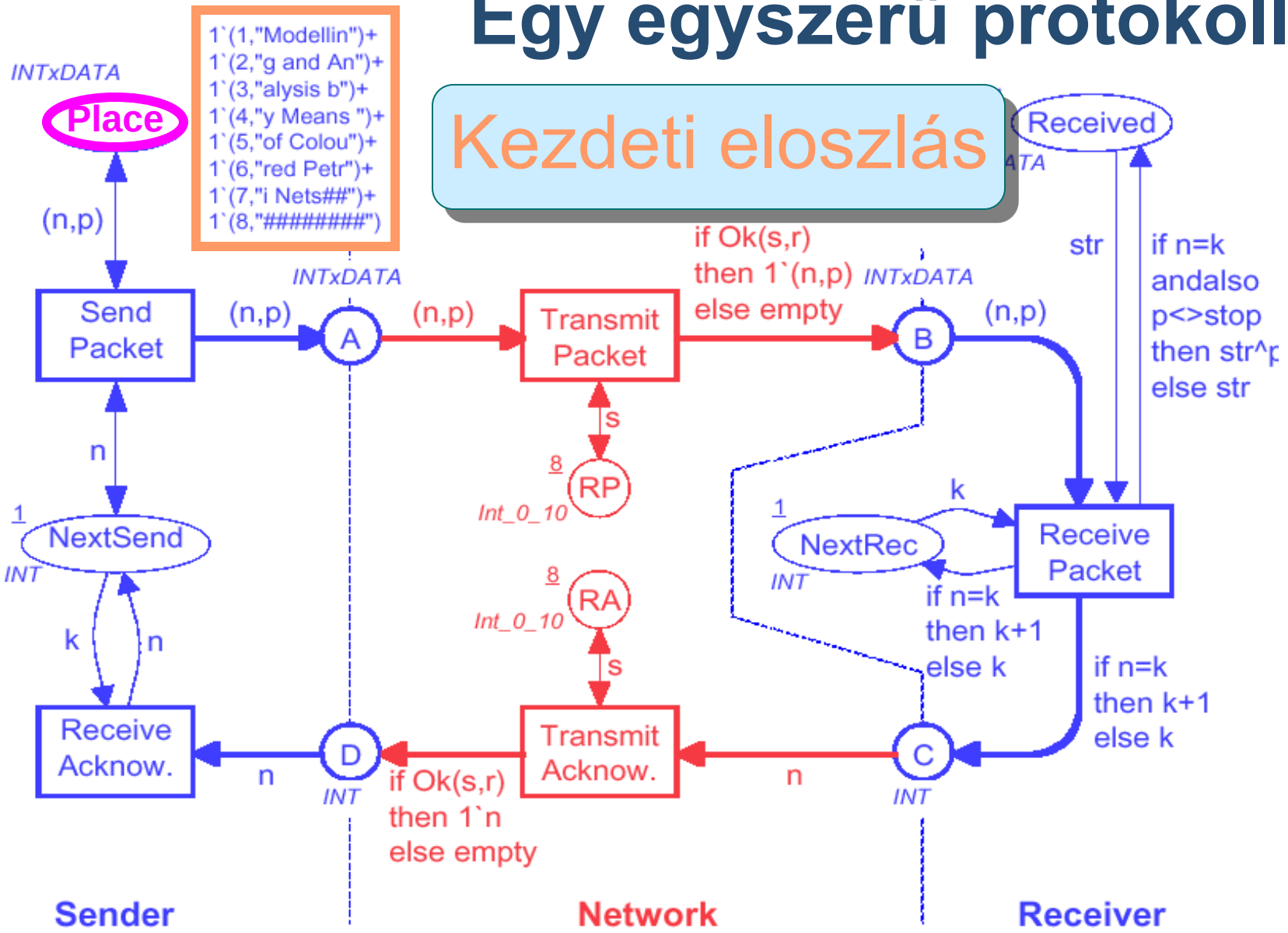
Egy egyszerű protokoll



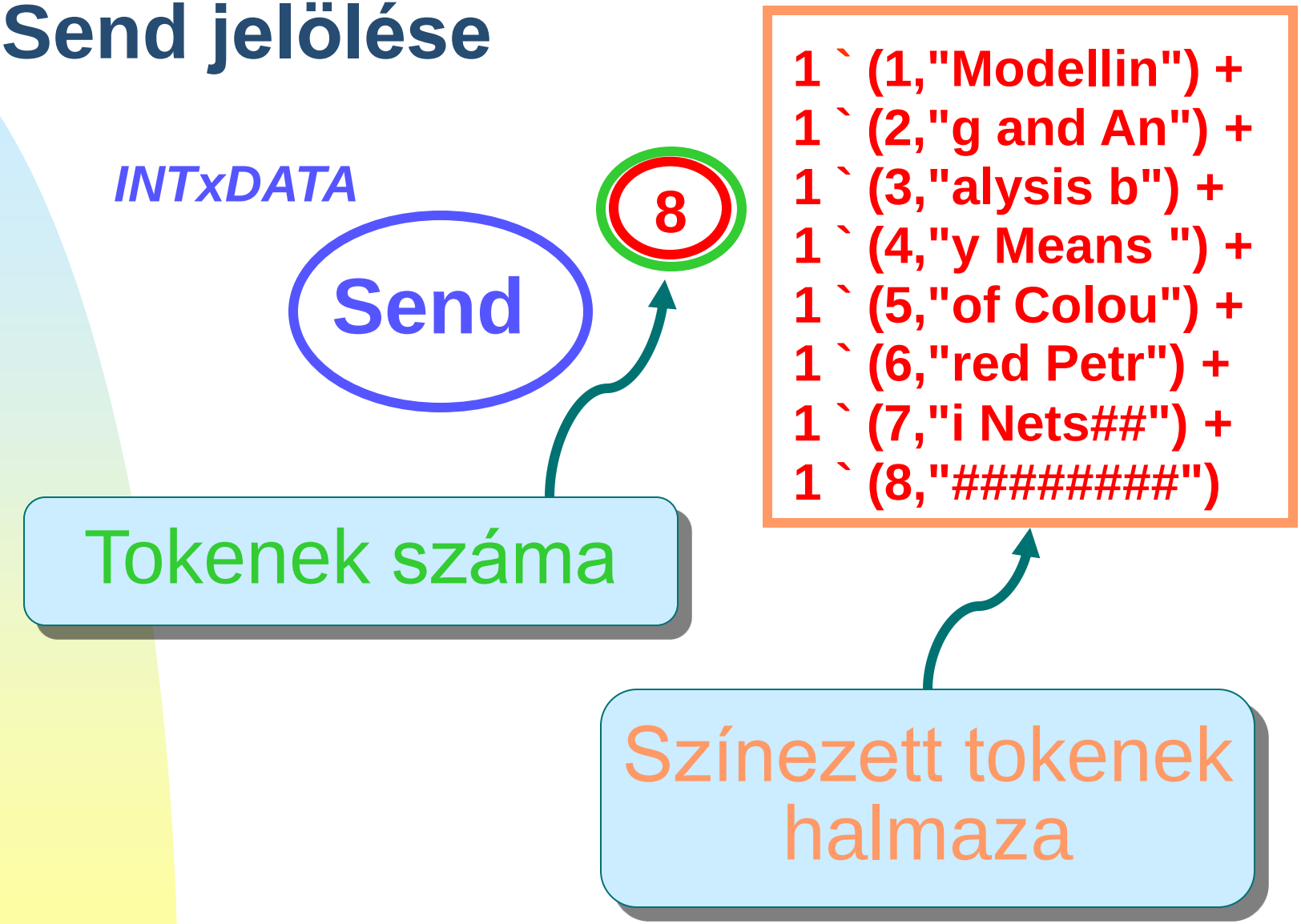
Egy egyszerű protokoll



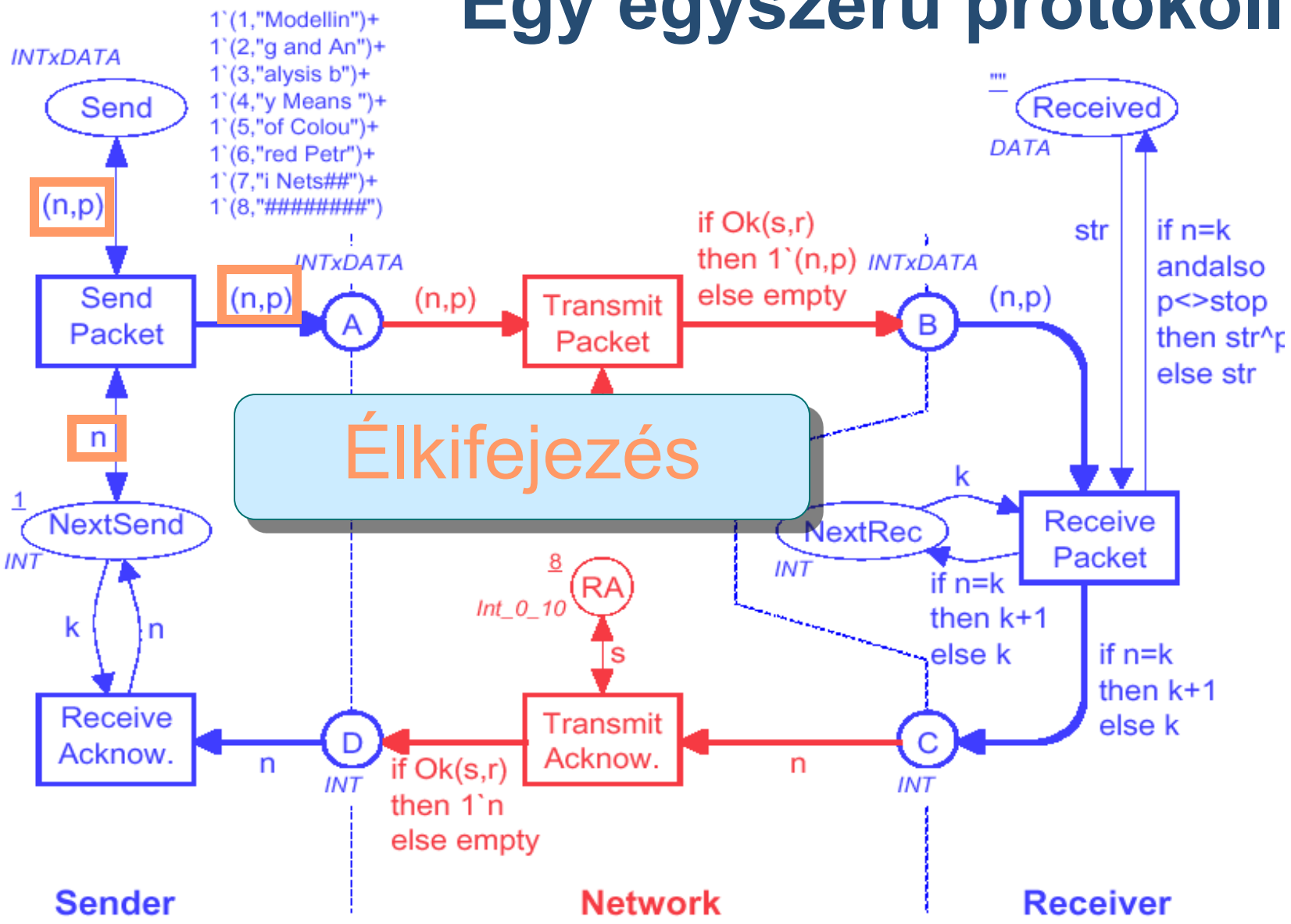
Egy egyszerű protokoll



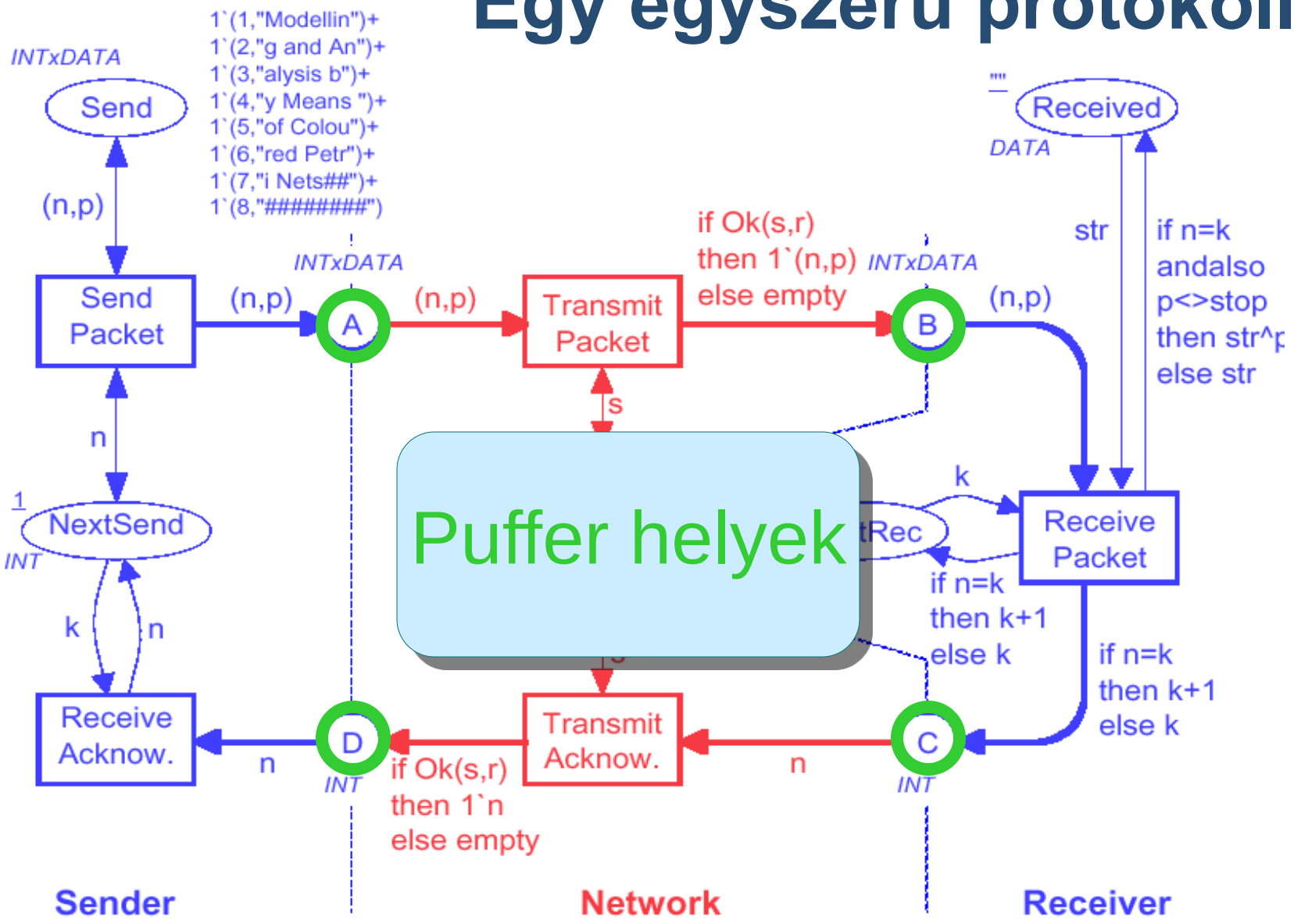
Send jelölése



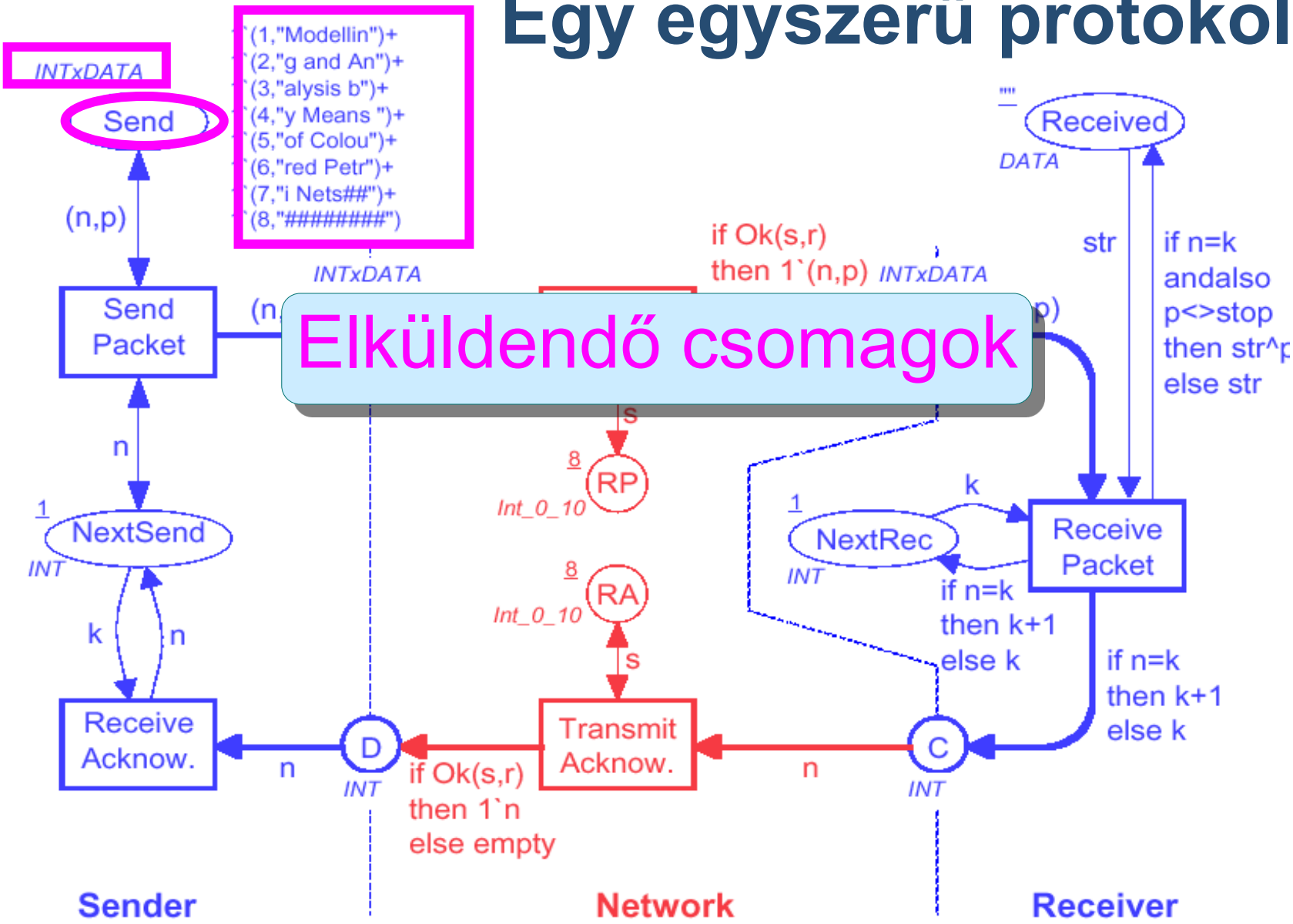
Egy egyszerű protokoll



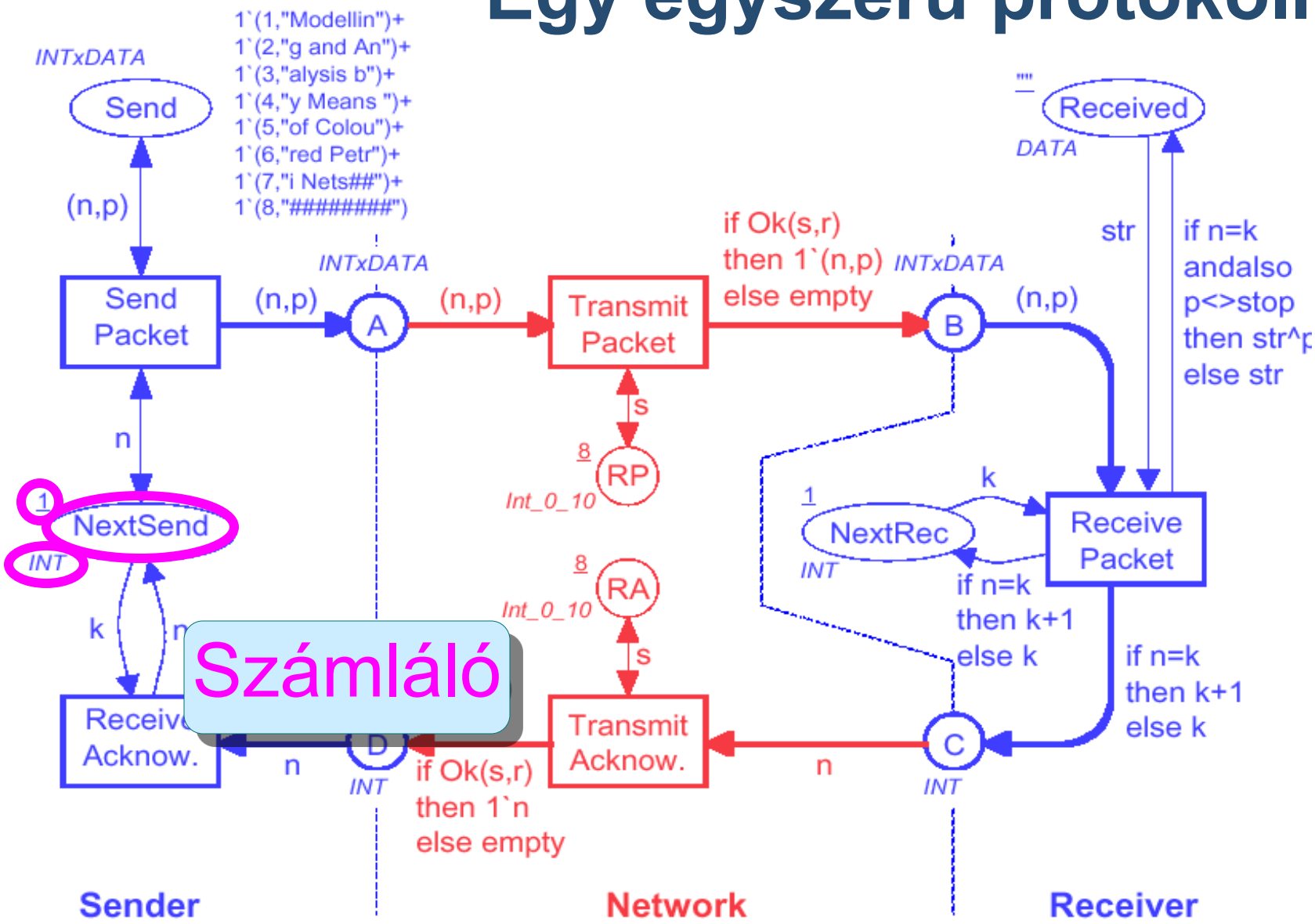
Egy egyszerű protokoll



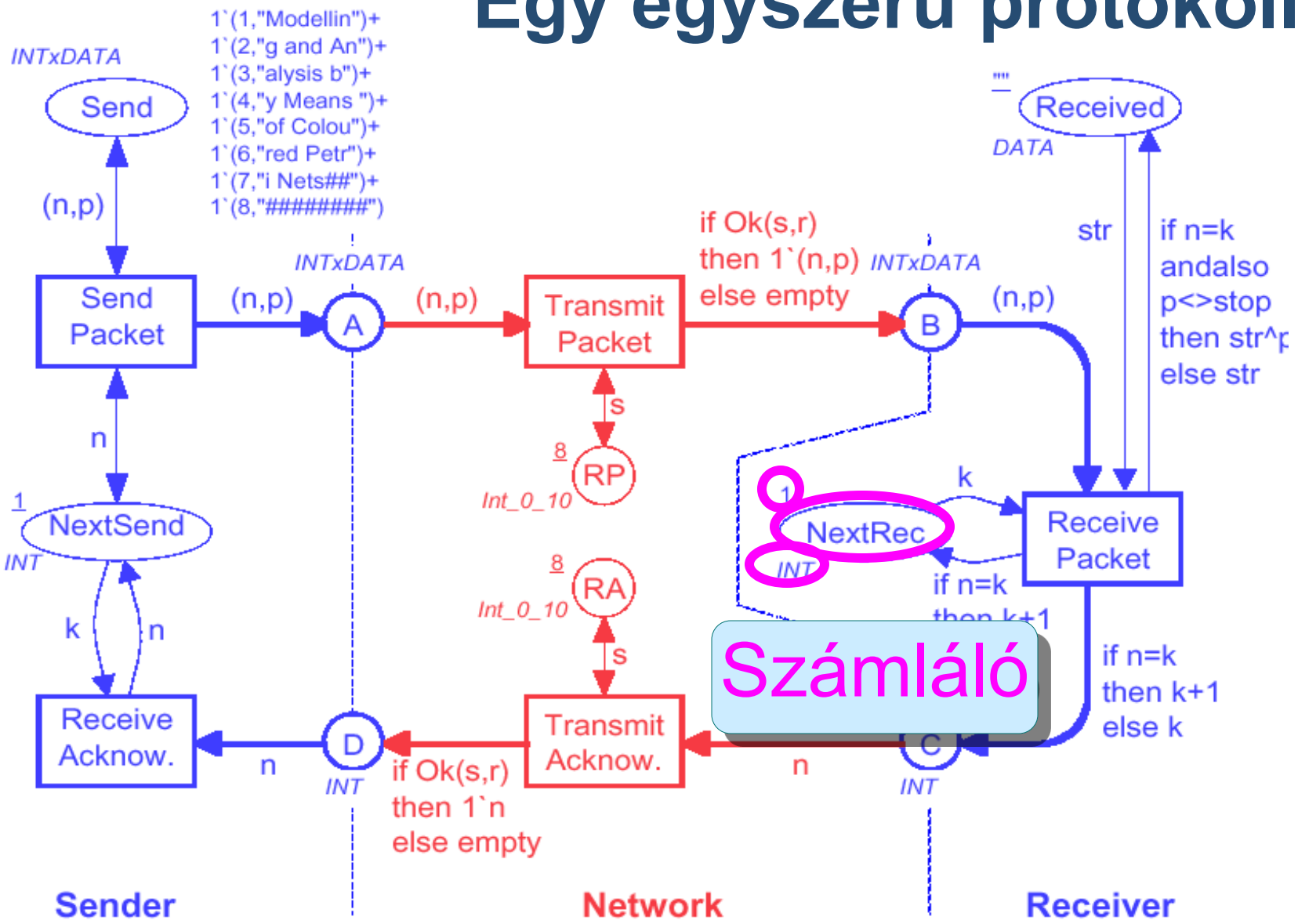
Egy egyszerű protokoll



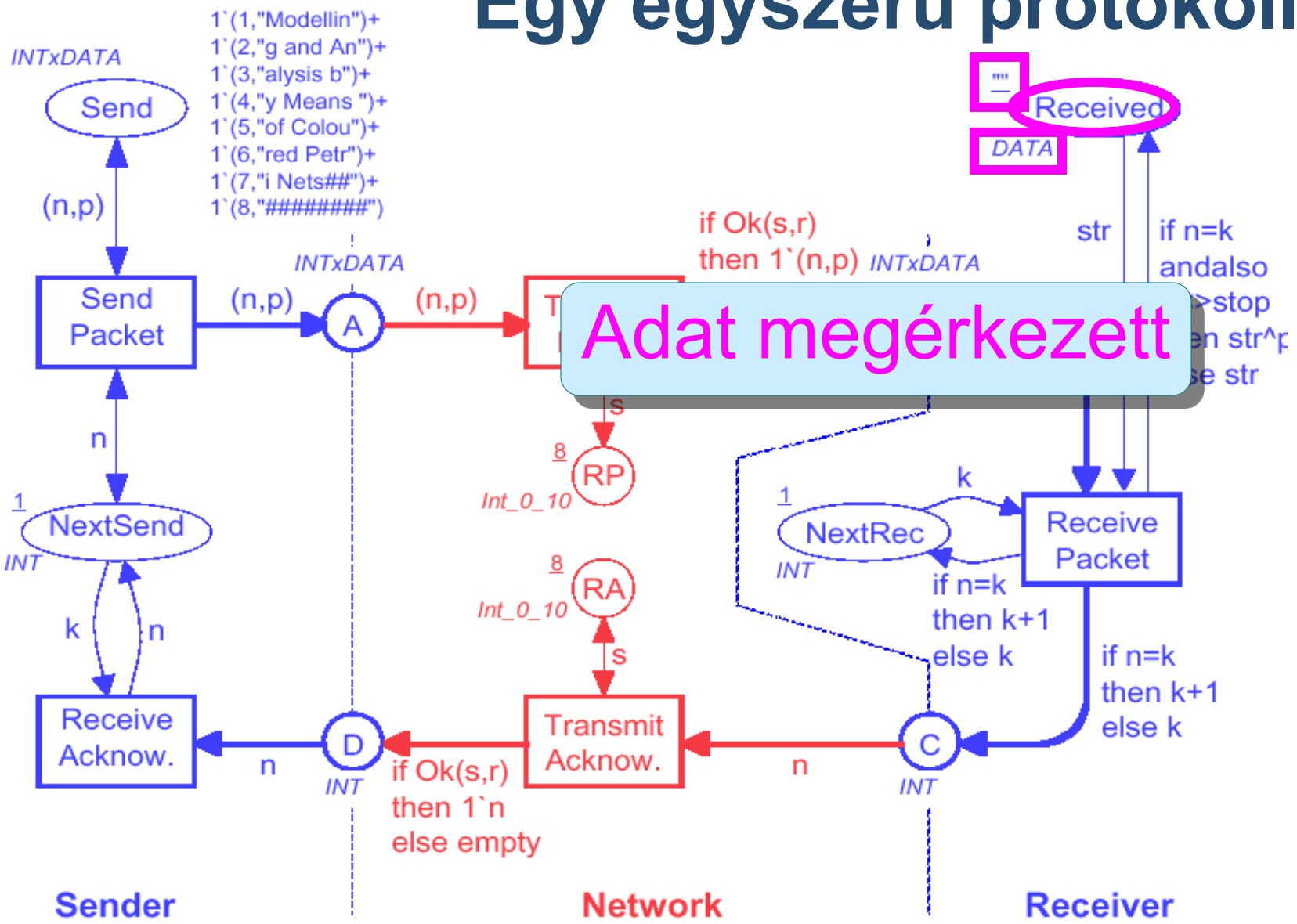
Egy egyszerű protokoll



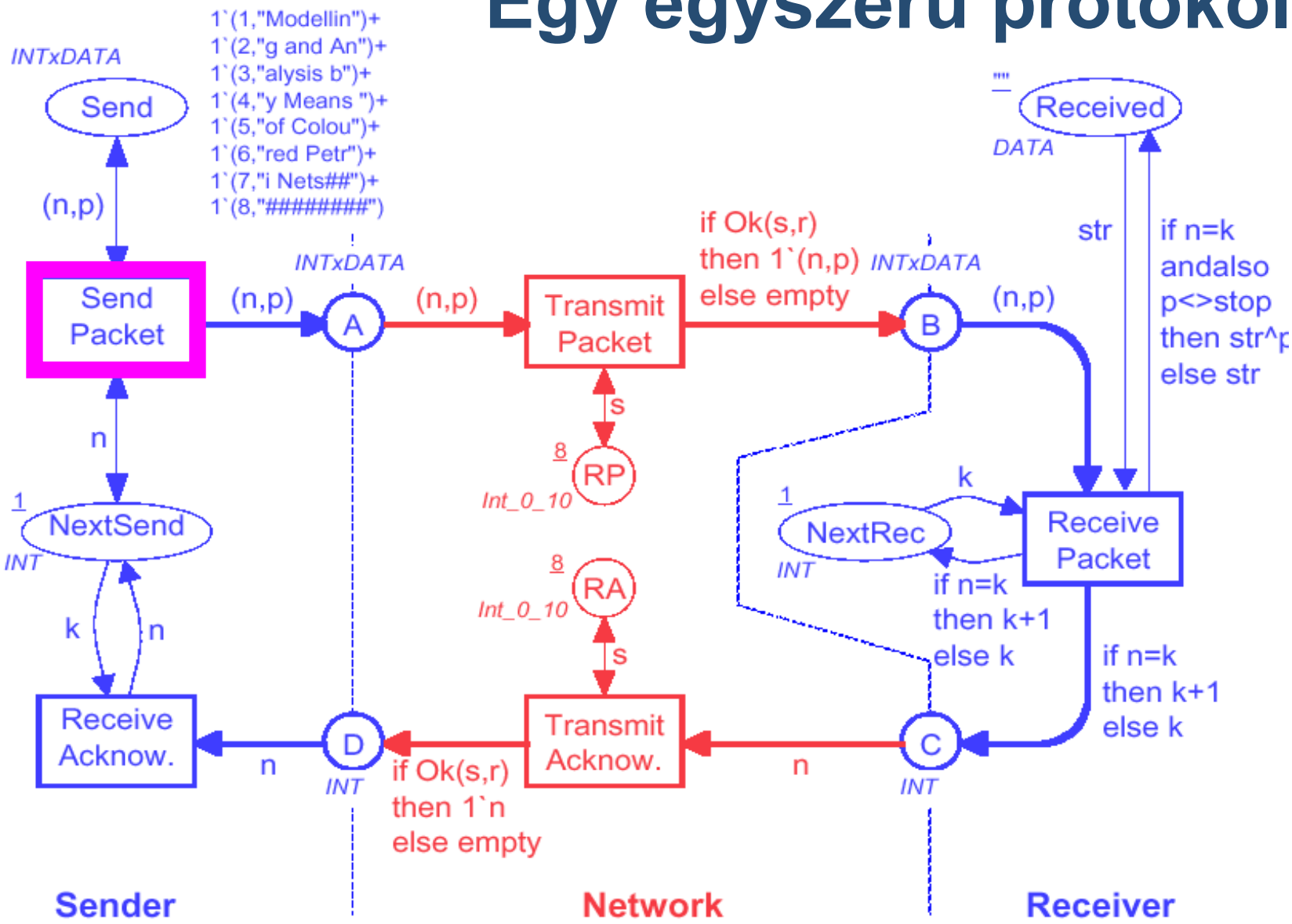
Egy egyszerű protokoll



Egy egyszerű protokoll



Egy egyszerű protokoll

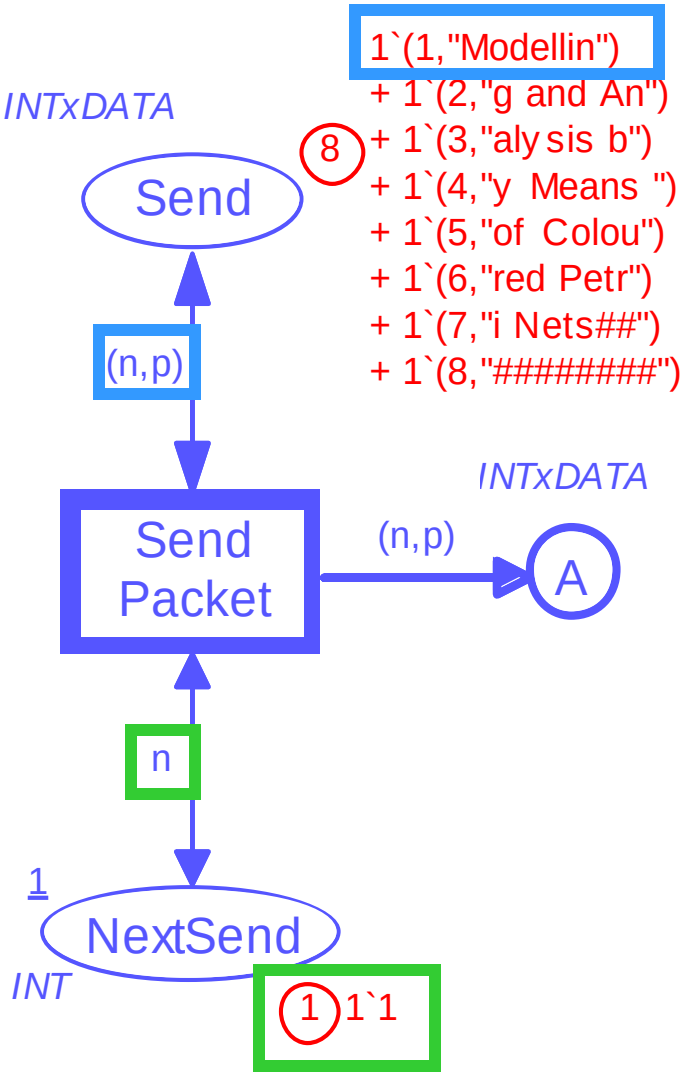


Csomagküldés

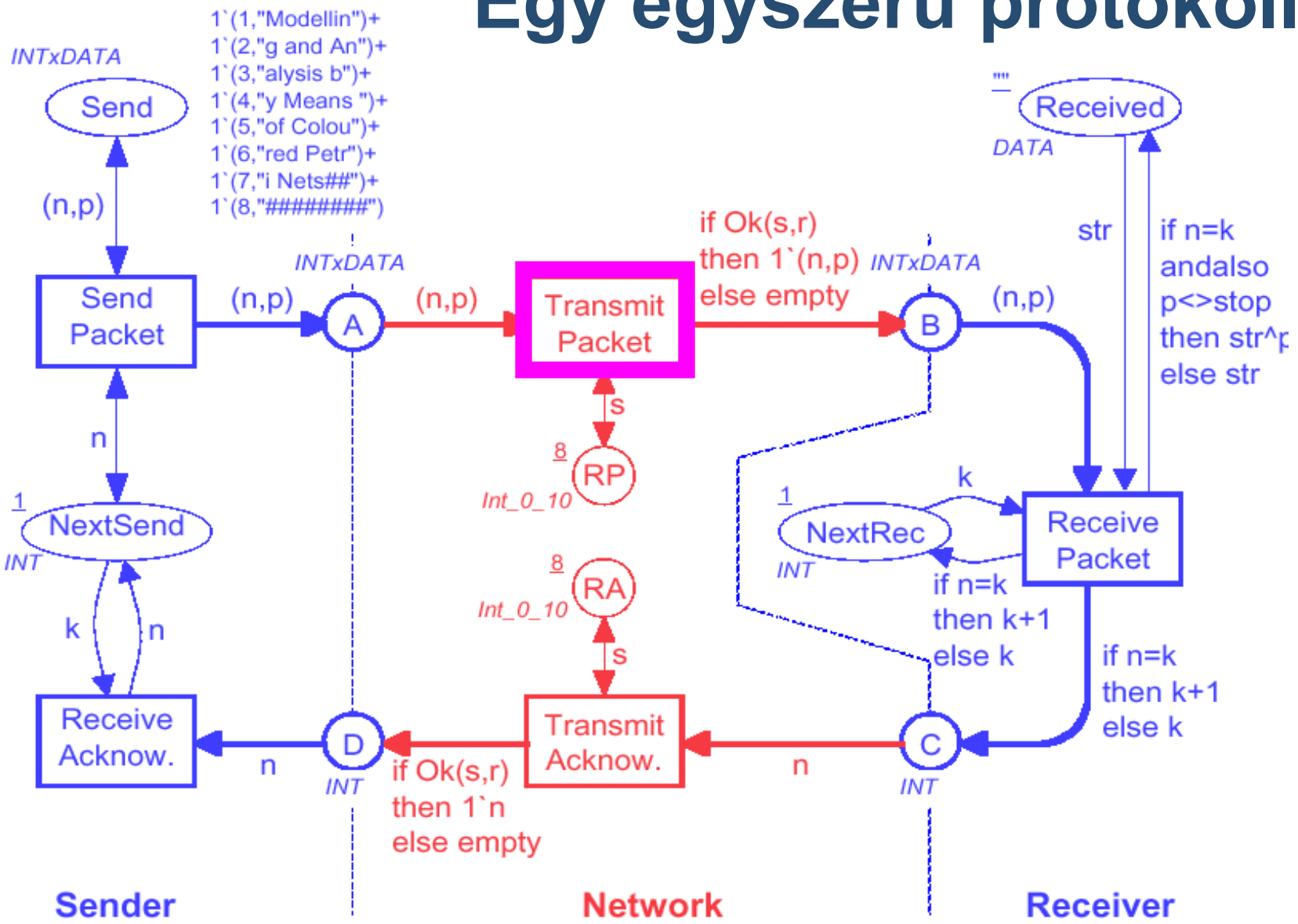
- Csak a

  <n=1,p="Modellin">

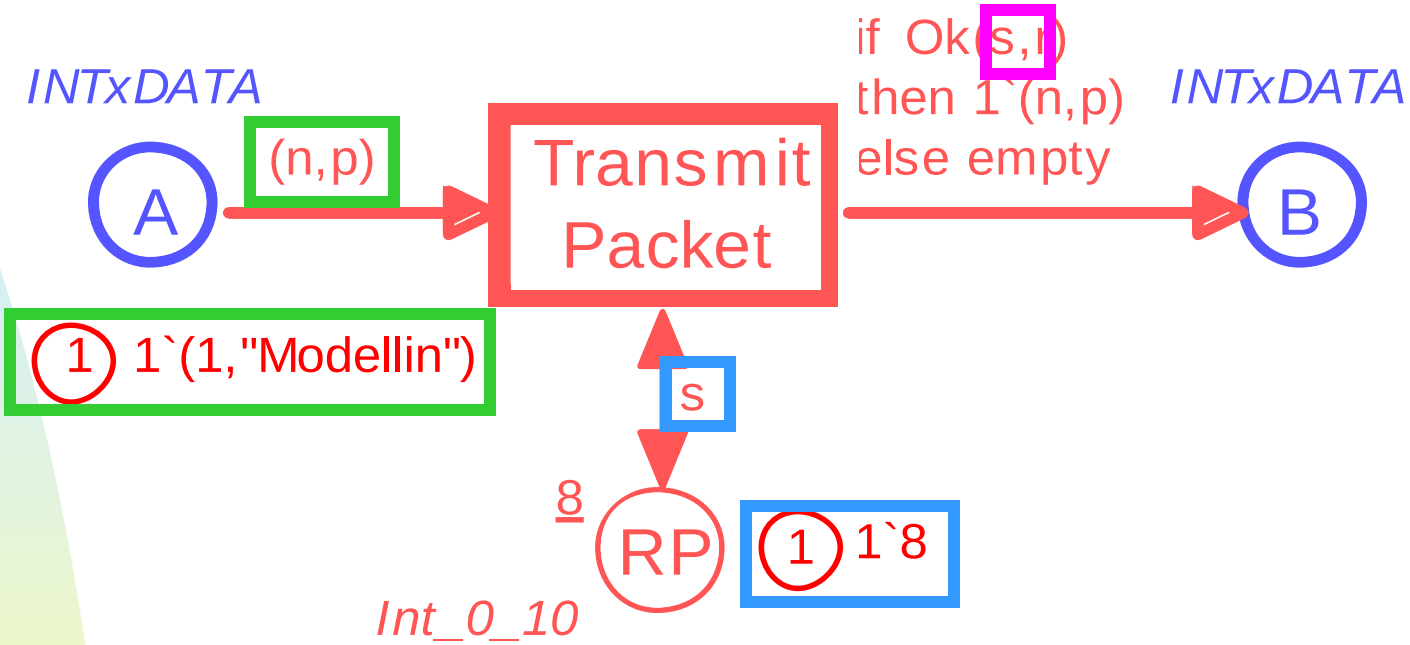
  pár *engedélyezett.*
- Ha található ilyen pár, akkor a *Send Packet* elküldi azt *A-nak*.
- Ez azt jelenti, hogy (1,"Modellin")-t elküldtük a hálózaton.
- A csomag nem törlődik a *Send* helyről és a *NextSend* sem változik.



Egy egyszerű protokoll



Csomag átvitele



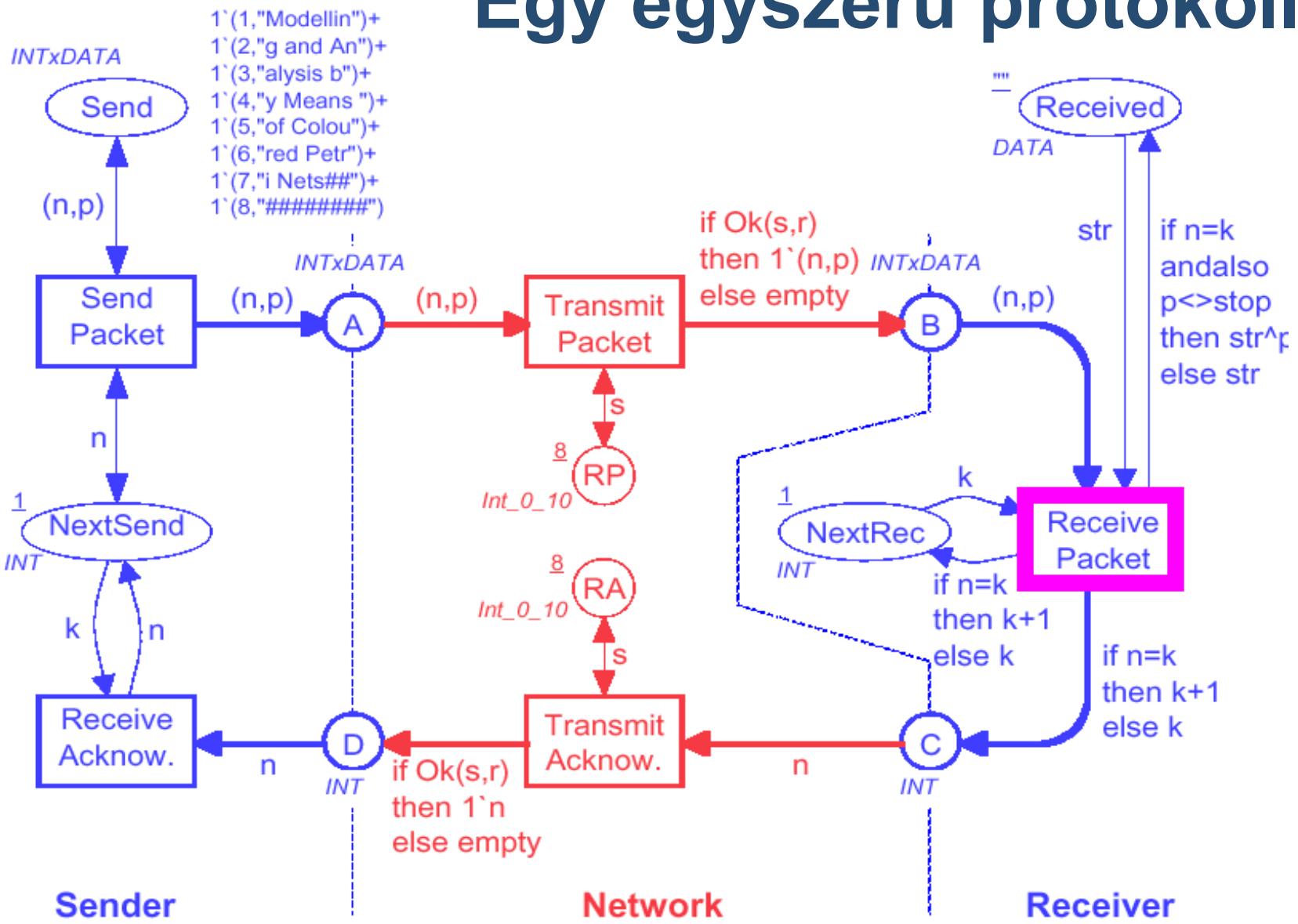
- Az engedélyezett pár így néz ki:
  - <n=1,p= "Modellin",s=8,r=...>
  - Típus: r ∈1. .10

Csomagvesztés

```
if Ok(s,r)
then 1`(n,p)
else empty
```

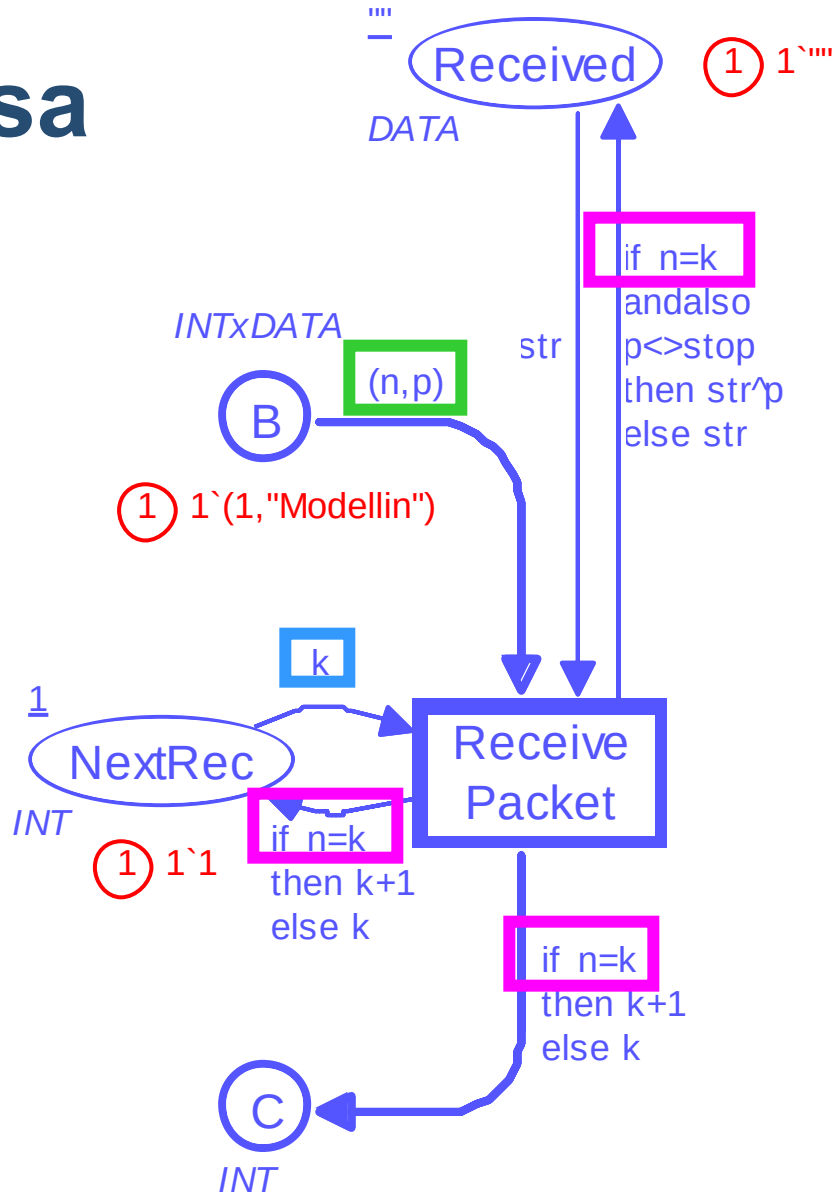
- A Ok(s,r) *függvény* ellenőrzi, hogy r ≤ s
  - For r ∈1. .8,*Ok(s,r)=true*.
    A tokent A-ból B-be tesszük. Ez azt jelenti, hogy a csomag *sikeresen átment* a hálózaton.
  - For r ∈ 9. .10,*Ok(s,r)=false.*
    Ekkor B-be nem került token. Ekkor *veszett el* a csomag.
- A CPN szimulátor *véletlenszerűen választja* ki a párokat: 80% eséllyel sikeresen.

Egy egyszerű protokoll



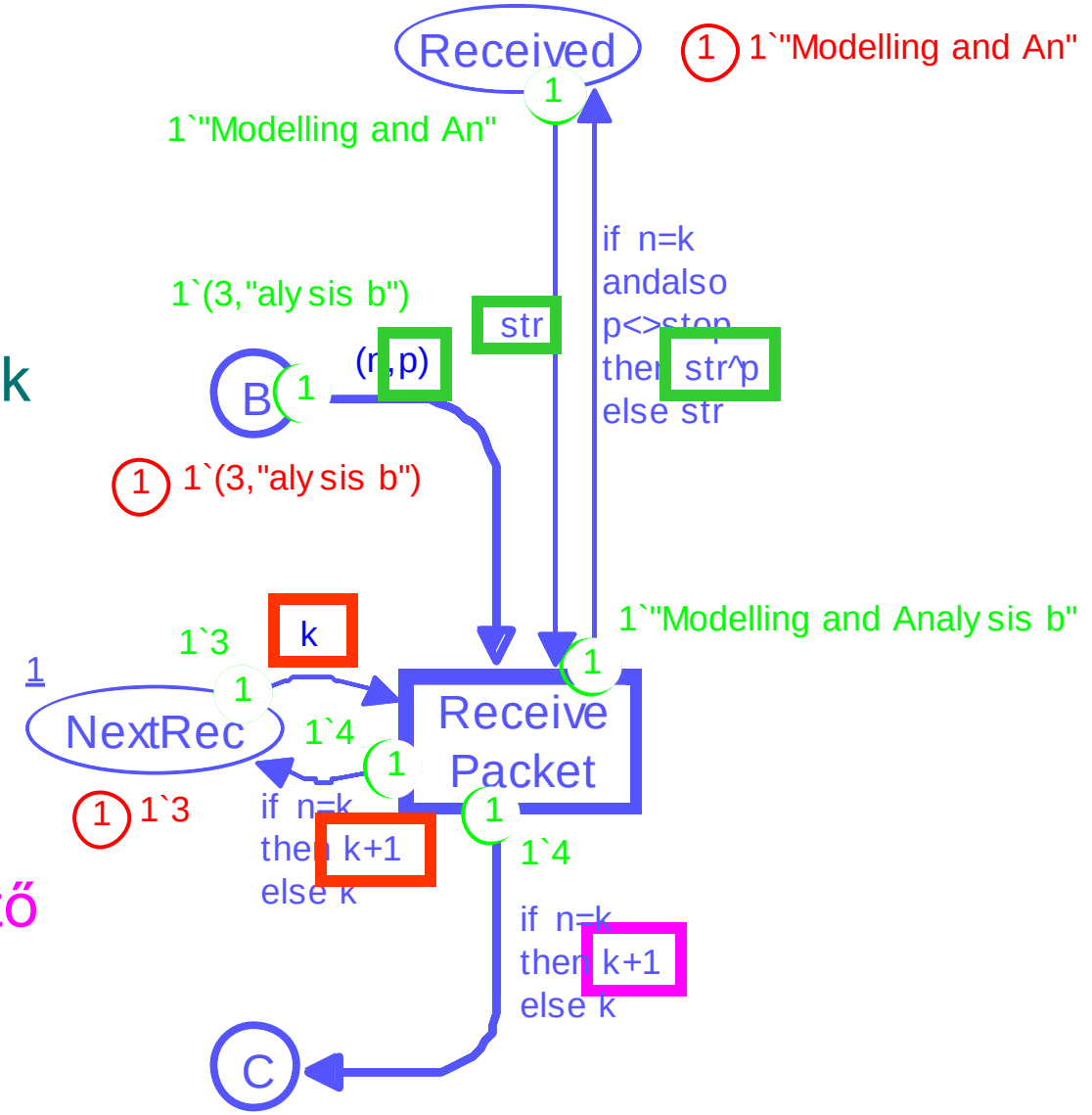
Csomag fogadása

- A *bejövő* (n) és a *várt* csomagok sorszámát (k) összehasonlítjuk.



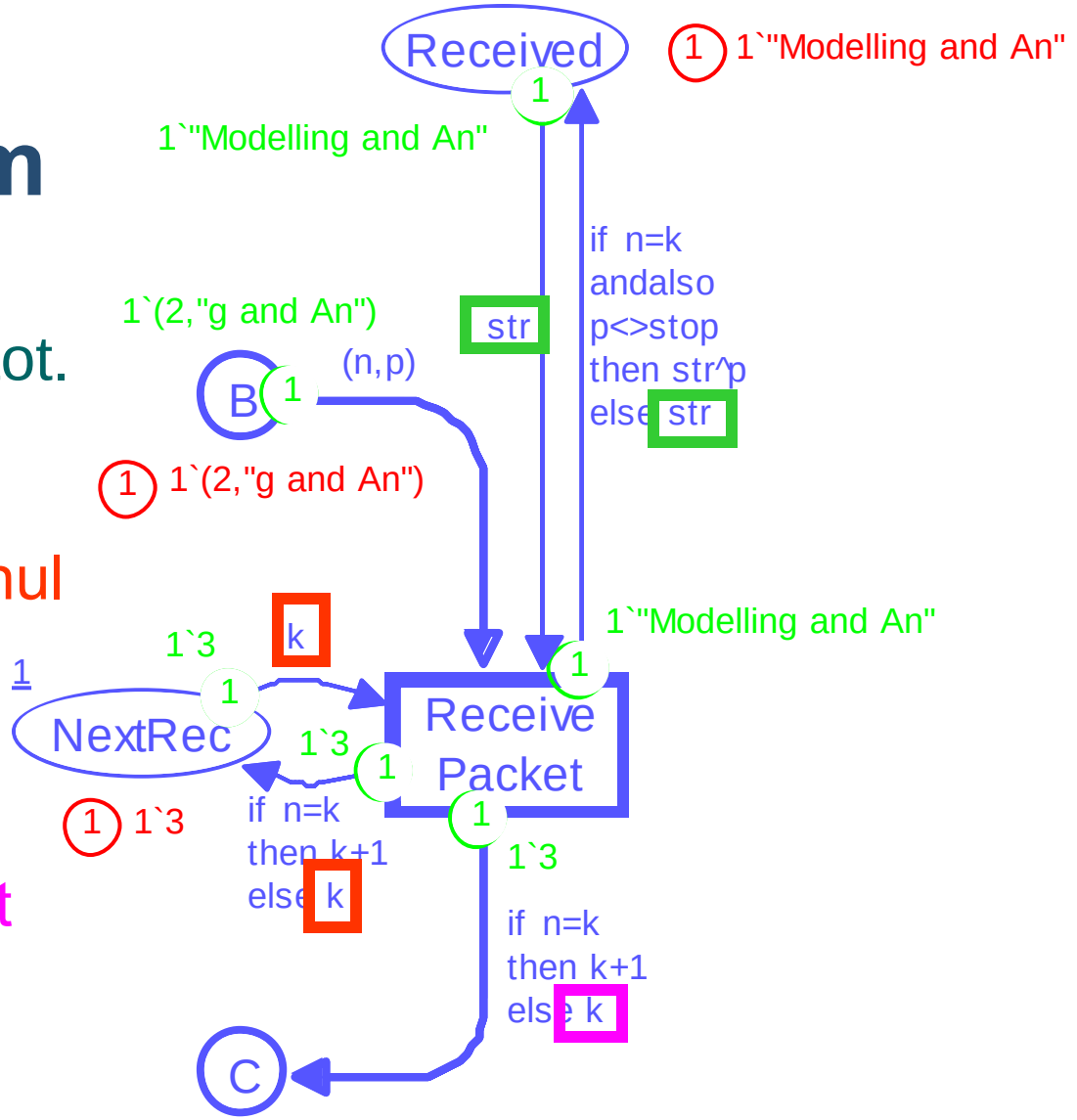
Csomagok sorszámának módosítása

- A már megkapott adatokhoz hozzáfűzzük az újonnan jöttet.
- A *NextRec* számlálót egyenként növeljük.
- *Acknowledgement üzenetet küldünk*. Mellékeljük a következő körben várt üzenet sorszámát.

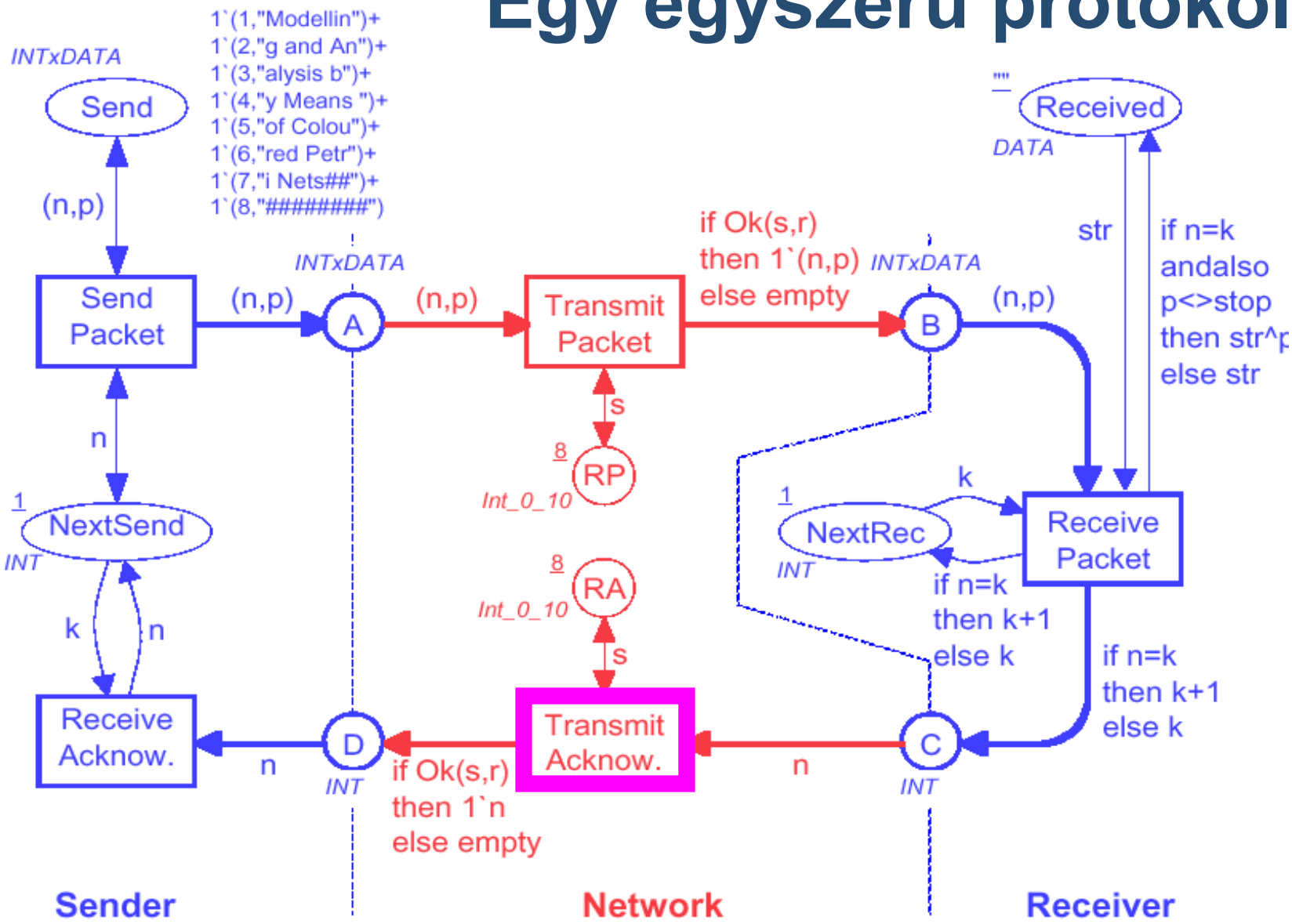


Rossz csomagsorszám

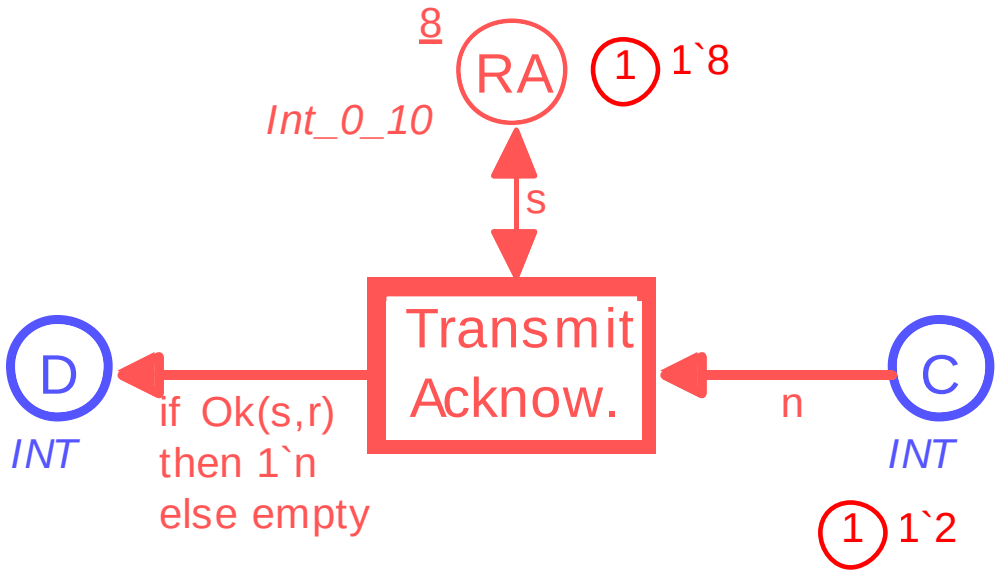
- *Eldobjuk* a kapott adatot.
- A *NextRec-t* változatlanul hagyjuk.
- Egy *acknowledgement üzenetet* küldünk. Ez a várt csomag sorszámát tartalmazza.



Egy egyszerű protokoll

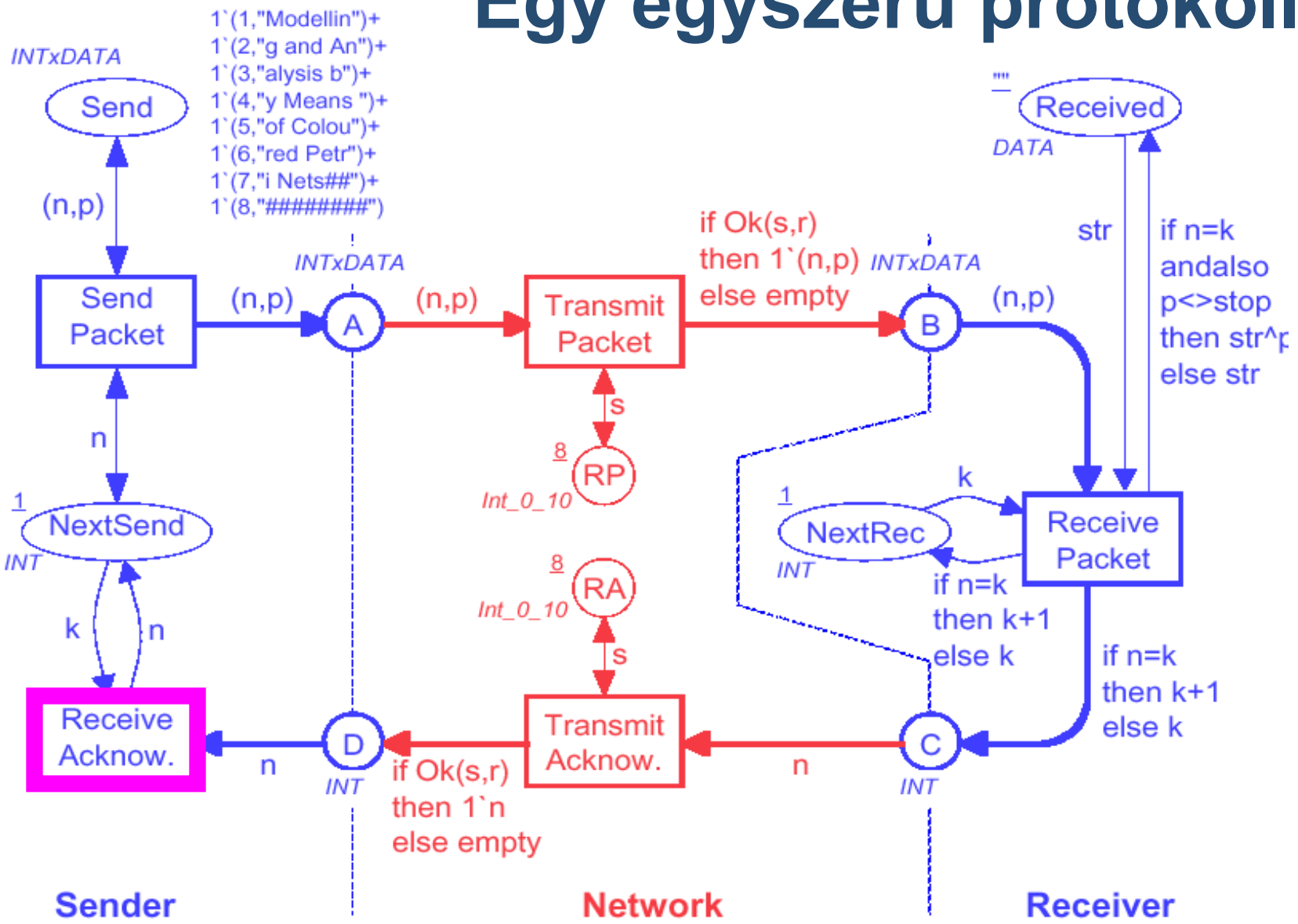


Átvitel visszajelzés

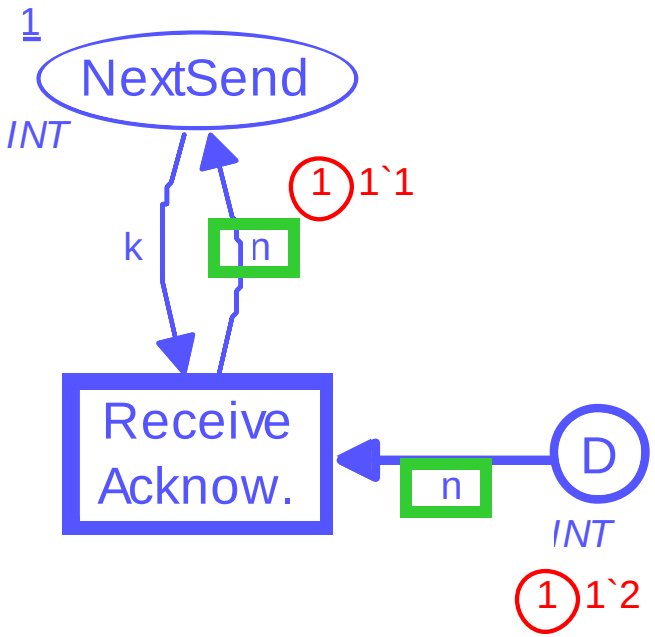


- Hasonlóan működik a *Csomag átvitelhez.*

Egy egyszerű protokoll



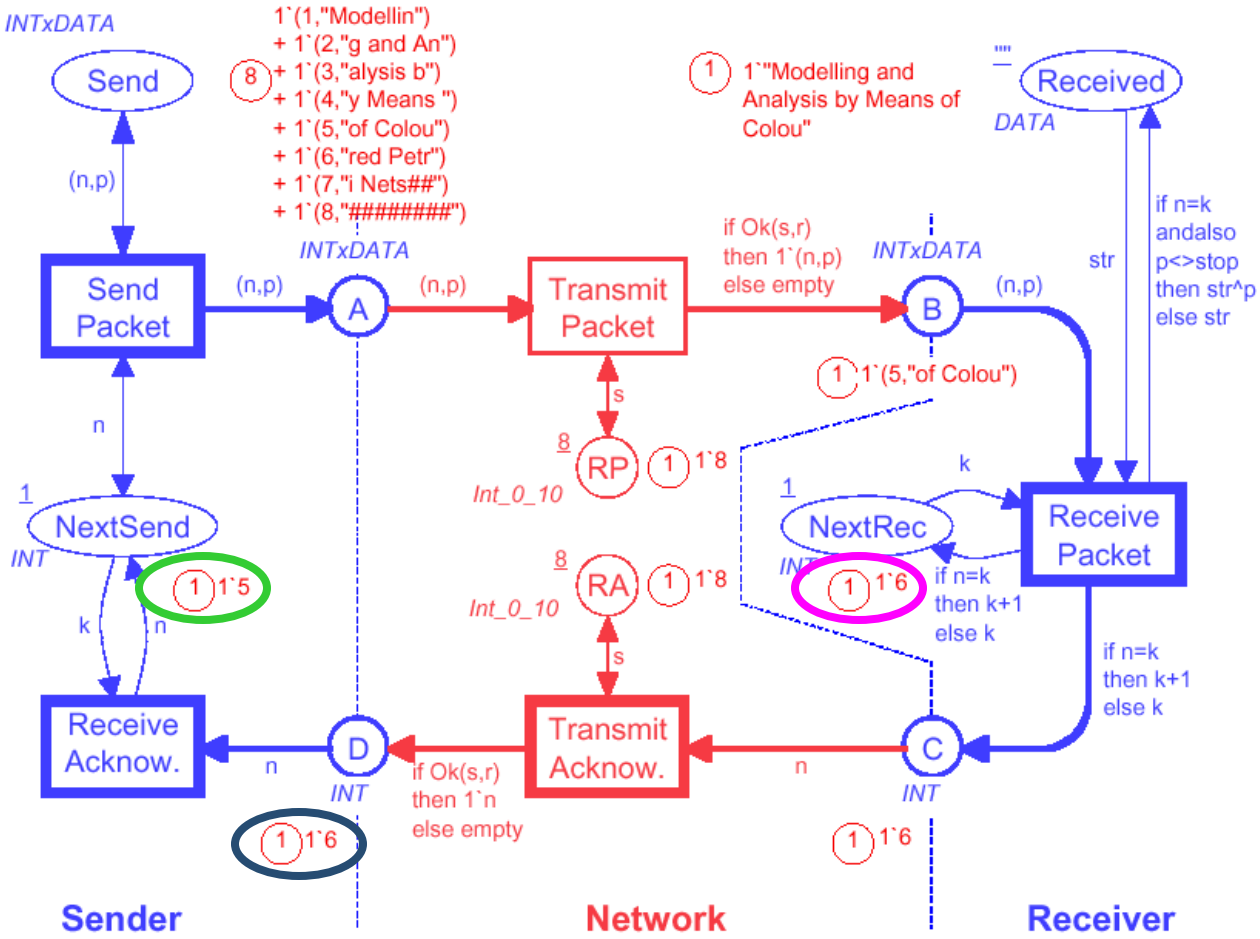
Fogadási visszajelzés



- Amikor az acknowledge megérkezik a *Küldőhöz*, az azonnal frissíti a *NextSend* számlálót.
- Ebben az esetben a számláló értéke 2 lesz, így legközelebb a Küldő a 2-es csomagot fogja küldeni.

Közbülső állapot

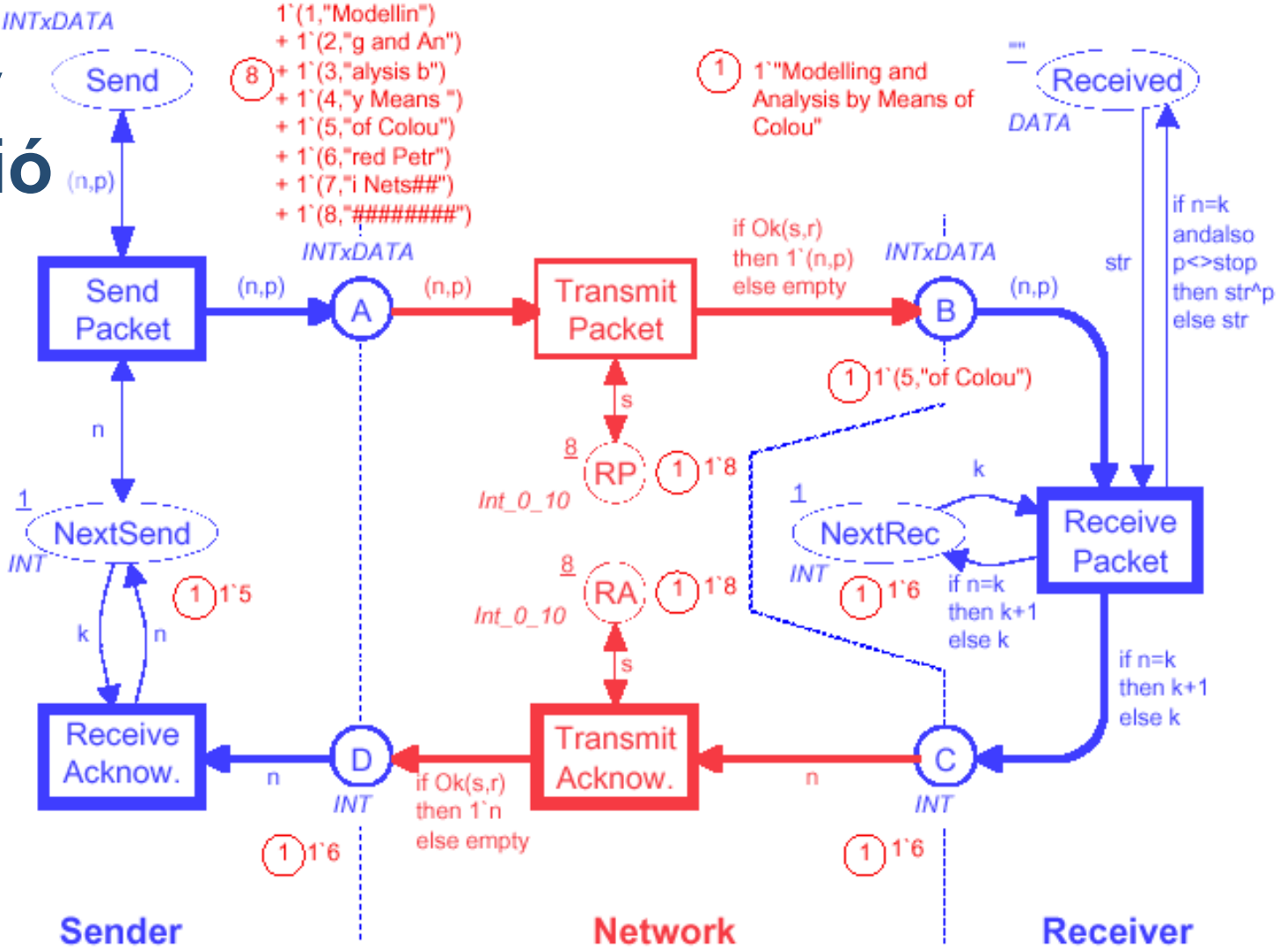
- *Fogadó* várja a 6-os csomagot.
- *Küldő* még csak az 5-ös csomagot küldi.
- A 6-os acknowledgement kérő csomag érkezik meg.
- Ezután a *NextSend* frissítődik majd a *Küldő* elindítja a 6-os csomagot.



Színezett Petri háló szimulációja

- A *szintaktikailag helyes* CPN ábrából a CPN tool *legenerálja* a szükséges *kódot.*
  - Kiszámítja, hogy az egyes tranzíciók *engedélyezettek-e.*
  - Majd *végrehajtja* a tranzíciókat.
- A szintaxis ellenőrzés és a kódgenerálás *inkrementális*. Emiatt a kis módosítások könnyen végrehajthatóak.
- Két különböző szimulációt különböztetünk meg:
  - *Interaktív* szimuláció: a felhasználó beavatkozhat, de nagyrészt a rendszer dolgozik.
  - *Automata* szimuláció: a rendszer mindent megcsinál.

Interaktív szimuláció



- *A Szimuláció* eredményét az ábrán láthatjuk.
  - *A következő tüzelést* manuálisan vagy automatikusan lehet kijelölni.
  - A felhasználó *watchpointot* és *breakpointokat* definiálhat.

Automatikus szimuláció

- Amennyiben *nem* kívánjuk nyomon követni magát a szimulációt:
  - ez *nagyon gyors lehet* – sokezer lépes/mp.
  - Lehet *megállási kritériumokat* definiálni
  - Megállásnál a *grafikus képernyő* frissül
  - Ekkor a kialakult helyzetet tanulmányozhatjuk
- Az automatikus és interaktív szimuláció *hibrid is lehet.*
- Az automatikus szimuláció eredményének megismeréséhez a felhasználó számára számos eszköz áll rendelkezésre.

Szimuláció report

```
1 SendPack@(1:Top#1) {n=1,p="Modellin"}
2 TranPack@(1:Top#1) {n=1,p="Modellin",r=6,s=8}
3 SendPack@(1:Top#1) {n=1,p="Modellin"}
4 TranPack@(1:Top#1) {n=1,p="Modellin",r=3,s=8}
5 RecPack@(1:Top#1)  {k=1,n=1,p="Modellin",str=
6 SendPack@(1:Top#1) {n=1,p="Modellin"}
7 SendPack@(1:Top#1) {n=1,p="Modellin"}
8 TranAck@(1:Top#1)  {n=2,r=2,s=8}
```