

MÉRNÖKI MODELLALKOTÁS AZ ELMÉLETTŐL A GYAKORLATIG

HÁLÓZATI KÓDOLÁS A JÖVŐ KOMMUNIKÁCIÓS HÁLÓZATAIBAN

Dr. Babarczi Péter
egyetemi adjunktus

BME Távközlési és Médiainformatikai Tanszék
MTA-BME Lendület Jövő Internet Kutatócsoport

2

Hatalmas sikert értek el a BME kutatói

Researchers find the organization of the human brain to be nearly ideal



nature.com > journal home > archive by date > july > full text

NATURE COMMUNICATIONS | ARTICLE OPEN

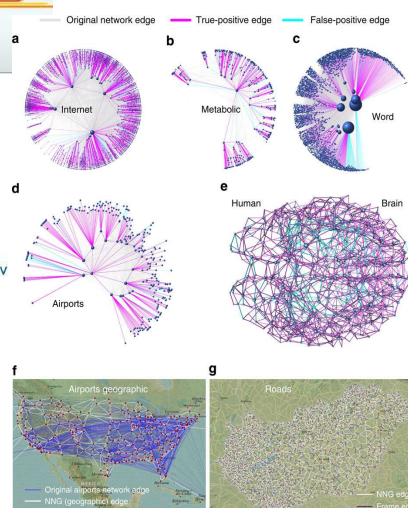
Navigable networks as Nash equilibria of navigation games

András Gulyás, József J. Biró, Attila Kőrösi, Gábor Rétvári & Dmitri Krioukov

Affiliations | Contributions | Corresponding authors

Nature Communications 6, Article number: 7651 | doi:10.1038/ncomms8651

Received 04 December 2014 | Accepted 26 May 2015 | Published 03 July 2015

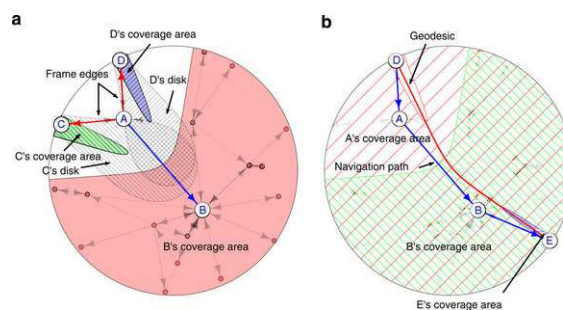


MI VÁRHATÓ A FÉLÉVBEN?

Dr. Gulyás András

Hálózatok kialakulásának vizsgálata játékelmélettel

- Valós hálózatok (mérnöki, biológiai, közösségi, gazdasági) tulajdonságainak áttekintése
- Történelmi jelentőségű (klasszikus) hálózatmodellek ismertetése (véletlen gráfok, növekedési modellek stb.)
- Játékelmélet alapú modellek alapjai (játékelmélet alapfogalmai, egyszerű példák, hálózatformációs játékok, navigációs játékok)



Csomagok továbbítása szűkös erőforrásokon

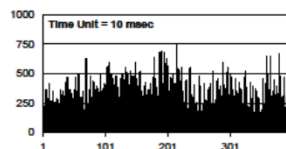
- Összeköttetések versengés helyett működjenek együtt a hálózati erőforrások kihasználásáért
 - Feladat: Maximalizáljuk a throughputot kommunikációs hálózatokban!
- Megfelelő leíróerejű modell alkalmazása
 - Ami évtizedekig nem ment gráfelmélet, az algebrailag megközelítve egy új kutatási területet indított 2000-ben
 - 2008 óta egyre több gyakorlati megvalósítás
- Az elmélettől a gyakorlatig...
 - Elméleti alapok, nehézségek, alapfogalmak
 - Megbízható hálózat tervezési követelmények
 - Egy egyszerű ötlettől egy megvalósult EU projektig...



Az Internet forgalmi tervezése



Erőforrások: sávszélesség és annak megosztása



Forgalmi igények:
 stacionárius érkezési
 folyamat
 (csomag, folyam, kapcsolat)

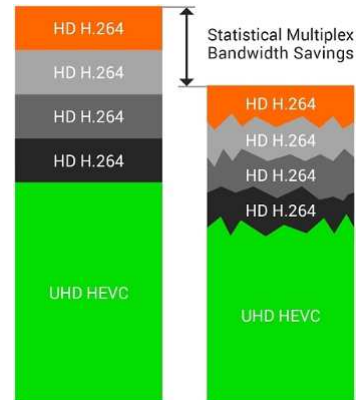
?

QoS, QoE

Teljesítmény:
 csomagvesztési
 arány, válaszadási
 idő, stb.

Statisztikus multiplexelés, sávszélesség becslés

- Mekkora sávszélesség szükséges kötegelt (multiplexelt) beszéd vagy videófolyamok esetén ?
- Tudunk-e megtakarítani sávszélességet a csúcsigények összegéhez képest ?
- Olyan eljárásokat mutatunk, amelyek kevés paraméter alapján adnak jó becsléseket !
- Nemcsak hálózattervezőknek, hanem biztosítási kockázatelemzőknek is hasznos!!

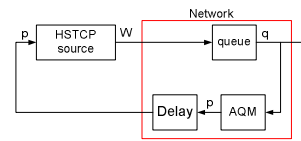


Bevezetés a dinamikus programozásba: IP forgalomtovábbítási táblák tömörítése

- **Dinamikus programozás:** komplex problémák megoldásának visszavezetése megfelelően megválasztott egyszerű részproblémák rekurzív megoldására
- Hasznos eszköz jól struktúrált feladatok megoldására
- A módszert az IP forgalomtovábbítási táblák tömörítésének gyakorlati feladatán keresztül szemléltetjük
 - Háttér
 - IP routing táblák minimalizálása
 - Forgalomtovábbítás szint-tömörített prefix fákkal
 - Gyakorlat: DP-k felírása, fa-bejárások, prefix fák tömörítése

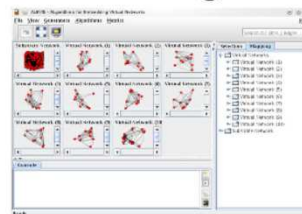
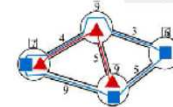
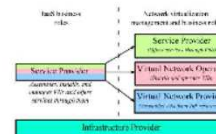
Forgalomszabályozás az Interneten

- Sok területen jön elő, egyik legfontosabb: TCP (Transmission Control Protocol)
- Probléma:
 - a küldő határozza meg az adási sebességet a vevő felé
 - sok küldő használja a közös hálózati erőforrásokat (linkek, routerek, switch-ek, bufferek, ...)
 - ki milyen sebességgel adjon, hogy "optimálisan" használjuk a hálózatot?
- Egy megoldás: TCP
 - első verzió – 1974
 - "congestion collapse" – 1986 -> torlódásvezérlés, implementáció (BSD)
 - akkori környezetben meglepően jó működés
 - de miért lett ilyen jó?
 - "reverse engineering" – '90-es évek -> matematikai modellek
- folytonos idejű visszacsatolt rendszer
 - folyadékmódel
 - szabályozástechnika
 - stabilitásvizsgálat
 - ...
$$\dot{W}(t) = \frac{a(W(t))}{R(t)} - b(W(t)) W(t) \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t))$$



Virtual Network Embedding Virtuális hálózat beágyazás

- VNE feladat áttekintése
 - a feladat alkalmazási területei a gyakorlatban
 - a probléma formalizálása, paraméterek, mérőszámok, célok
 - megoldási módszerek, algoritmusok áttekintése
- Egy kiválasztott megoldási módszer részletei
- Szimulációs eszköz (gyakorlat)
 - VNE algoritmusok kiértékelése



Források:

- A Fischer, J.F. Botero, M.T. Beck, H. De Meer, X. Hesselbach, *Virtual Network Embedding: A Survey*, IEEE Communications Surveys & Tutorials, 2013.
- <http://sourceforge.net/projects/alevin/>

Tárgykövetelmények – Aláírás, vizsga

- Aláírás:
 - Előadások
 - ajánlott ☺ részvétel, 7 külön tématerületbe kaphattok betekintést
 - **ZH nincs**
 - Gyakorlatok:
 - félév folyamán 6 gyakorlat
 - **gyakorlatok min. 70%-án kötelező a részvétel** (min. 5 gyakorlat)
 - gyakorlatokra aki tud, lappal érkezzen!
 - Octave telepítve: <https://www.gnu.org/software/octave/>
 - **Házi feladatok „elégéses” szintű teljesítése**
- Vizsga jegy:
 - Írásbeli vizsga (Nagyon nehéz! De tényleg, látni fogjátok!)
 - **Házi feladatok alapján megajánlott jegy szerezhető!**

Tárgykövetelmények – Házi feladat

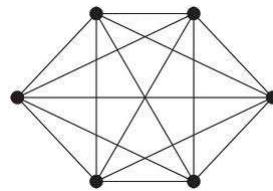
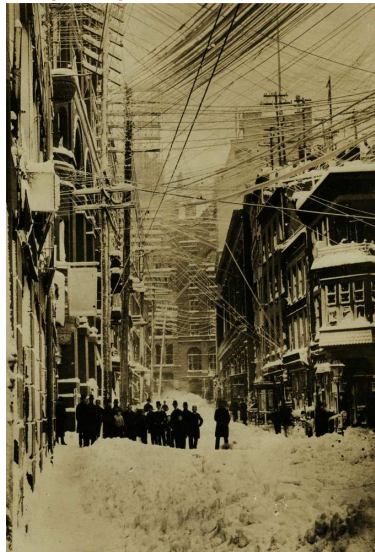
- Három házi blokk (2-2-3 téma anyagát lefedve)
 - **1. házi:** kiadás: szeptember 24-i gyak (3. hét), beadás: október 13-i előadás (6. hét)
 - **2. házi:** kiadás: október 22-i gyak (7. hét), beadás: november 10-i előadás (10. hét)
 - **3. házi:** kiadás: november 19-i gyak (11. hét), beadás: december 8-i előadás (14. hét)
 - Mind a 7 téma anyagából 2 kis feladat (7x2x10 pont = 140 pont)
 - Az első 6 téma anyagából 1 nagy feladat (6x30 pont = 180 pont)
 - **Megajánlott jegy:** Max. 320 pont, **jeles (5)** >= 260 pont, **jó (4)** >= 230 pont
 - **Aláíráshoz: min.10 pont mind a 7 téma anyagából külön-külön!**
 - Házik pótlása: Aláíráshoz hiányzó házik (és csak azok!) pótlási héten **személyesen** pótolhatók

COMPUTE-AND-FORWARD ELV

Hálózati evolúció / revolúció a telefontól az adatközpontokig

Telefonhálózat evolúciója (1870-es évek)

Kezdetben pont-pont összeköttetések: teljes gráf



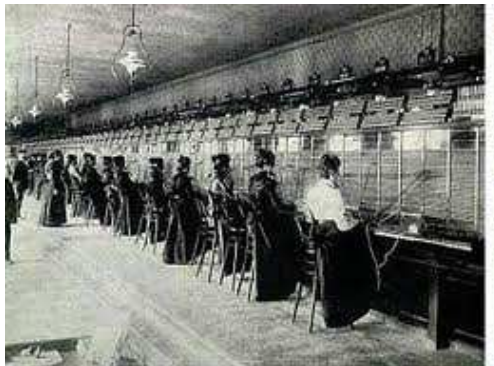
Telefonhálózat evolúciója (1870-es évek)

Később mindenki egy központba bekötve – csillag topológia



Telefonhálózat evolúciója (1870-es évek)

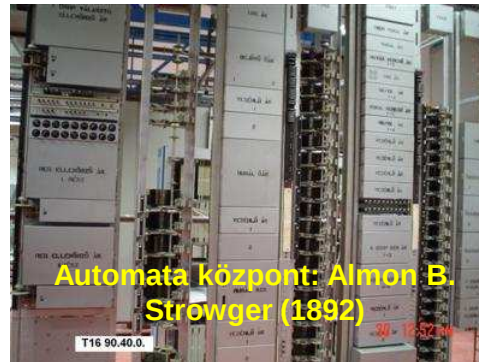
Később központok hierarchiája – szövevényes hálózati struktúra



- Alapelv: **áramkörkapcsolás**
 - „fémes kapcsolat” a hívó és a hívott között
 - beszéd forgalomra optimalizálva
 - <https://www.youtube.com/watch?v=qaJYWrYKVRo>

Telefonhálózat evolúciója (1890-es évektől)

Automatizálás, Digitális technológia – közös platform



- Rendszerek világméretű együttműködése
- Fizikai áramkörök helyett virtuális áramkörök megjelenése

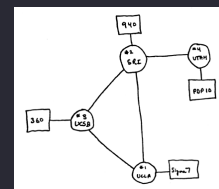
Forradalom (revolution, 1960-as évek)

Áramkörkapcsolás - csomagkapcsolás

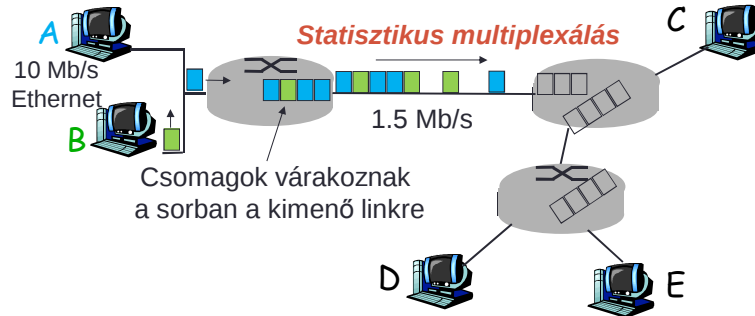
- **Megbontjuk az áramkör koncepciót -> csomagok**
 - 1961: Kleinrock
 - Első publikáció csomagkapcsolt hálózatokról, sorbanállás-elmélet
 - 1969: első távoli hozzáférés, az Internet őskének születése

"We set up a telephone connection between us and the guys at **SRI**...", Kleinrock ... said in an interview: "We typed the **L** and we asked on the phone, "Do you see the **L**?"
 "Yes, we see the **L**," came the response.
 "We typed the **O**, and we asked, "Do you see the **O**."
 "Yes, we see the **O**."
 "Then we typed the **G**,
 and the **system crashed**"...
 Yet a **revolution had begun**"...

Source: *Sacramento Bee*, May 1, 1996, p.D1



Store-and-forward



- Az **A** és **B** gépek által küldött csomagoknak nincs meghatározott sorrendje, igény szerint osztoznak az erőforráson = **statisztikus multiplexálás**.
- **Store-and-forward elv**
 - A routerek várakozási soraiban az adatok várakoznak, amíg a csatorna szabad

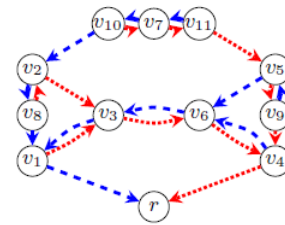
TCP/IP protokoll megjelenése

[illegible]

Internet evolúciója (Napjainkban, 2010-es évek)

Multicast és multipath előretörése

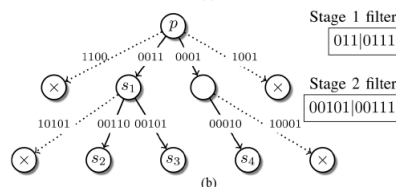
- Az áramkör koncepció felbontása előnyös
 - **Multicast** (többesadás): egy forrás több nyelő
 - Video forgalom ma már a domináns (pl. IPTV)
 - **Multipath Internet**: egyetlen út helyett akár több út is rendelkezésére áll a forrásnak
- Multipath Internet előnyei:
 - Terhelés kiegyenlítés
 - IP rétegben is törekvések szabványosításra (MRT)
 - SDN megjelenése felgyorsította ezt a folyamatot
 - Magasabb megbízhatóság
 - Forgalom adaptálódik a hálózati viszonyokhoz
 - A forrás képesek a hibát kikerülni
 - Biztonság (lehallgatás ellen)
 - Throughput
 - Multipath TCP (MPTCP szállítási rétegben)



Forradalom (revolution, 2020-as évek)

Új architektúrákhoz új szemlélet – compute-and-forward

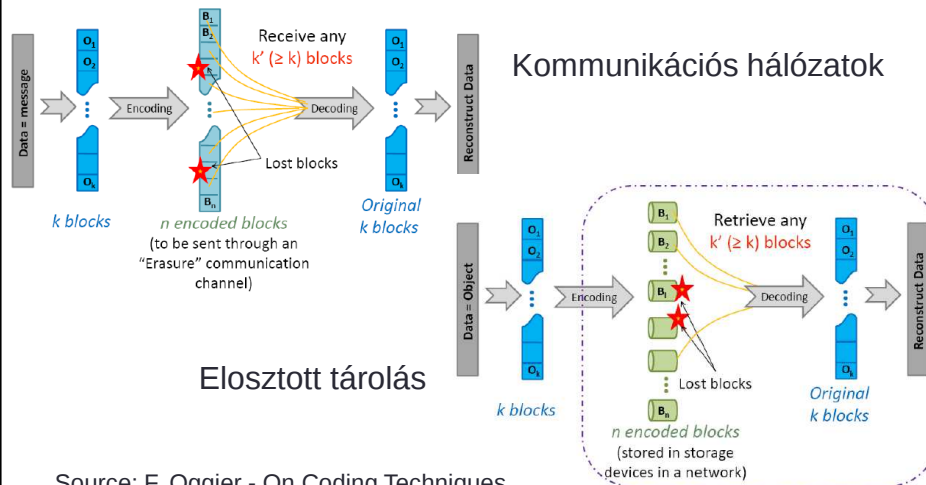
- Példa: Tartalom/információ centrikus hálózatok (Content/Information Centric Networks, CDN/ICN)
- A kommunikáció sokkal inkább arra irányul, hogy **milyen** információra van szükség ahelyett, hogy az információ **hol** található
 - Publish/subscribe architektúra (**randevú, topológia kialakítása, továbbítás**) alapvetően **multicast alapú továbbítást** valósít meg
 - IP helyett új fajta címzési architektúrák jelennek meg



- Ha megint változik az architektúra, akkor ismét lehetőség a csomag koncepció megbontására: **csomagok helyett kódolt információ**

Forradalom (revolution, 2020-as évek)

Az adatok továbbítása (routerek) és tárolása (distributed storage) logikailag nem válik el egymástól

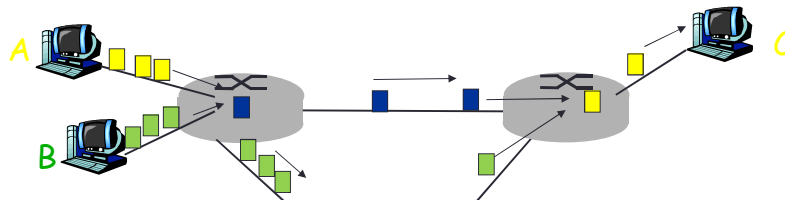


Source: F. Oggier - On Coding Techniques for Networked Distributed Storage Systems

Milyen előnyei lehetnek a kódolásnak?

Compute-and-forward (a korábbi store-and-forward helyett)

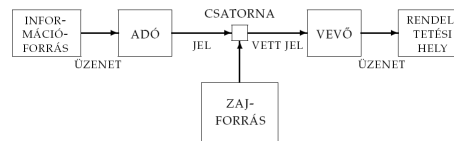
- Throughput
 - Több információ átvitele kevesebb csomaggal/rövidebb idő alatt
 - Köv.: **kisebb késleltetés**, vezeték nélküli hálózatokban kevesebb energia
- Ellenállóság
 - Link hibák ellen
 - Csomagvesztés esetén
- Komplexitás
 - A csomag/folyam koncepció megbontása sokkal hatékonyabb algoritmusok alkalmazását teszi lehetővé (pl. kapacitás foglalásra)
- Biztonság
 - Egy-egy út lehallgatásával a teljes üzenet nem állítható helyre
 - Tárolás esetén adatunkat megosztjuk több felhő között



De mit is kódoljunk és hogyan?

Más koncepció, mint amiket eddig ismertünk

- Csatorna kódolás
 - Zajos csatornán üzenetet átküldeni
 - „**redundanciát adunk az adathoz**” (pl.: Reed-Solomon kódok)
 - erasure coding: k szimbólumból álló üzenetet $n > k$ szimbólumba alakítok úgy, az üzenet bármely k darabból helyreállítható
- Forrás kódolás
 - Az információforrás üzeneteit gazdaságosan, tömören reprezentáljuk
 - „**eltávolítjuk a redundanciát**” (pl.: Huffman kódok, más prefix kódok)
- GOND: Csatorna- és forrás kódolás mind-mind *end-to-end*
 - A hálózathoz (pl.: eltörlődéses hibák, link meghibásodások) való adaptációt a végpontok végzik (nem tudok „re-kódolni” **a hálózat belsejében** anélkül, hogy **dekódolnom kellene az eredeti üzenetet**)
 - Dekódoláshoz bufferelnem kell, meg kell várnom a szükséges csomagokat...
 - Nehézséges, lassú, szakítani kell ezzel a koncepcióval.
 - A gyors („tactile”) 5G Interneten ez az extra késleltetés nem megengedhető!



Hálózati kódolás – Network coding

A küldendő információ egy másik reprezentációja

- Csomagokat **a hálózat belsejében** nyugodtan újra kódolhatom egymással dekódolás nélkül!
 - Lehetőség szerint minden csomag tartalmazzon új információt a nyelőknek (**Coupon collector problem**)
- Feladat: hogyan töltsen ki a kódoló mátrixot?

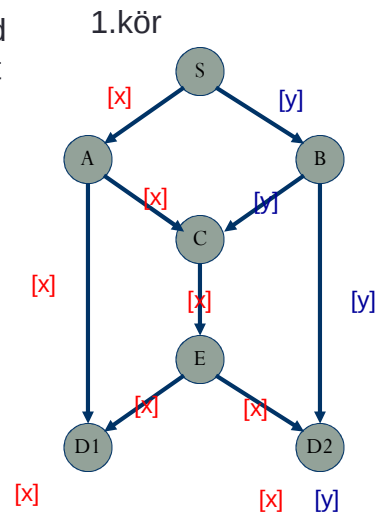
$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}$$

Kódolt információ Kódoló együtthatók (később: **transzfer mátrix**) Eredeti információ (pl.: bit, bájt, csomag)

Példa 1 („Butterfly network”)

Üzenetküldés multicast estén (Cél: max throughput elérése)

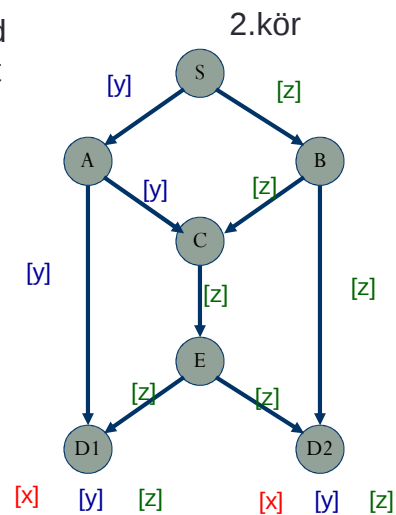
- Hagyományos store and forward koncepció (throughput: 3 üzenet / 2 körben = 1.5)



Példa 1 („Butterfly network”)

Üzenetküldés multicast estén (Cél: max throughput elérése)

- Hagyományos store and forward koncepció (throughput: 3 üzenet / 2 körben = 1.5)



Példa 1 („Butterfly network”)

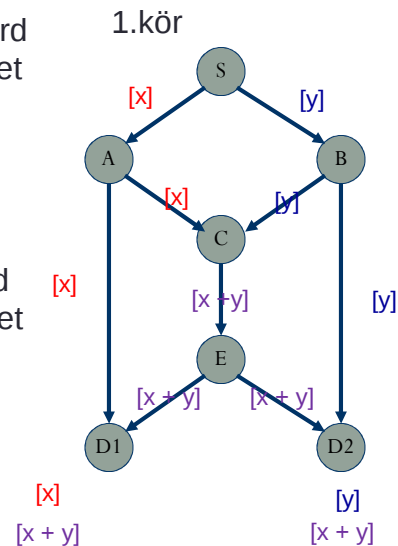
Üzenetküldés multicast estén (Cél: max throughput elérése)

- Hagyományos store and forward koncepció (throughput: 3 üzenet / 2 körben = 1.5)

- Compute (encode) and forward koncepció (throughput: 2 üzenet / 1 körben = 2)

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}$$

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Hálózati kódolás alaptétele

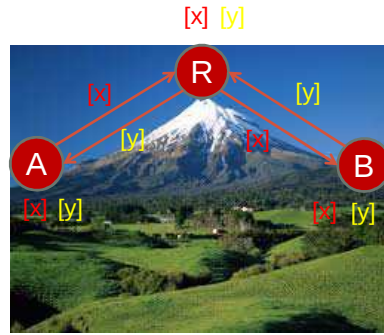
Multicast esetén megvalósítható maximális adatsebesség

- **Tétel (multicast):** Az s forrás és a $t_1, t_2, \dots, t_L$ nyelők között megvalósítható adatsebesség maximális értéke megegyezik az $s - t_i$ unicast összeköttetések esetén elérhető maximális adatsebességek minimumával.
 - Az előző példában 2-2 volt mindkét nyelőre a maximális folyam értéke, melyet meg is tudtunk valósítani.
 - Ennél jobbat nem is várhatunk, hiszen külön-külön sem tudnánk többet küldeni.
- Azaz hálózati kódolás esetén az egyes folyamatok **együttműködnek egymással** a szükséges erőforrások gazdaságos kihasználásáért
 - Kódolás nélkül a **felhasználók** versenyeztek (folyadék modell)
 - **Network Information Flow** (hálózat által elérhető adatsebesség)

Példa 2 („Wireless butterfly”)

Üzenetküldés vezeték nélküli relay estén (**broadcast csatornát kihasználva**)

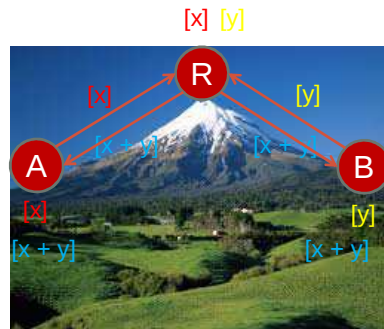
- Hagyományos store and forward koncepció (4 üzenet)
 - A küld üzenetet R-nek $[x]$
 - B küld üzenetet R-nek $[y]$
 - R továbbítja A üzenetét B-nek $[x]$
 - R továbbítja B üzenetét A-nak $[y]$



Példa 2 („Wireless butterfly”)

Üzenetküldés vezeték nélküli relay estén (**broadcast csatornát kihasználva**)

- Hagyományos store and forward koncepció (4 üzenet)
 - A küld üzenetet R-nek $[x]$
 - B küld üzenetet R-nek $[y]$
 - R továbbítja A üzenetét B-nek $[x]$
 - R továbbítja B üzenetét A-nak $[y]$
- Compute (encode) and forward koncepció (3 üzenet)
 - A küld üzenetet R-nek $[x]$
 - B küld üzenetet R-nek $[y]$
 - **R broadcastolja A és B üzenetének kombinációját $[x+y]$**



$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,4} \\ \alpha_{1,3} & \alpha_{2,4} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix}$$

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Hálózati kódolás alkalmazási területei

- Wireless networks
 - Broadcast, MAC további előnyöket biztosít (LTE, WiFi, stb.)
- 5G hálózatok
 - Minimális késleltetés biztosítása random kódolással (edge cloudok)
- Peer-to-peer
 - Média folyamatok hatékonyabb továbbítása (IETF WebRTC)
 - Microsoft Avalanche (BitTorrent-hez hasonlóan, csak NC-vel)
- Distributed storage systems (hasonlóan erasure coding-hoz)
 - Duplikálás ($n = 2$) sok erőforrást igényel, hogyan tudjuk a legkevesebb redundanciával tárolni az adatot (M/k) a felhőben n szerveren, hogy
 - Bármely k szerver ($k \ll n$) segítségével helyreállítható
 - Bármely k szerver adatából helyettesíthető egy kiesett szerver
- Multipath Internet
 - Resilience in Software Defined Networks- következő előadásban látni fogjuk (**MINERVA projekt**)

HÁLÓZATI KÓDOLÁS

Algebrai reprezentáció

Lineáris kódok

2000: A kezdetek...

Gráfelméleti megközelítés

- Ahlswede et. al.
 - A legendás „butterfly network” pillangó hálózat megálmodása
 - Hálózati kódolás alaptételének bizonyítása
 - Max-flow min-cut theoremen, azaz **gráfelméleti módszerek**en alapul
 - Túl bonyolult volt ahhoz, hogy sokan elkezdjenek vele foglalkozni
- Aztán elérkezett 2003...
 - R. Koetter and M. Médard. *An algebraic approach to network coding*. IEEE/ACM Transactions on Networking, 11(5):782–795, 2003
 - Megmutatják, hogy a hálózati kódolás feladata valójában ekvivalens egy mátrix megfelelő kitöltésével
 - a hálózati kódolás **algebrai megfogalmazása**

Egy kis algebra...

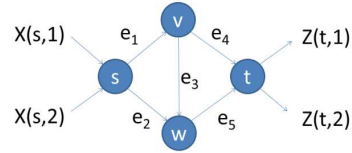
Műveletek véges testek felett

- $GF(p)$ - p elemű véges test (Galois Field)
 - Test axiómák (+ -re és $\times$ -ra nézve): nullelem, egységelem, asszociativitás, kommutativitás, disztributivitás
- Ha p prím, akkor az elemek **modulo műveletekre** testet alkotnak, pl.: $p = 2$ esetén 0,1 a két elem, a műveletek (XOR, kizáró vagy):

$0 + 0 = 0 \bmod 2$	$0 \times 0 = 0 \bmod 2$
$0 + 1 = 1 \bmod 2$	$0 \times 1 = 0 \bmod 2$
$1 + 0 = 1 \bmod 2$	$1 \times 0 = 0 \bmod 2$
$1 + 1 = 0 \bmod 2$	$1 \times 1 = 1 \bmod 2$
- Ha p prím hatvány, pl.: $p = 4 = 2^2$, akkor testelemek 0,1,2,3, a műveletek pedig **polinom aritmetika** szerint működnek
 - Egy irreducibilis polinom, pl.: $a(y) = y^2 + y + 1$ maradékosztályaként előállnak a testelemek $\{[0], [1], [y], [y + 1]\}$, azaz binárisan 00, 01, 10, 11):
 - $[y] + [y + 1] = [1]$ (helyiértékenkénti $\bmod 2$ összeadás)
 - $[y] \times [y] = [y + 1]$, mert $y^2 \bmod (y^2 + y + 1) = [y + 1] = 11$ (binárisan) = 3

Hálózati kódolás

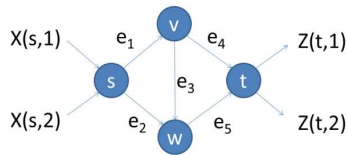
Algebrai feladat megfogalmazás



- **Bemenet**
 - A hálózat topológiája $G = (V, E)$
 - C összeköttetés igény (forrás, cél(ok), információ mennyiség)
- **Modell**
 - *Idealizált modell*: teljes szinkronizáció, hibamentes működés (nincs csomagvesztés), **késleltetés-mentes linkek**
 - $X(s, i)$ az s forrásnál található ($i = 1, \dots, \mu$) random információforrások (gyakorlatban pl.: *VLC stream csomagjai*)
 - Az $X(s)$ -ben keletkező random **információt m hosszú részekre osztjuk, és a $GF(q = 2^m)$ véges test szimbólumainak tekintjük őket**
 - Megj: bármely más prím is jó 2-n kívül, de nem jelent további előnyöket
 - $Z(t, i)$ a t nyelőben kapott szimbólumok az i forrástól
- **Feladat**:
 - 1) Kódoló részgráf meghatározása (ezzel most nem foglalkozunk, adottnak tekintjük, **általában DAG** a késleltetés-mentesség miatt)
 - 2) **Az adott kódoló részgráfon hálózati kód konstruálása**

Lineáris hálózati kódok

Példa 3: Unicast igény $G = (V, E), C = (s, t, \{1, 2\})$



$$\begin{aligned}
 Y(e_1) &= \alpha_{1,e_1} X(s, 1) + \alpha_{2,e_1} X(s, 2) \\
 Y(e_2) &= \alpha_{1,e_2} X(s, 1) + \alpha_{2,e_2} X(s, 2) \\
 Y(e_3) &= \beta_{e_1,e_3} Y(e_1) \\
 Y(e_4) &= \beta_{e_1,e_4} Y(e_1) \\
 Y(e_5) &= \beta_{e_2,e_5} Y(e_2) + \beta_{e_3,e_5} Y(e_3) \\
 Z(t, 1) &= \varepsilon_{e_4,1} Y(e_4) + \varepsilon_{e_5,1} Y(e_5) \\
 Z(t, 2) &= \varepsilon_{e_4,2} Y(e_4) + \varepsilon_{e_5,2} Y(e_5)
 \end{aligned}$$

- $m = 1$ esetén a test $GF(2)$, azaz bitenként nézem felhasználó folyamat, és ha egy adott pillanatban $X(s, 1) = 0, X(s, 2) = 1$, ekkor $\alpha \in \{0, 1\}$, és

$$Y(e_1) = (\alpha_{1,e_1} \times 0 + \alpha_{2,e_1} \times 1) \bmod 2 = \{0, 1\}$$

- $m = 8$ esetén a test $GF(256)$, azaz bájtanként nézem felhasználó folyamat, és egy adott pillanatban $X(s, 1) = 00001001 = y^3 + 1 = 9$, $X(s, 2) = 00010000 = y^4 = 16$, ekkor $\alpha \in (0, 255)$, és

$$Y(e_1) = (\alpha_{1,e_1} \times [y^3 + 1] + \alpha_{2,e_1} \times [y^4]) \bmod (y^8 + y^4 + y^3 + y + 1) = (0, 255)$$

Lineáris hálózati kódok

Valójában az előbbi egyszerű (lineáris) műveletek mindig elegendőek

- Lineáris kódok (irányított körmentes gráfokban)
 - Az $e = (v, u)$ linken küldött szimbólum a v -ben keletkező információ, és a v -be befolyó éleken érkező szimbólumok lineáris kombinációja

$$Y(e) = \sum_{l=1}^{|X|} \alpha_{l,e} X(v, l) + \sum_{e' \in be(v)} \beta_{e',e} Y(e')$$

- A nyelő (cél-) csomópontnál az információ (pl.: $m = 8$ hosszú darabja) előáll a bejövő éleken érkező szimbólumok lineáris kombinációjaként

$$Z(v, l) = \sum_{e' \in be(v)} \varepsilon_{e',l} Y(e')$$

- Feladat: $\alpha, \beta, \varepsilon$ **együtthatók meghatározása**
 - CÉL: $\forall l: Z(v', l) = X(v, l)$, azaz pontosan dekódolni tudjuk a v' nyelőben a v forrás által küldött szimbólumokat
 - Dekódolhatóság feltétele: **az M transzfer mátrix ($z = Mx$) teljes rangú** ($\det(M) \neq 0$)
 - $\alpha, \beta, \varepsilon$ -t az adott test feletti **változóknak** tekintjük
 - Következmény: M transzfer mátrix elemei a $GF(2)[\alpha, \beta, \varepsilon]$ polinomgyűrű elemei
 - Lemma:** Végtelen számú nemnulla determináns eredményező megoldás létezik. (Nekünk csak egy kell!)

S. Li, R. Yeung, and N. Cai. *Linear network coding*. IEEE Transactions on Information Theory, 49(2):371–381, 2003.

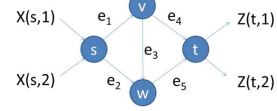
Megfelelő leíróerejű modell megalkotása

R. Koetter and M. Médard, 2003: Kapcsolat gráfelmélet és algebra között

- Tétel (unicast):** Az alábbi három állítás ekvivalens egymással:
 - (C1) Létezik r értékű folyam s és t között.
 - (C2) A minimális vágás értéke s és t között legalább r .
 - (C3) Az M transzfer mátrix determinánsa nemnulla a $GF(2)[\alpha, \beta, \varepsilon]$ polinom gyűrű felett.**
- (C1) és (C2) a jól ismert **max-flow min-cut theorem**, melyre a Ford-Fulkerson algoritmus megtalálja a megoldást.
 - DE: csak unicast összeköttetések estén tudjuk alkalmazni.
- (C3) **A gráfelméleti probléma helyett egy (könnyebb) algebrai feladatok kell megoldanunk**
 - Tetszőleges kommunikáció (multicast, multi-source multicast, stb.) esetén is csak egy mátrix determinánsát kell vizsgálnunk

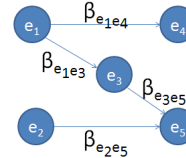
Lineáris kommunikációs hálózat

Kimenet és bemenet közötti összefüggés ($z = Mx$) átviteli mátrixszal leírható



- Irányított címkézett élgráf
 - éleknek fix sorrendje a DAG egy topologikus rendezésének megfelelően (pl.: e_1, e_2, e_3, e_4, e_5)
 - az élgráf szomszédossági mátrixa

$$F = \begin{pmatrix} 0 & 0 & \beta_{e_1 e_3} & \beta_{e_1 e_4} & 0 \\ 0 & 0 & 0 & 0 & \beta_{e_2 e_5} \\ 0 & 0 & 0 & 0 & \beta_{e_3 e_5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



- **Lemma:** $(I - F)$ -nek létezik inverze $GF(2)[\beta]$ felett.
- Bemeneti mátrix $(\mu \times |E|)$: $A = \begin{pmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & 0 & 0 & 0 \\ \alpha_{2,e_1} & \alpha_{2,e_2} & 0 & 0 & 0 \end{pmatrix}$
- Kimeneti mátrix $(|E| \times \mu)$: $B = \begin{pmatrix} 0 & 0 & 0 & \varepsilon_{e_4,1} & \varepsilon_{e_5,1} \\ 0 & 0 & 0 & \varepsilon_{e_4,2} & \varepsilon_{e_5,2} \end{pmatrix}$

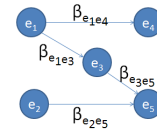
Lineáris kommunikációs hálózat

Átviteli (transzfer) mátrix elég természetesen adódik...

- **Tétel:** Tegyük fel, hogy a hálózat az A, B, F mátrixokkal adott. Ekkor a hálózat átviteli mátrixa $M = A(I - F)^{-1}B^T$ módon számolható ($z = Mx$), ahol I az $|E| \times |E|$ egységmátrix.
- **Bizonyítás:**
 - A, B valójában csak a bemeneti és a kimeneti folyamatok lineáris keverését végzik
 - $X(v, i)$ bemeneti folyamat és a $Z(v', j)$ kimeneti folyamat **impulzus válaszához** az összes lehetőséget figyelembe kell vennünk, ahogy $X(v, i)$ hozzájárulhat $Z(v', j)$ -hez
 - Az összes v és v' közötti $0, 1, 2, 3, \dots$ élből álló, végtelen számú utat figyelembe kell vennünk
 - A szomszédossági mátrix alapján egyszerűen meghatározhatjuk, mennyivel járul hozzá $X(v, i)$ a $Z(v', j)$ -hez egy k hosszú úton (F^k)
 - A végtelen összeg: $I + F + F^2 + F^3 + \dots$

Lineáris kommunikációs hálózat

Kimenet és bemenet közötti összefüggés átviteli mátrixszal leírható



• Bizonyítás (folyt):

- Viszont a **DAG tulajdonság** miatt az F szomszédossági mátrix egy felső háromszög mátrix, és mint ilyen, **nilpotens**, azaz (a fenti példára):

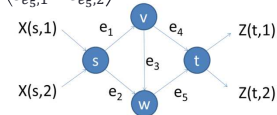
$$F^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & \beta_{e_1e_3}\beta_{e_3e_5} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, F^3 = F^4 = F^5 = \dots = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Azaz $I + F + F^2 + F^3 + \dots$ egy **véges sorösszeg**, ami pont a mértani sor sorfejtése, azaz $(I - F)^{-1}$ inverzet adja eredményül.
- A transzfer mátrix a mi példánkra **(teljes rangúnak kell lennie!)**:

$$M = \begin{pmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & 0 & 0 & 0 \\ \alpha_{2,e_1} & \alpha_{2,e_2} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & \beta_{e_1e_3} & \beta_{e_1e_4} & \beta_{e_1e_3}\beta_{e_3e_5} \\ 0 & 1 & 0 & 0 & \beta_{e_2e_5} \\ 0 & 0 & 1 & 0 & \beta_{e_3e_5} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \varepsilon_{e_4,1} & \varepsilon_{e_4,2} \\ \varepsilon_{e_5,1} & \varepsilon_{e_5,2} \end{pmatrix} = \begin{pmatrix} P & Q \\ R & S \end{pmatrix}$$

- $X(s, 1)$ bemenet és a $Z(t, 1)$ kimenet közötti összefüggés:

$$P = \alpha_{1,e_1}\beta_{e_1e_4}\varepsilon_{e_4,1} + \alpha_{1,e_1}\beta_{e_1e_3}\beta_{e_3e_5}\varepsilon_{e_5,1} + \alpha_{1,e_2}\beta_{e_2e_5}\varepsilon_{e_5,1}$$



HÁLÓZATI KÓD KONSTRUKCIÓK

Determinisztikus / random kódoló algoritmusok

Ki mondja meg, hogy hogy kódoljak?

Az F szomszédossági mátrix (kódoló együtthatók) meghatározása, hogy $\det(M) \neq 0$

- Determinisztikus hálózati kód konstrukció
 - - Előre meg kell mondanom, hogy a hálózatban pontosan ki mit csináljon (hogyan állítsa elő a bementi csomagokból a kimeneti kódot)
 - SDN esetén a kontroller, egyébként pl.: a forrás
 - + Hatékonyabb (kisebb test méret elegendő)
- Random Linear Network Coding (RLNC)
 - + Mindenki elosztott módon, véletlenszerűen kódolja a bejövő csomagokat
 - Nincs hozzáadott késleltetés: ha egy csomag nem érkezett meg, nem várunk rá („coding on-the-fly”)
 - Meglepő módon így is elég hatékonyan tudunk továbbítani (a rekódolás, azaz a **nem** end-to-end tulajdonság következményeképp)
 - - Helyreállításra nincs 100% garancia, de ha elég ügyesek vagyunk, akkor majdnem biztosan tudunk dekódolni
 - Minden új üzenet hordoz valami új információt a nyelő számára

Determinisztikus kód konstrukció multicast esetén

Folyam alapú szisztematikus algoritmus

- Kiindulás:
 - Irányított körmentes hálózati topológia (**kódoló részgráf**)
 - Késleltetésekkel nem foglalkozunk, ideális modell!
 - Meghatározzuk a maximálisan megvalósítható adatsebességet (**Hálózati kódolás alaptétele**: ez pont az unicast összeköttetések maximális adatsebességének minimuma, legyen r).
 - Javító utakkal (Ford-Fulkerson) meghatározunk s forrás csomópontból r útvonalat minden egyes nyelőbe (mindegyiken egy-egy csomagot fogunk célba juttatunk a forrásból egy adott körben).
 - **Hálózati kódolás alaptétele**: ennél több él nem is kell, a többi élen a kódoló együtthatók nullára állíthatóak (értsd törölhetőek)
 - d darab nyelő csomópont, akik mind szeretnék mind az r csomagot megkapni minden egyes körben.
- Jaggi algoritmus:
 - A fenti feladatra létezik polinom idejű determinisztikus algoritmus, mely szisztematikusan meghatározza a hálózati kódot a megoldás élein
 - A testméret legalább akkora, mint a nyelők száma, azaz $GF(q)$ megoldás létezik, ha $q \geq d$.

Jaggi, Sidharth, et al. "Polynomial time algorithms for multicast network code construction.", **IEEE Transactions on Information Theory**, 51.6 (2005): 1973-1982.

Determinisztikus kód konstrukció multicast esetén

Jaggi algoritmus vázlat

• Algoritmus

- Vesszük a javító utakkal kapott (Ford-Fulkerson) élek A halmazát egy előre megadott topologikus sorrendben (mivel DAG-ban vagyunk, ez OK)
- Válasszuk a testméretet úgy, hogy $q = 2^m \geq d$
- Minden egyes d nyelőre karban tartunk egy S_d vektor halmazt (Frontier Set), amely az r darab d -be menő élfüggetlen úton az utoljára beállított élek hálózati kódjait tartalmazza
 - Ezek a vektorok az s forrásnál $GF(q)$ feletti r -dimenziós tér egy bázisát alkotják (pl.: egység választással) minden egyes nyelőre
- Vesszük a topologikus sorrendben az éleket A -ból
 - CÉL: olyan hálózati kódok választása az adott élre, hogy minden egyes lépés után minden d nyelőre S_d -ben lévő vektorok továbbra is az r -dimenziós tér egy bázisát alkossák
 - Ezáltal az algoritmus leállásakor (azaz a kész kódoláskor) minden egyes d nyelőbe beérkező r úton lineárisan független vektor érkezik

Jaggi, Sidharth, et al. "Polynomial time algorithms for multicast network code construction.", *IEEE Transactions on Information Theory*, 51.6 (2005): 1973-1982.

Determinisztikus kód konstrukció multicast esetén

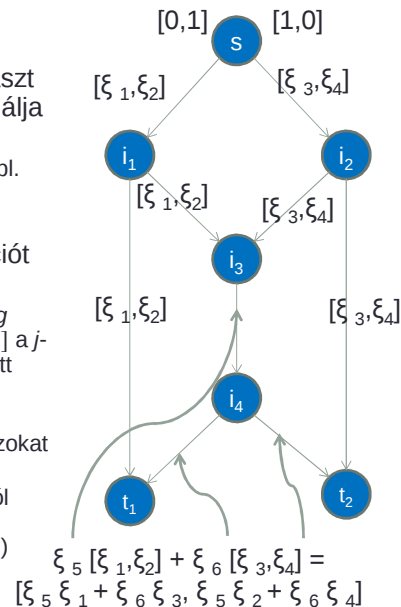
Jaggi algoritmus hatékonysága – Hogyan válasszuk a linkek hálózati kódját?

- Véletlenszerűen addig próbálkozunk, amíg jó nem lesz
- Gyorsítás (hatékony hálózati kód választás)
 - Minden egyes l élhez S_d -ben karbantartunk egy $\underline{l}$ vektort, mely merőleges S_d -ben lévő többi $r-1$ vektorra.
 - Ekkor az a link kódjának lineárisan függetlenség ellenőrzéséhez elég csak az $\underline{l}v = 0$ egyenlőséget vizsgálnunk valamennyi d -re (ahol a benne van az r út valamelyikében).
 - A hálózati kód választás már történhet akár véletlenszerűen is, amíg minden feltétel nem teljesül.
- Jaggi algoritmus komplexitása
 - $O(|A|dr(r + d))$

Jaggi, Sidharth, et al. "Polynomial time algorithms for multicast network code construction.", *IEEE Transactions on Information Theory*, 51.6 (2005): 1973-1982.

Random Linear Network Coding (RLNC)

- Minden csomópont véletlenszerűen választ (lokális) kódoló együtthatókat és konstruálja meg a hálózati kódot a kimeneti linkjein
- Válasszuk a testméretet $q = 2^m > d$, nézzünk pl. bájtankénti kódolást $m = 8$.
- A nyelőnél tudni kell, hogy a bemeneti csomag milyen forrás csomag kombinációt tartalmaz:
 - a csomag fejlécében tároljuk *global encoding vector*-t ($\underline{g}$), kezdetben $\underline{g} = \underline{e}_j = [0 \dots 010 \dots 0]$ a j -edik forrás csomagra (a transzfer mátrix adott csomaghoz tartozó sora)
 - minden továbbításakor a csomag bájtjain (**szimbólumain**), és magán a $\underline{g}$ -n is ugyanazokat az algebrai műveleteket hajtjuk végre
 - A nyelőknek a kapott csomagok $\underline{g}$ vektoraiból alkotott mátrix sorain kell Gauss-eliminációt végezni (továbbá a csomag szimbólumain is)



Random Linear Network Coding (RLNC)

- A t_1 nyelőnél a bejövő csomagok $\underline{g}$ vektoraiból alkotott mátrix:

$$\begin{pmatrix} \xi_1 & \xi_2 \\ \xi_5 \xi_1 + \xi_6 \xi_3 & \xi_5 \xi_2 + \xi_6 \xi_4 \end{pmatrix}$$
- Ha a fenti mátrix invertálható, akkor a csomagok tartalma dekódolható
- Minimális test méret (kis számítási komplexitás érdekében), ami felett a mátrix minden egyes nyelőnél nagy valószínűséggel dekódolható lesz:

$$\left(1 - \frac{|d|}{2^m}\right)^{|E|}$$
- A példára (9 él, 2 nyelő, bájt) a dekódolhatóság valószínűsége 0.9825.

