

# Új protokoll terve vezeték nélküli MIDI kapcsolatok megvalósítására Bluetooth rendszerben

HUSZTY CSABA, BALÁZS GÉZA

Távközlési és Médiainformatikai Tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem  
info@midioverb.com

**Kulcsszavak:** vezeték nélküli MIDI, Bluetooth

A cikk egy új protokoll terveit mutatja be vezeték nélküli MIDI kapcsolatok megvalósítására Bluetooth rendszerben. A gyakorlati felhasználást figyelembe véve a terv közel tetszőleges összeköttetési topológiát tesz lehetővé. Bemutatjuk egy általános Bluetooth-alapú MIDI rendszer tervét és adatátviteli módját, kiszámoljuk a késleltetését és megvizsgáljuk alkalmazhatóságának korlátait, egyúttal javaslatot téve a rendszer funkcióinak néhány lehetséges további bővítésére.

## 1. Bevezetés

A MIDI (Musical Instrument Digital Interface) legfőbb feladata a hangkeltéshez szükséges vezérlési információk átvitele mellett a rendszer szinkronizációja [1]. A MIDI protokollt használó berendezések többnyire három csatlakozási lehetőséggel vannak ellátva. Az IN csatlakozót a szomszédos egység OUT csatlakozójával fizikai kábellel kötik össze, a THRU kimenet segítségével pedig az összeköttetés láncszerkezetűvé alakítható úgy, hogy a THRU portra az IN-re érkező adatok kerüljenek.

Ezzel a módszerrel új eszközök használata nélkül csak korlátozott bonyolultságú összeköttetéseket lehet létrehozni. Gyakori probléma több kimenet egy bemeneten való egyesítése is, amelyet szintén csak kiegészítő berendezéssel, úgynevezett MIDI Merger-rel lehet megvalósítani. Tekintettel az összekötő kábelek néhány méteres maximális hosszára, az összekötendő berendezések nem helyezhetők el akármilyen távolságban és sorrendben, valamint azt is számításba kell venni, hogy 15-20 egység esetében már az összeköttetést biztosító elosztó és összekapcsoló berendezések is számottevő helyet, árnövekedést és késleltetést vihetnek a rendszerbe. A szabványnak az implementációra vonatkozó korlátaival a továbbiakban nem foglalkozunk.

Az olcsó vezeték nélküli alkalmazások közül többek között a Bluetooth is ideális a MIDI összeköttetések helyettesítésére. Az egyes egységek kis teljesítményfelvétele, elegendő hatótávolsága és kedvező tulajdonságú zavarérzékenysége is alkalmazhatóvá teszi erre a célra. A Bluetooth-technikával kialakított MIDI összeköttetések nem pusztán a kábelt, hanem az összeköttetést segítő egyéb berendezéseket is ki tudják részben kuszóbolni, illetve integrálhatóvá tenni már létező berendezésekbe. Bár más munkák már foglalkoztak a MIDI és a Bluetooth együttes alkalmazásával [4], e cikk új módszert mutat be a MIDI összeköttetések megvalósítására és továbbmegy a MIDI kábel helyettesítésénél.

## 2. MIDI összeköttetések megvalósítása Bluetooth rendszerben

### 2.1. Az alkalmas kapcsolati és csomagtípus megválasztása a MIDI-hez

A Bluetooth jelenlegi, 1.2 változata pikonetenként egy master egységet és legfeljebb hét aktív slave egységet támogat, bár parkolási állapotban hétnél több slave is lehet a pikonetben [2,3].

A csatorna-hozzáférést a master egység vezérli. Egy pikoneten belül a master egység órajeléhez igazodik a pikonet összes többi tagja. Minden egység időben és ugratásban (frekvenciában) is a masterhez szinkronizált. A master csak páros számú időszeletekben (slot) kezdeményezhet adást, páratlanokban vételet, a slave-ek pedig fordítva: csak páratlan számú időszeletben kezdeményezhetnek adást. A fentiek csak a kezdeményezésre vonatkoznak: a már megkezdett adás eltarthat több slot ideig is.

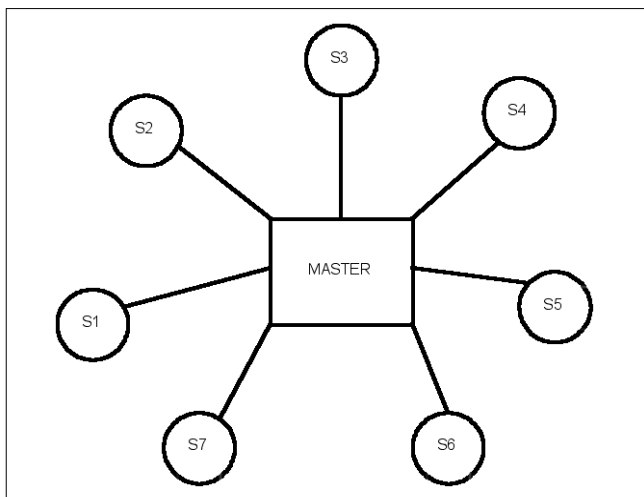
A MIDI alkalmazás szempontjából az ACL átvitelt választjuk. Ebből egy pikoneten belül egyidejűleg egy adatcsatorna működése támogatott. Az átvitelhez a szabványban hétféle csomagot definiáltak, az ezekkel átvihető byte-ok maximális számát a *táblázatban* tüntetjük fel. Az átvitelre az M (Medium) csomagot választjuk, mert az M csomag esetén 2/3 FEC hibajavító kódolást alkalmaznak, míg a H (High) csomagok esetében egyáltalán nincs hibajavító kódolás.

### 2.2. A hub-alapú topológia

A MIDI-s alkalmazást nagyban megkönnyíti a csillagpontos elrendezésű Bluetooth. Broadcast jellegű üzenetek és megfelelő beállítások mellett az adatfolyamot módosító (pl. Merge) bármely összeköttetés megvalósítható.

1. táblázat  
BT csomagok maximális  
byte-száma

csomag típus	max. átvitel
DM1	18
DH1	28
DM3	123
DH3	185
DM5	226
DH5	341
AUX1	30



1. ábra Hub-based topology

A kapcsolat beállításakor már tudjuk, hogy melyik egység működik MIDI Out-ként, melyik MIDI In-ként. A Bluetooth alapú megvalósítás járulékosan annak a lehetőségét is magával hozza, hogy egy egység kimenete önmagába visszahurkolódjon – ez a MIDI Echo funkció.

**2.3. Az alapvető MIDI összeköttetések megvalósítása**

A tetszőleges összeköttetési topológiához a MIDI In, Out, MIDI Thru Box (Hub), MIDI Merge, Echo és Patchbay eszközöket és funkciókat helyettesítő logikai összeköttetéseket kell megvalósítani (2. ábra).

*MIDI Out/In (MIDI kábel)*

A slave egységekhez kapcsolódó MIDI eszközök adatai közvetlenül, a master egységen keresztül jutnak el a másik eszközhöz. Az In és Out portokat a felhasználó jelöli ki. A master egység az első slave-et (S1) poll üzenettel lekérdezi, ennek hatására az beküldi MIDI adatait, majd a master broadcast üzenetben kiküldi az adatokat az összes slave-nek. A beállított topológiától függően az S1 egység figyelmen kívül hagyhatja a hozzá megérkezett adatokat.

*MIDI Hub (Thru Box)*

Egy bemenet több kimenettel történő összeköttése logikai összeköttetésekkel szintén egyszerűen megvalósítható. A master egység poll üzenetére az S1 slave

adatot küld. A rendszerben nincs az S1-en kívül további Out port, így a lekérdezés befejeződik, a master egység broadcast üzenetben továbbítja az adatokat az összes slave felé. Az S1 slave azonban nincs beállítva In-ként, így az nem fogja figyelembe venni a hozzá érkező MIDI üzeneteket. Járulékos előny, hogy több slave esetén az adatok megérkezésének késleltetése nem nő meg.

*MIDI Merge modul*

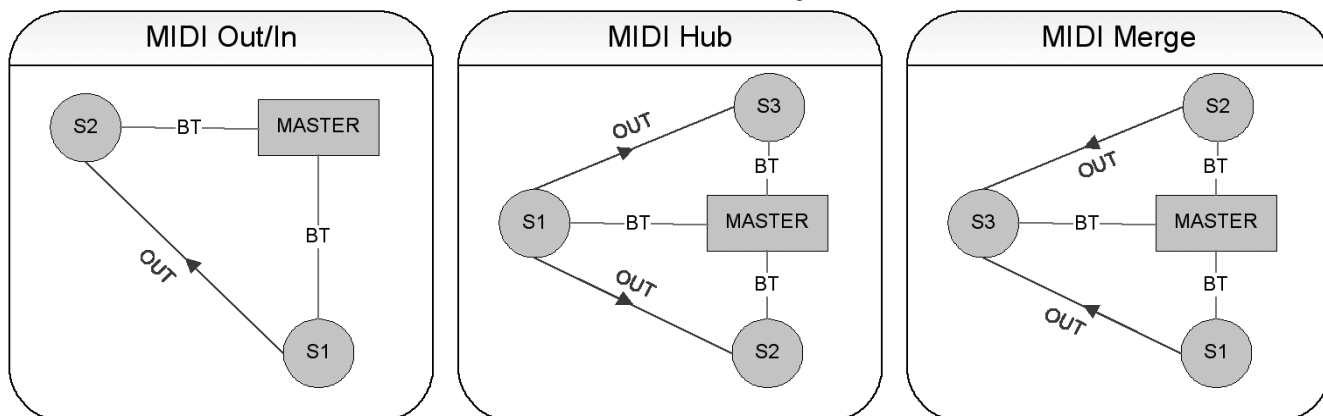
Két vagy több bemenet egy kimenetű történő egyesítésére a Master egység az S1 slave-nek egy Poll üzenetet küld, melyre az S1 slave válaszol, és visszaküldi az időközben összegyűlt MIDI adatait. Az adatok a master egységben tárolódnak, amelyek az S2 egység lekérdezése és adatbeküldése után egy többes-küldés (broadcast) üzenetben egyszerre kerülnek elküldésre. Szükséges azonban a MIDI adatokat még kiküldésük előtt feldolgozni annak érdekében, hogy konzisztenciájuk megmaradjon, de ezt a feladatot az S3 egység lokálisan el tudja végezni még azelőtt, hogy a hozzá kapcsolt MIDI berendezésnek eljuttatná az adatokat.

**3. A javasolt protokoll terve és időzítése**

A MIDI adatok logikai egységeit, az üzeneteket a Bluetooth rendszerben csomagokba kell szervezni. Az időegység (MIDI slot) alatt átvitt csomagokban levő MIDI byte-okat mindig üzenethatáron tördeljük, ez alól egyetlen kivétel a System Exclusive Message (SysEx), amely tetszőlegesen hosszú lehet. Sorban, egymás után olvassuk be az előre definiált topológia szerinti MIDI Out portokról a beérkező MIDI byte-okat, ezeket minden Out számára konstans hosszúságú csomagokba foglaljuk, majd elküldjük a megfelelő In-ekre.

Egyes esetekben a küldés előtt fel kell dolgozni a csomagokat, hogy ne haladják meg a MIDI sávszélességét (például Merge). Egy ilyen ciklust nevezünk MIDI slot-nak, amelynek hosszúságát időben állandónak kell rögzíteni és tartani. Tekintettel arra, hogy változó számú byte érkezik a MIDI slot ideje alatt, a csomag hosszát és a késleltetési időt maximális adatmennyiség esetére számoljuk.

2. ábra MIDI összeköttetések megvalósítása



A MIDI slot időzítése (DM3 válasz csomag esetében egyszerűes küldéssel):

1. A Master Poll üzenete (DM1 csomag) a 0. Bluetooth slotban az első olyan slave-nek, amelyhez egy Out porttal rendelkező MIDI eszköz van csatlakoztatva [5].
2. A megcímezett Slave visszaküld egy állandó hosszúságú DM3 csomagot az addig felgyülemlt MIDI adataival az 1. BT slotban. Az átvitel a 3. BT slotban ér véget. Ha van még Out a rendszerben, a folyamat a soron következő slave-vel megismétlődik.
3. 2 darab üres BT slot következik, majd az Out-ok számától függően DM3, DM5 vagy DH5 típusú csomagban broadcast csomag az összes slave-nek. A master egység a slave-ek adatainak fogadásával egyidőben (a BT modulok UART portjai full-duplex üzeműek) elkezdheti összeállítani a küldendő broadcast csomagot, így csak az utolsó slave adatainak beillesztését kell a broadcast csomag kiküldése előtt megvárni. Mivel azonban az utolsó slave adatcsomag megérkezése után páros BT slot következik, a várakozás miatt meg kell várni a következő páros BT slot-ot, mielőtt a broadcast üzenetet kiküldենék. Ez okozza a 2 BT slot idejű további késleltetést.
4. Végül ismét egy üres BT slot következik, mert a következő poll ciklus csak páros BT slot-ban kezdődhet és a broadcast csomagra nem válaszolhatnak az egységek. A ciklus ezután az első lépéstől ismétlődik.

Összesen tehát

$$N_{BT\_SLOTS} = 4 \cdot O + 2 + x + 1$$

BT slot-nak megfelelő idő telik el, ahol  $O$  az Out-ok száma,  $x$  pedig 3 DM3 vagy 5 DM5 csomag esetében. Legyen a MIDI slot hossza byte-okban

$$B = S_{MIDI} \cdot T_{BT},$$

ahol a MIDI vonal átviteli sebessége  $S_{MIDI} = 3125$  byte/s,  $T_{BT}$  pedig egy BT slot ideje: 625  $\mu$ s.

Az időzítés tervezése érdekében kiszámítjuk, mekkora átvindó adatmennyiségre lehet igény. Egy MIDI slotban a maximálisan átvitt MIDI byte-ok száma egy aktív Out esetén:

$$B_{1\_OUT} = S_{MIDI} \cdot T_{BT} \cdot (1+1+2+1+1) = 6 \cdot S_{MIDI} T_{BT} = 11.71875 \Rightarrow 12 \text{ byte.}$$

Felfelé kerekítettünk a legrosszabb esettel számolva. Az összeg tagjai: 1 poll a mastertől a slave felé, 1 válasz a slave-től a master felé, 2 üres slot, 1 broadcast üzenet a mastertől a slave-ek felé és 1 üres slot, így összesen 6 slot hosszúságú egy MIDI slot.

A fent kiszámolt eredménynél a tényleges érték kettővel nagyobb, mert a MIDI csomagok általában kettő illetve három byte-os üzenetekből állnak, így előfordulhat, hogy a 11-ik byte után egy 3 byte-os üzenet kezdődik, amit még át kell vinni. A maximálisan átvindó MIDI byte-ok száma tehát

$$B_{\text{ÁTVIENDŐ}} = B_{1\_OUT} + 2 = 14 \text{ byte.}$$

Ahhoz, hogy az időzítést terhelt és terheletlen rendszer esetén egyaránt tartani tudjuk, válasszunk minden átvitt csomagot egyforma hosszúságúnak, függetlenül attól, hogy hány byte hasznos adat van benne. Az átvitelhez további 2 byte adminisztrációs információ szükséges (fejléc és lábléc). Az 1 byte-os fejléc az Out sorszámát fogja tartalmazni, amely alapján az esetlegesen implementált Merger programmodul azonosítja majd, melyik folyamba kell beillesztenie az érkező byte-okat, az ugyancsak 1 byte-os lábléc pedig a csomag végét jelző byte-ot, amit egy a MIDI szabványban nem definiált byte-tal valósítunk meg. Összesen tehát

$$B_{\text{MERGER}} = B_{\text{ÁTVIENDŐ}} + 2 = 16 \text{ byte}$$

kerül átvitelre egy MIDI slot-ban. Mivel a DM1 csomag pontosan 18 byte-ot képes átvinni és 1 BT slot ideig tart, ez ideális ennek az adatmennyiségnek az átvitelére.

Egyszerűsége és rugalmassága miatt kézenfekvő, hogy UART-típusú Host Controller Interface-szel (HCI) épített Bluetooth modult használjunk a realizációban, így a következőkben UART-alapú rendszerre számítjuk ki az időzítést.

A korrekt számításokhoz először is figyelembe kell venni, hogy az UART HCI BT moduloknál a csomagformázáshoz további 5 byte-ot át kell vinni (1 byte ACL azonosító, 2 byte connection handle, a master-slave fizikai kapcsolat azonosítójával; valamint 2 byte flag a csomag hosszáról, amely az 5 fejléc byte nélkül veendő figyelembe). Ezek a byte-ok a küldés során a levegőbe nem kerülnek ki, így csomagtypust nem kell váltanunk miattuk. Összesen tehát 21 byte fog a BT modulhoz megérkezni, és a master egység is ugyanennyit kap a saját soros portjáról.

Ez egy 1 382 400 bit/s sebességű UART esetén 152  $\mu$ s-ig tart (1 start bit + 8 adatbit + 1 stop bit = 10 bit. Az idő:  $T_{\text{UART}} = 10 \cdot 21 / 1382400 = 152 \mu$ s), ami 24,3%-a a BT slot hosszának.

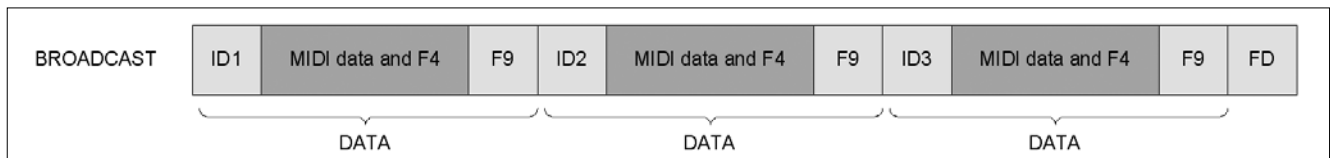
Ugyanez a számítás 2 Out-ot feltételezve már (DM3-as csomaggal) 506  $\mu$ s-ra adódik, de még nem éri el a BT slot hosszát – vagyis az időzítés tartható.

3 Out esetében már más a helyzet:  $B_{3\_OUT} = 20 \cdot S_{MIDI} T_{BT} = 40$  byte, mert a broadcast csomag már 5 slot hosszúságú. A broadcast hossza 132 byte – ez már DM5-ös csomagot igényel. Az átvitel a slave egységeknél 354  $\mu$ s, a master egységnél 998  $\mu$ s-is tart.

4 Out esetében a broadcast-ra 204 byte adódik. Az átviteli idő slave-eknél 405  $\mu$ s, a masternél 1519  $\mu$ s.

Végeredményben egy pikoneten belül még akár 5 Out is megvalósítható lenne, csak itt már a broadcast-hoz nem lehet M csomagot használni, mert 295 byte-ot kellene elküldeni, ez pedig a DM5-be sem fér már bele. Az átvitel időtartama slave esetén 463  $\mu$ s, master esetén pedig 2177  $\mu$ s. Ez utóbbi érték BT slotokban kifejezve 3.48, tehát még mindig nem veszélyes az időzítésre.

Ennél több MIDI Out nem valósítható meg egy pikoneten belül, mivel ezt a BT sávszélessége nem teszi lehetővé.



3. ábra A broadcast csomag felépítése

Adatvédelem szempontjából sajnos a fenti megoldások meglehetősen rosszak – annak ellenére, hogy késleltetésük kétségkívül a legkisebb –, hiszen csak egyszer küldenek el minden adatot a címzettnek. A 2/3 FEC kódolás valamivel javítja a zavarérzékenységet, de az érték továbbra is függ az adó egységek térbeli elhelyezésétől és távolságától is. A legtriviálisabb módja a adatvesztés elkerülésének a többszörös küldés.

1 Out esetén akár háromszor is megtehetjük ezt, ebben az esetben a késleltetési idő 16 ms, de négyszeres újraküldés esetén is csak 21 ms.

2 Out-nál már csak kétszer küldhetjük újra az üzeneteket (mind a poll-okat, mind a válaszokat, mind pedig a broadcast-ot), ekkor a késleltetési idő 18,75 ms.

3 Out-nál DM5 csomag esetén még éppen meg lehet kétszer ismételni a lekérdezést, ekkor viszont a broadcast már csak egyszer fér bele a MIDI slot-ba. Ha a broadcast-hoz DH5-ös csomagot használunk, akkor megismételhető a küldés.

3-nál több Out-nál már sajnos semmilyen ismétlést nem tudunk megvalósítani a MIDI slot idejében a jelenlegi Bluetooth szabvány adatsebessége mellett.

Az előzőeket a 2. táblázatban foglaltuk össze. (Megj.: 1 Out esetében a 14-szeres újraküldés a jelenlegi Bluetooth szabvány teljesítőképességének határa.

### 3.1. A MIDI csomagok felépítése

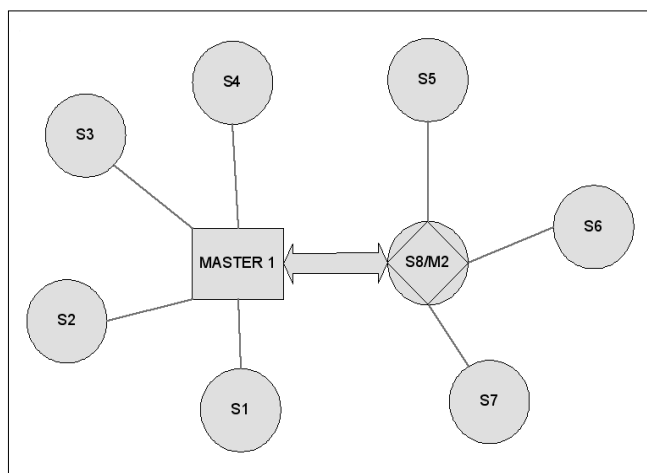
Maximálisan 3 Out-ot feltételezve a broadcast csomag a 3. ábra szerint épül fel.

A csomag DATA jelzései az egyes Slave egységek MIDI adatait jelentik. Az adatcsomag lezárása a MIDI-ben nem definiált funkciójú 0xF9-cel történik, a broadcast csomag lezárása pedig a szintén nem definiált 0xFD-vel. Ha a MIDI berendezés a nem definiált 0xF4, 0xF5, 0xF9 vagy 0xFD MIDI byte-ok közül valamelyiket saját speciális működéséhez felhasználja, akkor a csomag struktúráját és az időzítést is módosítani kell, hogy ezeket a byte-okat is át tudjuk vinni.

2. táblázat

Az átvitel késleltetése – a kiemelt sorban levők szerint javasoljuk az összeköttetést megvalósítani.

	MIDI kimenetek száma	Válasz csomag típusa (Slave)	Broadcast csomag típusa (Master)	Késleltetés (slot)	Átvitt slot-ok (sorrendben)	Késleltetés	fizikailag átvitt MIDI byte-ok	logikailag átvitt MIDI byte-ok
1-szeres küldés	1	DM1	DM1	6	1 lekérdezés (poll), 1 válasz, 2 üres, 1 broadcast, 1 üres	<b>3,75 ms</b>	12	14
	2	DM3	DM3	14	2x1 lekérdezés (poll), 2x3 válasz, 2 üres, 3 broadcast, 1 üres	<b>8,75 ms</b>	28	30
	3	DM3	DM5	20	3x1 lekérdezés (poll), 3x3 válasz, 2 üres, 5 broadcast, 1 üres	<b>12,50 ms</b>	40	42
	4	DM3	DM5	24	4x1 lekérdezés (poll), 4x3 válasz, 2 üres, 5 broadcast, 1 üres	<b>15,00 ms</b>	47	49
	5	DM3	DH5	28	5x1 lekérdezés (poll), 5x3 válasz, 2 üres, 5 broadcast, 1 üres	<b>17,50 ms</b>	55	57
2-szeres küldés	1	DM3	DM3	18	2x1 lekérdezés (poll), 2x3 válasz, 2 üres, 2x3 broadcast, 2x1 üres	<b>11,25 ms</b>	36	38
	2	DM3	DM5	30	2x2x1 lekérdezés (poll), 2x2x3 válasz, 2 üres, 2x5 broadcast, 2x1 üres	<b>18,75 ms</b>	59	61
	3	DM3	DH5	38	2x3x1 lekérdezés (poll), 2x3x3 válasz, 2 üres, 2x5 broadcast, 2x1 üres	<b>23,75 ms</b>	75	77
3-szoros küldés	1	DM3	DM3	26	3x1 lekérdezés (poll), 3x3 válasz, 2 üres, 3x3 broadcast, 3x1 üres	<b>16,25 ms</b>	51	53
	2	DM3	DH5	44	3x2x1 lekérdezés (poll), 3x2x3 válasz, 2 üres, 3x5 broadcast, 3x1 üres	<b>27,50 ms</b>	86	88
4-szeres küldés	1	DM3	DM3	34	4x1 lekérdezés (poll), 4x3 válasz, 2 üres, 4x3 broadcast, 4x1 üres	<b>21,25 ms</b>	67	69
5-szörös küldés	1	DM3	DM3	42	5x1 lekérdezés (poll), 5x3 válasz, 2 üres, 5x3 broadcast, 5x1 üres	<b>26,25 ms</b>	83	85



4. ábra  
Scatternet multifunkcionális (master-slave) egységgel

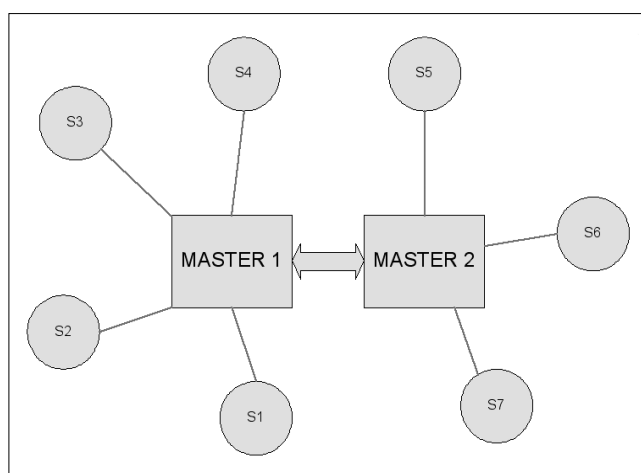
A broadcast csomag maximális hossza 238 byte (DH5 csomag, 3 Out, kétszeres küldés). A lekérdezés (poll) üzenet tartalma közömbös.

### 3.2. A késleltetés csökkentése scatternet alkalmazásával

Ha egyszerre több master-t alkalmazunk, csökkenthető a késleltetési idő, mert az Out egységeket szét lehet osztani a master-ek között. Sajnos nem könnyű elkerülni, hogy a két master egység véletlenül se adjon ugyanazon a frekvencián, bár a Bluetooth 1.2 szabványával ez megoldható, ezért még inkább szükség lehet a többszörös újraküldésre, ami viszont megnöveli a késleltetést. A tetszőleges topológia ezek után a pikonetek összekapcsolásával valósítható meg.

Scatternet kialakítása a pikonetekből többféleképpen is lehetséges: megtehető, hogy egy slave egység egy másik pikonetben master-ként üzemel (4. ábra), vagy hogy a master egységeket egyszerűen összehalozzuk egy nagysebességű busszal (5. ábra). A MIDI megvalósítása szempontjából az első módszer semmiképpen sem megfelelő, hiszen a többfunkciós S8/M2 egység egyszerre csak egyik funkcióját láthatja el – a szinkronitás nem tartható. Felmerül viszont a kérdés, hogy mi történik akkor, ha kiegyenlítetlenül helyezkednek el az Out funkciójú slave egységek az egyes pikonetekben. Az időzítés tárgyalásakor már láttuk, hogy ahogyan nő az Out-ok száma, úgy nő a késleltetés, és úgy változnak az alkalmazandó protokoll paraméterei is, vagyis az újraküldések száma (ismételt adás) és a Bluetooth csomagok típusa.

Egy olyan rendszer megépítését tartjuk legcélszerűbbnek, melyben a felhasználónak nem kell az újabb master egységek üzembe helyezésékor az egész addigra meglévő és működő rendszerét újrakonfigurálni, és kényelmetlen, hosszadalmas úton az új összeköttetési topológiát a rendszerbe bevinni. Az OUT-ok elosztása során törekedni kell arra, hogy minden master a lehető legkevesebb Out-ot kapja meg, és hogy az ugyanazokat az Out-okat megkapó In-ek lehetőleg egy pikonetbe kerüljenek, vagyis minél kevesebb adat kerüljön a pikoneten kívülre.



5. ábra  
Scatternet nagysebességű kapcsolattal

## 4. Összefoglalás

A cikkben javaslatot tettünk egy Bluetooth protokoll-koncepcióra és megvizsgáltuk annak megvalósíthatóságát vezeték nélküli MIDI kapcsolatok megvalósítására. Annak ellenére, hogy a Bluetooth segítségével történő megvalósítások eddig nem jártak túl nagy sikerrel, az itt felvázolt rendszer egészében kihasználja a Bluetooth egyre növekvő teljesítőképességét. Állandó késleltetési idő tartása mellett biztonságos adatforgalmat is lehetővé tesz. A cikkben ismertetettek alapján megépített rendszer bővíthető oly módon, hogy a felhasználónak semmit sem kell megváltoztatnia az addigra már működő régi rendszerében, csökkentheti a rendszer késleltetési idejét, és növeli a kiszolgálható egységek számát.

A legfontosabb konklúziók a rendszer megvalósíthatóságára: (1) legalább kétszeres újraküldést kell alkalmazni, (2) max. 3 Out lehet 1 pikonetben, amelyből a saját pikonetből származó Out-ok száma maximálisan 2 lehet. (3) Ha 1 Out sincs bekötve, nem alkalmazunk broadcast-ot, 1 és 2 bekötött Out esetén egyaránt DM3 csomagot alkalmazunk (a késleltetés 16,25 ms és 18,75 ms), 3 Out esetén pedig DH5-öt (23,75 ms). (4) Legfeljebb 3 master egységet kapcsolhatunk össze egy légtérben úgy, hogy minden egységhez nagy valószínűséggel megérkezzenek az adatok Bluetooth 1.1 szabvány alkalmazása esetén. Ez nem vonatkozik az 1.2 szabványra, ami lehetővé teszi, hogy egy pikonetben tetszés szerint osszuk el a rendelkezésre álló 79 frekvenciát.

## Irodalom

- [1] MMA MIDI Specifications, 1983-2003
- [2] Specification of the Bluetooth System v1.1
- [3] Specification of the Bluetooth System v1.2
- [4] J. Keniston, S. Sturdivant (2003)  
Wireless MIDI Network Implemented Via Bluetooth
- [5] R. Mettala: Bluetooth Protocol Architecture; v 1.0  
(Bluetooth White Paper Document # 1.C.120/1.0)