



# Фрагментація IP-дейтаграм

# Основні положення фрагментації

**Фрагментація** – процедура розділення дейтаграми на декілька дейтаграм меншого розміру.

Фрагментація необхідна, оскільки кожний сегмент (канал) мережі має конкретні параметри передачі: пропускну спроможність, значення MTU тощо.

Кожного разу при передачі дейтаграми IP-модуль в процесі маршрутизації визначає, з якого інтерфейсу вона має бути відправлена, і запитує значення **MTU** (Maximum Transmission Unit) цього інтерфейсу. Після цього IP-модуль порівнює MTU з розміром дейтаграми і, якщо остання перевищує максимальну одиницю передачі, розбиває дейтаграму на фрагменти.

**Фрагментація може виконуватись як на кінцевій станції, що відправляє IP-дейтаграму, так і на будь-якому проміжному комунікаційному модулі.**

# Основні положення фрагментації

**MTU** – максимальний розмір поля даних (поля корисного навантаження) кадру, який формується протоколом канального рівня, і може бути переданий без фрагментації.

Зазвичай в **MTU** не входить заголовок протоколу, хоча в деяких мережних технологіях і може враховуватись.

# Основні положення фрагментації

**IP-дейтаграма (і все повідомлення), що була фрагментована, збирається тільки в кінцевій станції (пункті призначення).**

На відміну від IP-мереж, в деяких мережах інших технологій використовується поетапна схема, при реалізації якої фрагменти збираються в найближчому пункті пересилки.

Фрагментація, по суті, є прозорою для транспортного рівня.

Фрагментацію необхідно враховувати на рівні протоколів TCP та UDP у випадку, коли треба попередити зниження продуктивності, яке супроводжує фрагментацію.

Використовуючи UDP, легко спровокувати фрагментацію. В протоколі TCP передбачені заходи попередження фрагментації, і додаток практично не може змусити TCP пересилати сегмент такого розміру, що вимагає фрагментування.

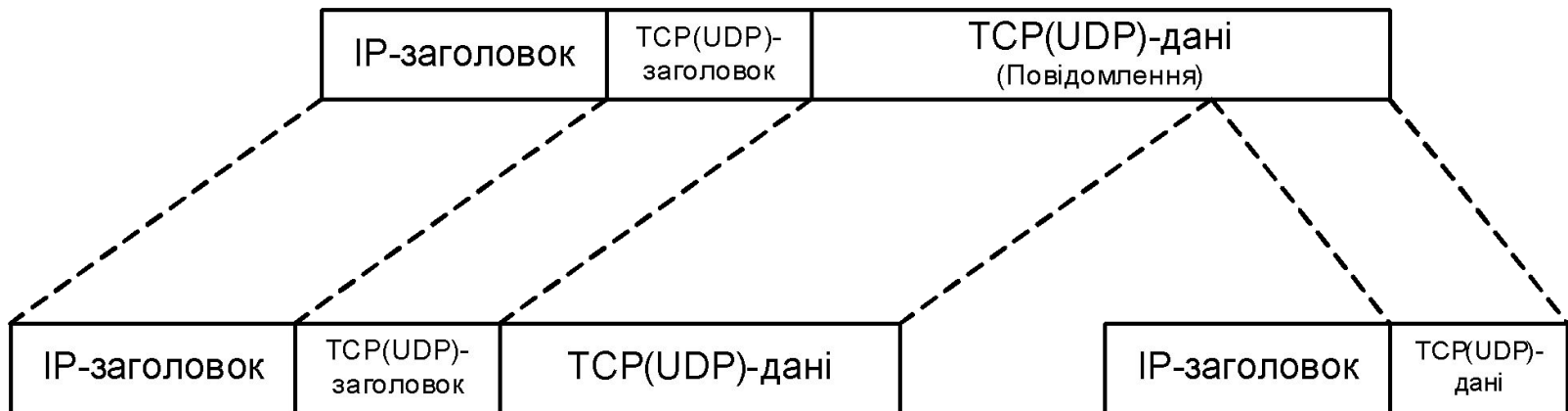
# Основні положення фрагментації

Якщо IP-дейтаграма була фрагментована, то кожний фрагмент стає окремим незалежним блоком зі своїм заголовком. Вони аналізуються та маршрутизуються комунікаційними вузлами (маршрутизаторами) незалежно, і тому порушується послідовність приходу в кінцевий пункт призначення.

При втраті хоча б одного фрагменту дейтаграми ініціюється повторна передача всієї дейтаграми. Але оскільки в самому протоколі IP не передбачені процедури таймауту та повторної передачі, ці функції необхідно перекласти на більш високі рівні.

# Основні положення фрагментації

Приклад фрагментації IP-дейтаграми, що містить TCP-сегмент (UDP-дейтаграму)



# Основні положення фрагментації

**Одне з основних правил фрагментації вимагає:**

розмір області даних, тобто розмір поля корисного навантаження дейтаграми (сюди відноситься вся інформація, що міститься в дейтаграмі за IP-заголовком) був кратним 8 байтам для всіх фрагментів, крім останнього (останній фрагмент може містити навіть всього 1 байт).

## Фрагментація дейтаграм (приклад)

**Приклад.** Повідомлення 4080 байт, що має ідентифікатор 14825 і обробляється протоколом UDP, необхідно терміново передати з максимальною безпекою в межах даного континенту через мережу з максимально допустимим розміром корисного навантаження 1400 байт. Які будуть значення всіх полів заголовка дейтаграми в кожному з отриманих фрагментів при відсутності необов'язкових опцій (параметрів).



# Фрагментація дейтаграм (приклад)

## Фрагменти дейтаграм

Перший фрагмент



Всі фрагменти, крім першого



# Фрагментація дейтаграм (приклад)

4	5	126	4108
14825		0	0
128	17	Контр. сума	
IP-адреса відправника			
IP-адреса отримувача			
Дані (байти 1 – 4088: UDP-заголовок та повідомлення)			

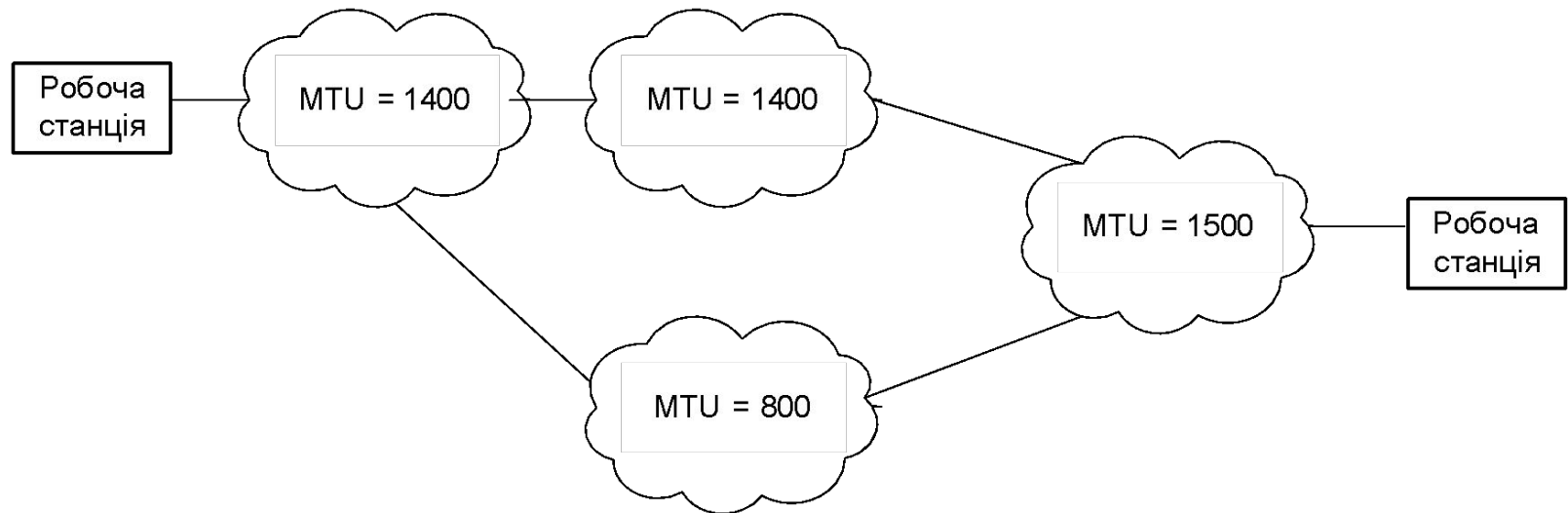
4	5	126	1396
14825		1	0
127	17	Контр. сума	
IP-адреса відправника			
IP-адреса отримувача			
Дані (UDP-заголовок та байти 1 – 1368 повідомлення)			

4	5	126	1396
14825		1	172
127	17	Контр. сума	
IP-адреса відправника			
IP-адреса отримувача			
Дані (байти 1369 – 2744)			

4	5	126	1356
14825		0	344
127	17	Контр. сума	
IP-адреса відправника			
IP-адреса отримувача			
Дані (байти 2745 – 4080)			

# Фрагментація дейтаграм (приклад)

## Структура мережі



# Фрагментація дейтаграм (приклад)

