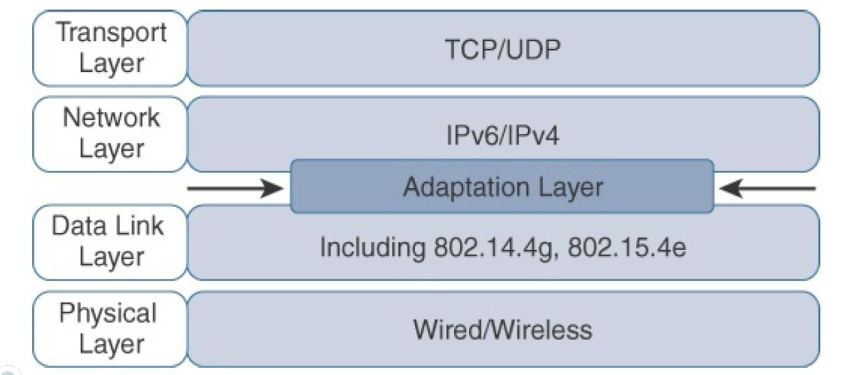
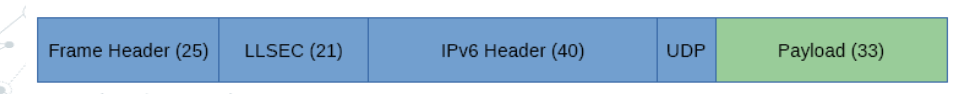
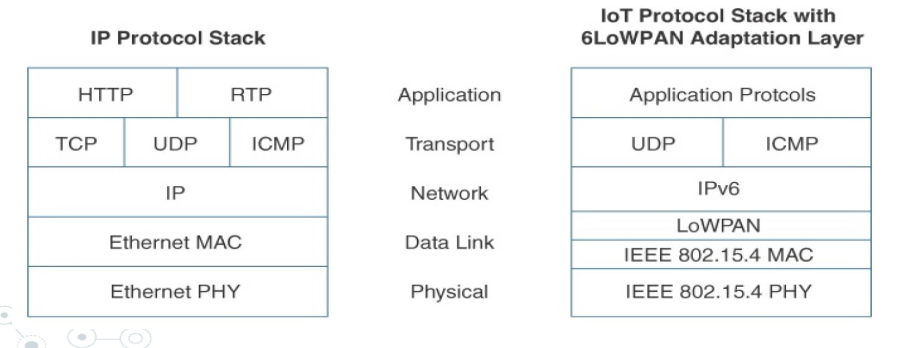
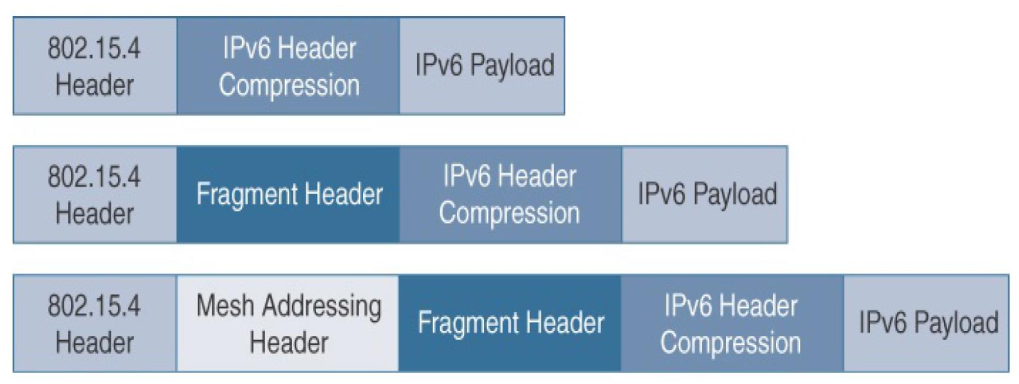
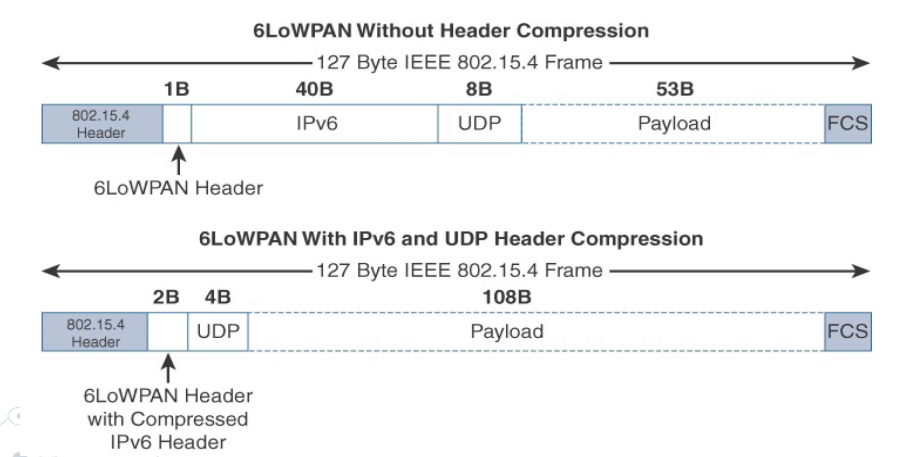
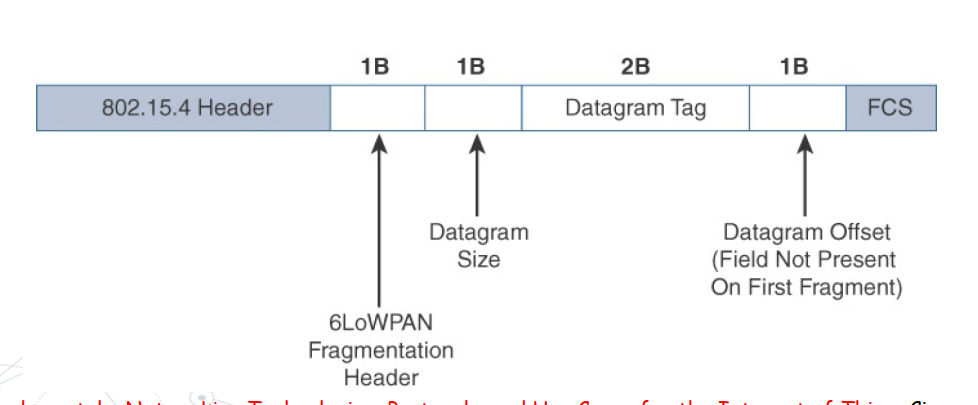
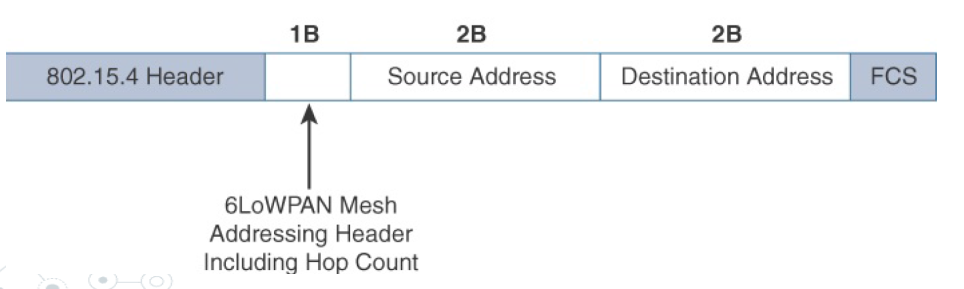
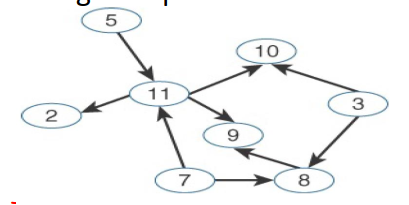
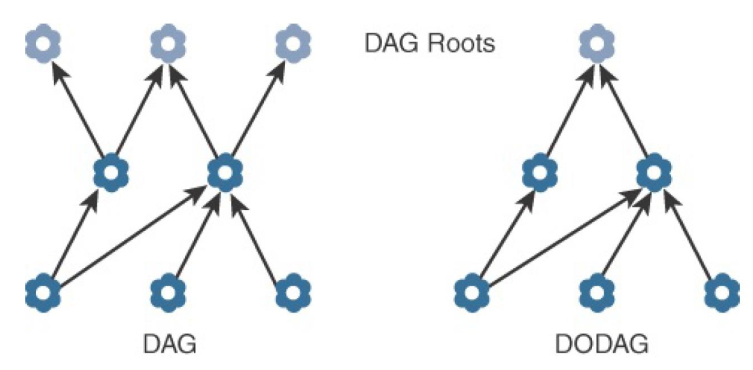
فصل 4

* مزیت‌های استفاده از IP برای لایه نتورک
  + رایج، استاندارد، scalable، امن، قابل مدیریت، stable
* دو مدل برای استفاده از IP در آیوتی
  + Adaptation
    - بتونیم پروتکل‌های غیر IP رو ترجمه کنیم که هماهنگ با IP کار کنن
  + Adoption
    - همه پروتکل‌های غیر IP رو با معادل IPشون عوض کنیم
  + کدوم مدل بهتره؟ فاکتورهای زیر مهمن:
    - ارتباط دوطرفه یا یکطرفه؟ گاهی دیوایس‌های آیوتی فقط اطلاعات میفرستن و چیزی دریافت نمیکنن => نمی‌صرفه یه IP Stack کامل بزنیم
    - سربار: اگه دیوایس آیوتی داده سبکی می‌فرسته، ممکنه هزینه ارسال هدر IP از هزینه ارسال داده بیشتر بشه!
  + مدل Data Flow
    - یکی از مزیت‌های IP اینه که هر دو نود دلخواه می‌تونن با هم ارتباط برقرار کنن، ولی توی آیوتی گاهی وقتا یه دیوایس با دو سه تا اپلیکیشن بیشتر کار نداره => adaptation بهتره
  + Constrained Nodes (نودهایی که قدرت و منابع کمی دارن)
    - دیوایس‌هامون ضعیف هستن، داده‌های کمی ارسال می‌کنن => adaptation بهتره
    - دیوایس‌ها Constrained نیستن => adoption یا adaptation بسته به کاربرد
    - دیوایس‌ها شبیه PC ها قوی هستن ولی نتورک ضعیفه (مثلا پهنای باند کمه) => معمولا adoption. ولی طراحی شبکه و رفتار اپلیکیشن‌ها طوریه که با ضعف شبکه کنار بیاد
  + Constrained Networks: لینک‌های low-power و low-bandwidth
    - نرخ رسوندن پکت‌ها (PDR) نوسان می‌کنه
    - ممکنه خیلی ارور بخوریم، یا کلا ارتباط قطع بشه
    - ترافیک کنترلی (داده‌های غیر اپلیکیشنی) باید کم بشه تا جای ممکن تا فضای داده کاربردی رو اشغال نکنه
    - مصرف باتری هم مسئلست
  + برای Constrained Nodes and Networks از optimization استفاده می‌کنیم (روی لایه‌ها و پروتکل‌ها)
  + 
  + Adaptation layer: بسته‌بندی IP روی لایه‌های پایینتر مثل PHY و MAC. این لایه شامل Optimization هاییه که برای مواجهه با محدودیت‌های شبکه‌ها و دستگاه‌های Constrained ازشون استفاده می‌کنیم.
* مشکل هدر
  + توی 802.15.4 ماکسیمم سایز فریم باید 127 بایت باشه، اگه سایز هدر‌های udp, ip, mac, link layer security رو ازش کم کنیم، فقط 33 کیلوبایت برای پیلود می‌مونه!
  + 
* استفاده از 6LoWPAN به عنوان Adaptation Layer برای کاربردهای آیوتی
  + 
  + سه تا مزیت داره: Header Compression, Fragmentation, Mesh Addressing
  + 
  + 6LoWPAN از دانش قبلی استفاده می‌کنه – استفاده از دانشی که از هر نود از نقشش توی نتورک کلی داره
  + 6LoWPAN بعضی از فیلدهای هدر‌ها رو با فرض کردن مقدار دیفالتِ رایج حذف می‌کنه => سربار هدر کم میشه
  + یه فریم 6LoWPAN بدون هدر کامپرشن:
    - 40 بایت هدر IPv6
    - 8 بایت هدر UDP
    - 1 بایت هدر 6LoWPAN
    - => 53 بایت از 127 بایتی که بالا گفتیم برای دیتا پیلود می‌مونه
  + یه فریم 6LoWPAN با هدر کامپرشن:
    - هدر 6LoWPAN میشه 2 بایت که هدر کامپرس شده IPv6 رو جا بده
    - هدر UDP میشه 4 بایت
    - پیلود از 53 بایت می‌رسه به 108 بایت!
  + 
  + مسئله Fragmentation: برای IPv6 یه MTU داریم که 1280 بایته. برای 802.15.4 یه MTU 127 بایتی داریم => بسته‌های IPv6 باید فرگمنت بشن و ارسال بشن (توی لایه 2 (از پایین قاعدتا!))
  + هدر Fragment در 6LoWPAN سه فیلد داره
    - Datagram Size: کل سایز پیلود Unfragmented
    - Datagram Tag: مجموعه فرگمنت‌ها برای یه پیلود رو مشخص می‌کنه
    - Datagram Offset: فرگمنتمون کجای پیلوده
  + 
  + Fragmentation Header
    - یه بیت‌داره که مشخص می‌کنه فیلد‌های قبلیش فیلد‌های Fragment هستن یا یه ویژگی دیگه مثل Header Compression
    - توی اولین فرگمنت Datagram Offset نداریم => اولین Fragmentation Header چهار بایته
    - بقیه Fragmentation Header ها 5 بایتن چون فیلد Datagram Offset هم دارن
  + ویژگی Mesh Addressing
    - هدف: فوروارد کردن پکت‌ها
    - سه فیلد:
      * Hop Limit: ماکسیمم تعداد فوروارد ها (با هر فوروارد، یدونه از مقدارش کم میشه، به صفر که برسیم پکت دراپ میشه)
      * Source Address
      * Destination Address
    - 
    - دو آپشن برای مسیریابی و فورواردینگ
      * Mesh-under: روتینگ توسط 6LoWPAN انجام بشه
        + از هدر Mesh Addressing استفاده میشه و پکت توی Link Layer فوروارد میشه.
        + در Link Layer هر نود Forwarding Table داره
        + Gateway ها

دامنه کاری Mesh-under تموم میشه (؟)

ترجمه لایه 2 و IP رو انجام میدن

* + - * Mesh-over: از روتینگ IP استفاده کنیم
        + هر نود حکم یه روتر رو داره
        + اگر یه LoWPAN با تکنولوژی متفاوت روی Link Layer پیاده‌سازی بشه، استفاده از این روش خوبه
        + بجای استفاده از الگوریتم‌ها و پروتکل‌های روتینگ معمول IP، از پروتکل RPL استفاده می‌کنه
  + 6Lo Working Group
    - دارن تلاش می‌کنن IPv6 رو روی Constrained-noded Networks هم ببرن
    - با اینکه 6LoWPAN روی 802.15.4 خوب کار می‌کنه، باید بتونیم IPv6 رو روی پروتکل‌های دیگه Link Layer هم استفاده کنیم
      * بلوتوث، 802.11ah، WIA-PIA، ...
  + RPL: Routing over Low-power and Lossy Networks
    - دو مود
      * Storing mode: توی هر نود کل جدول روتینگ ذخیره میشه، یا هر نود دقیقا می‌دونه چطوری به همه نودهای دیگه برسه
      * Non-Storing Mode: فقط روترهای مرزی کل جدول روتینگ رو دارن. نودهای دیگه فقط مسیر به روترهای مرزی رو دارن. هر پکتی که میاد به روترهای مرزی فوروارد میشه، و روترهای مرزی می‌دونن از کدوم مسیر فورواردش کنن که به مقصد برسه.
    - بر اساس DAG: Directed Acyclic Graph کار می‌کنه: گراف‌های جهت‌دار بدون دور
      * 
      * یال‌ها طوری مرتب شدن که همه مسیرها به root node ها ختم میشن
      * پروسه کلی: DODAG بساز. DODAG: یه DAG که مسیرهاش فقط به یه نود (root node) ختم میشن. این نود یه روتر مرزیه به اسم DODAG root
      * 
      * سمت چپی DAG عه و سه تا root داره. سمت راستی DODAG عه و فقط یدونه root داره.
      * هر نود در DODAG حداکثر به سه تا parent وصله که نهایتا میرسن به root. یکی از این parent ها معمولا preferred parent هستش که hop بعدی رو مشخص می‌کنه و برای رسیدن به root انتخاب میشه.
      * انواع مسیج مرتبط با DAG
        + DAG Information Object (DIO): مسیرهای رو به بالا (به سمت ریشه) با این پیام‌ها مشخص میشن. نودها منتظر رسیدن DIO ها می‌مونن تا تغییرات توی توپولوژی رو متوجه بشن. DIO شامل اطلاعات parent ها و بهترین مسیرها برای رسین به DODAG root هستن.
        + Destination Advertisement Objects (DAO): نودها با این پیام‌ها مسیرهای رو به پایین به دست میارن. با DAO به parent هاشون اعلام حضور می‌کنن.
        + این مسیج‌ها روی IPv6 ارسال میشن.
      * حالت Non-storing: نودها توی DAO اطلاعات parent هاشونو فقط به root می‌فرستن و root کل اطلاعات روتینگ رو داره.
      * حالت storing: هر نود اطلاعاتی که از DAO ها می‌گیره رو نگه می‌داره. مصرف توان و سی‌پی‌یو زیاد میشه ولی پکت‌ها مسیر کوتاه‌تری طی می‌کنن که به مقصد برسن، چون هر نود برای خودش می‌تونه تصمیم بگیره پکت‌ها رو چطور فوروارد کنه.
      * 